CURRICULUM

OF

COMPUTER SCIENCE,
SOFTWARE ENGINEERING, AND
INFORMATION TECHNOLOGY

(Bachelors & Masters Programs)

(Revised 2017)

HIGHER EDUCATION COMMISSION
ISLAMABAD

CURRICULUM DIVISION, HEC

Prof. Dr. Mukhtar Ahmed	Chairman, HEC
Prof. Dr. Arshad Ali	Executive Director, HEC
Mr. Muhammad Raza Chohan	Director General (Academics)
Dr. Muhammad Idrees	Director (Curriculum)
Syeda Sanober Rizvi	Deputy Director (Curriculum)
Mr. Riaz-ul-Haque	Assistant Director (Curriculum)
Mr. Muhammad Faisal Khan	Assistant Director (Curriculum)

CONTENTS

Contents

CURRICULUM OF COMPUTING PROGRAMS ....
INTRODUCTION ....
Curricula Consideration ....
Computing Discipline ....
Introduction ....
Bachelor Degree Programs in Computing ....
Eligibility Criteria ....
Duration ....
Degree Completion Requirements ....
Master Degree Programs in Computing ....
Eligibility Criteria ....
Duration ....
Degree Completion Requirements ....
Program Learning Outcomes (PLOs) ....
Curriculum Design for Bachelor Degrees in Computing ....
COURSES COMMON for BS (CS/ IT/ SE) ....
Computer Science Program BS (CS) ....
Development in Computer Science ....
Program Structure: ....
BS Computer Science ....
Coverage of ACM Knowledge Areas ....
Proposed Curriculum for BS-CS ....
COURSES COMMON to all computing bachelor programs ....
Domain Courses for BS (COMPUTER SCIENCE) ....
Proposed Study Plan for BS (Computer Science) ....
Proposed Curriculum for BS-IT ....
COURSES COMMON to all computing bachelor programs ....
Domain Courses for BS-IT ....
Proposed Study Plan for BS (Information Technology) ....
BS Software Engineering ....
Proposed Curriculum for BS-SE ....
COURSES COMMON to all computing bachelor programs ....
Domain Courses for BS-SE ....
Proposed Study Plan for BS (Software Engineering) ....
Master of Science ....
Programs ....

Composed by: Mr. Zulfiqar Ali, HEC, Islamabad

PREFACE

The curriculum, with varying definitions, is said to be a plan of the teaching-learning process that students of an academic program are required to undergo to achieve some specific objectives. It includes scheme of studies, objectives & learning outcomes, course contents, teaching methodologies and assessment/ evaluation. Since knowledge in all disciplines and fields is expanding at a fast pace and new disciplines are also emerging; it is imperative that curricula be developed and revised accordingly.

University Grants Commission (UGC) was designated as the competent authority to develop, review and revise curricula beyond Class-XII vide Section 3, Sub-Section 2 (ii), Act of Parliament No. X of 1976 titled “Supervision of Curricula and Textbooks and Maintenance of Standard of Education”. With the repeal of UGC Act, the same function was assigned to the Higher Education Commission (HEC) under its Ordinance of 2002, Section 10, Sub-Section 1 (v).

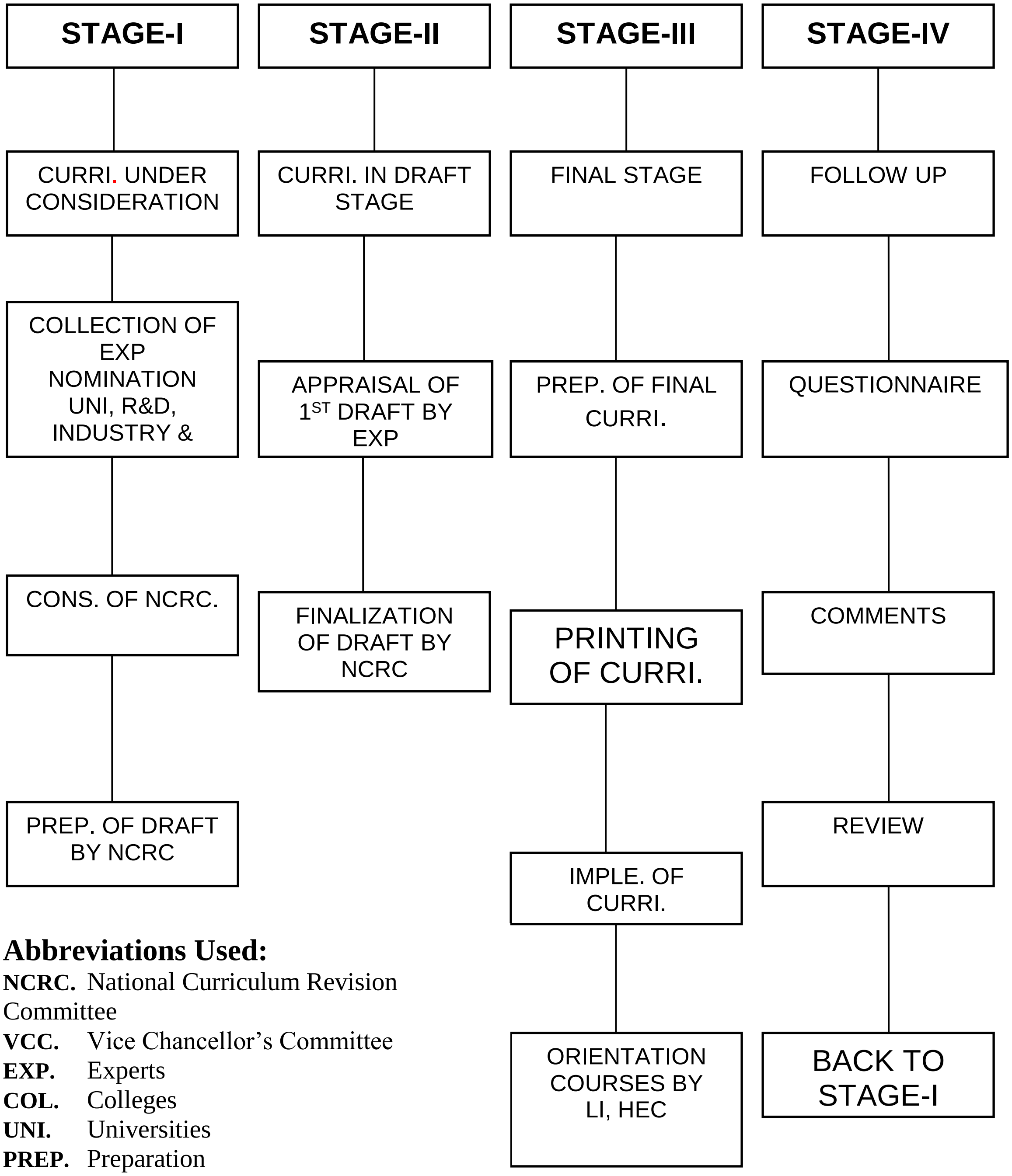
In compliance with the above provisions, the Curriculum Division of HEC undertakes the revision of curricula regularly through respective National Curriculum Revision Committees (NCRCs) which consist of eminent professors and researchers of relevant fields from public and private sector universities, R&D organizations, councils, industry and civil society by seeking nominations from their organizations.

In order to impart quality education which is at par with indigenous needs and international standards, HEC NCRCs have developed unified framework/ templates as guidelines for the development and revision of curricula in the disciplines of Basic Sciences, Applied Sciences, Social Sciences, Agriculture and Engineering.

It is hoped that this curriculum document, prepared by the respective NCRC’s, would serve the purpose of meeting our national, social and economic needs, and it would also provide the level of competency specified in Pakistan Qualification Framework to make it compatible with international educational standards. The curriculum is also placed on the website of HEC
http://hec.gov.pk/english/services/universities/RevisedCurricula/Pages/default.aspx

(Muhammad Raza Chohan)
Director General (Academics)

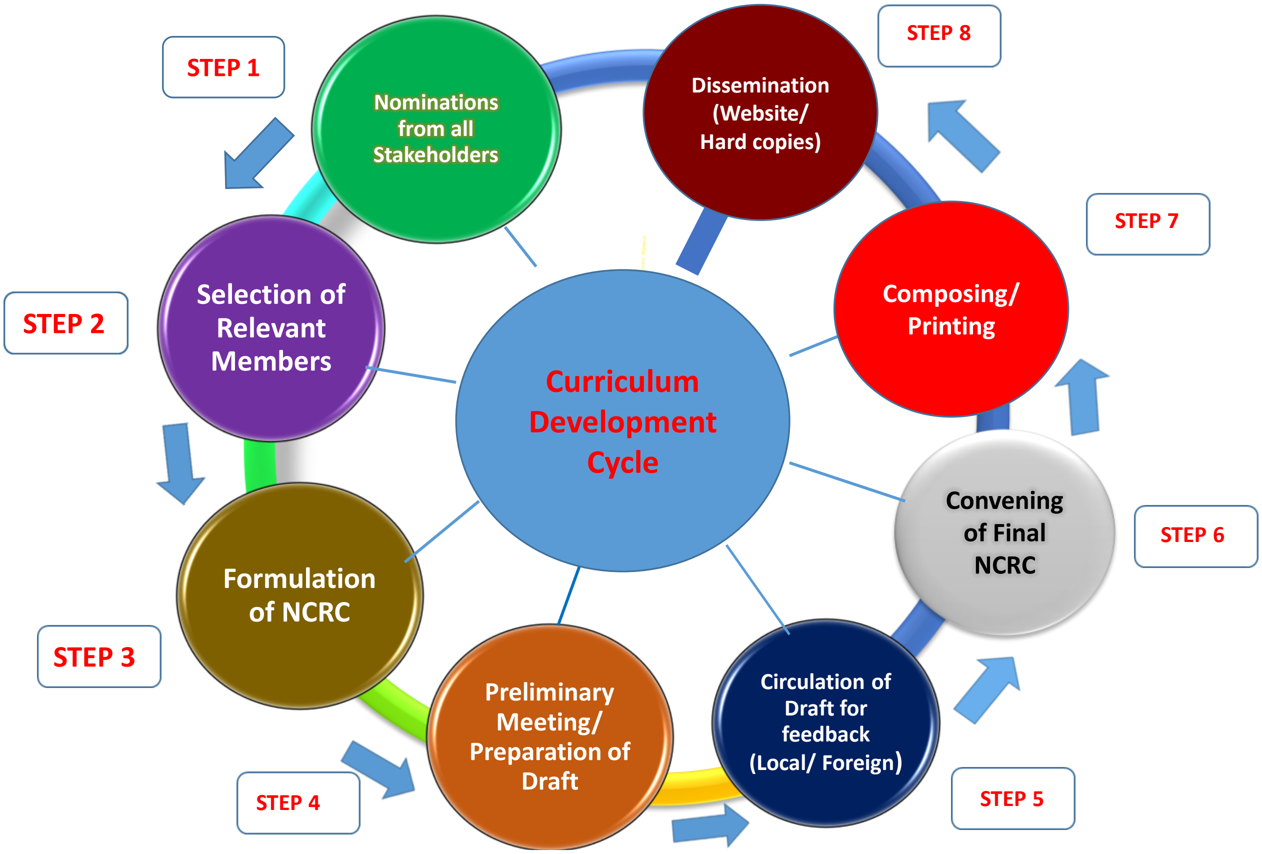
CURRICULUM DEVELOPMENT



Abbreviations Used:

NCRC.	National Curriculum Revision Committee
VCC.	Vice Chancellor's Committee
EXP.	Experts
COL.	Colleges
UNI.	Universities
PREP.	Preparation
REC.	Recommendations
LI	Learning Innovation
R&D	Research & Development Organization
HEC	Higher Education Commission
CONS:	Constitution

CURRICULUM DEVELOPMENT CYCLE


STEP 1
Nominations from all Stakeholders
STEP 2
Selection of Relevant Members
STEP 3
Formulation of NCRC
STEP 4
Preliminary Meeting/ Preparation of Draft
STEP 5
Circulation of Draft for feedback (Local/ Foreign)
STEP 6
Convening of Final NCRC
STEP 7
Composing/ Printing
STEP 8
Dissemination (Website/ Hard copies)
Curriculum Development Cycle

MINUTES OF THE FINAL MEETING:

The second and final meeting of the National Curriculum Revision Committee for computing programs was held from August 21-23, 2017 at HEC Regional Centre, Lahore. Aims and objectives of this meeting were to discuss and finalize the Preliminary Draft curriculum of Computer Science, Software Engineering & Information technology. Also to make the curriculum compatible with international standards, satisfying indigenous demands as well as ensuring uniformity of academic standards within the country.

2. Following honourable members took part in revising the curricula of Computing Programs.

Convenor

1. Dr. Mohammad Ayub Alvi
   Chairman, National Computing Education Accreditation Council (NCEAC)
   Higher Education Commission, Sector H-8/1, Islamabad

Secretary

2. Dr. Sharifullah Khan
   Associate Professor, SEECS
   NUST, Sector H-12, Islamabad

Content Coordinator

3. Dr. Shoab Ahmed Khan
   CEO, CARE
   HoD (C&SE), NUST, Sector H-12, Islamabad

Members (in alphabetical order)

4. Dr. Aarij Mahmood Hussaan
   Assistant Professor, Department of Computer Science
   Iqra University, Defence View, Karachi

5. Dr. Abdul Aziz
   Professor / Dean, Department of Computer Sciences
   The Superior College, Raiwind Rd, Lahore

6. Dr. Abdul Hussain Shah Bukhari
   Vice Chancellor/Rector, Sindh Institute of Management Technology
   LS-37/10 Sector-15, Main Korangi Industrial Road, Karachi

7. Dr. Adnan Abid
   Associate Professor, Department of Computer Sciences
   University of Management & Tech, Lahore

8. Dr. Akhtar Hussain Jalbani
   Associate Professor, Department of Information Technology
   Quaid-e-Awam University of Engg, Science & Technology, Nawabshah

9. Dr. Asad Habib
   Assistant Professor, Institute of Information Technology
   Kohat University of Science & Tech, Kohat

10. Dr. Bakhtiar Khan Kasi
Assistant Professor, Department of Computer Sciences
BUITEMS, Takatu Campus, Quetta

11. Dr. Fahad Tahir
Assistant Professor, Department of Computer Sciences
Air University, Service Rd, E-8, Islamabad

12. Dr. Fahima Tahir
Assistant Professor, Department of Computer Sciences
Lahore College for Women University, Jail Road, Lahore

13. Dr. Ghufran Ullah
HoD / Assistant Professor, Department of Computer Sciences
City University of Science & Information Technology, Peshawar

14. Dr. Hafiz Muhammad Shahzad Asif
Associate Professor, Department of Computer Science & Engineering
University of Engineering & Technology, G. T. Road, Lahore

15. Dr. Hannan Bin Liaqat
Assistant Professor, Department of Information Technology
Arfa Karim Block, Hafiz Hayat Campus, University of Gujrat, Gujrat

16. Dr. Huma Hayat Khan
Assistant Professor, Department of Software Engineering
National University of Modern Languages, Sector H-9/1, Islamabad

17. Dr. Husnain Mansoor Ali
Associate Professor, Department of Computer Science
Shaheed Zulfiqar Ali Bhutto Institute of Science & Tech., Clifton, Karachi

18. Dr. Iftikhar Azim Niaz
Adviser, Engineering Accreditation Department
Pakistan Engineering Council, Ataturk Avenue, Sector G-5/2, Islamabad

19. Dr. Junaid Haroon Siddiqui
Assistant Professor, Department of Computer Sciences
Lahore University of Management Sciences, Lahore

20. Dr. Kamran Taj Pathan
Associate Professor, Department of Software Engineering
University of Sindh, Jamshoro

21. Dr. Kifayat Ullah
Assistant Professor, Department of Computer & Software Technology
University of Swat, PTCL Campus, Kanju, Township, Swat

22. Dr. M. Abdul Rehman
HoD/ Associate Professor, Department of Computer Sciences
Sukkur Institute of Business Administration (IBA), Airport Road, Sukkur

23. Dr. M. Ahsan Latif
Assistant Professor, Department of Computer Sciences
University of Agriculture, Faisalabad

24. Dr. Mamoona Asghar
    Assistant Professor, Department of Computer Sciences
    The Islamia University of Bahawalpur, Baghdad Campus, Bahawalpur

25. Dr. Masood Raza
    HoD / Associate Professor, Department of Computer Science
    Muslim Youth University, Street No. 40, Sector G-10/4, Islamabad

26. Dr. Muhammad Amjad Iqbal
    Associate Professor, Faculty of Information Technology
    University of Central Punjab, Lahore

27. Dr. Muhammad Asif Habib
    Assistant Professor, Department of Computer Sciences
    National Textile University, Sheikhupura Road, Faisalabad

28. Dr. Muhammad Ilyas
    Assistant Professor, Department of Computer Science & IT
    University of Sargodha, Sargodha

29. Dr. Muhammad Imran
    Assistant Professor, Department of Computer Science & IT
    Sarhad University of Science & Information Technology, Peshawar

30. Dr. Muhammad Inam ul Haq
    HoD / Assistant Professor, Dept. of Computer Sciences & Bioinformatics
    Khushal Khan Khattak University, Karak

31. Dr. Muhammad Javed
    Assistant Professor, Department of Computer Science & IT
    University of Science & Technology, Township, Bannu

32. Dr. Muhammad Mobeen Movania
    Assistant Professor, Department of Computer Sciences
    DHA Suffa University, Phase-VII, DHA, Karachi

33. Dr. Muhammad Ramzan Talib
    Associate Professor, Department of Computer Sciences
    Government College University, Allama Iqbal Road, Faisalabad

34. Dr. Muhammad Riaz Mughal
    Dean/Professor, Faculty of Engineering & Technology
    Mirpur University of Science & Technology, Allama Iqbal Rd, Mirpur, AJK

35. Dr. Muhammad Saqlain
    Assistant Professor, Dept. of Computer Science & Software Engineering
    International Islamic University, Islamabad

36. Dr. Muhammad Shahbaz
    Professor, Department of Computer Science & Engineering
    University of Engineering & Technology, G. T. Road, Lahore

37. Dr. Najeeb Ullah Khan
    Chairman / Assistant Professor, Department of Computer Science
    CECOS Univ. of IT & Emerging Sciences, Hayatabad, Peshawar

38. Dr. Najeed Ahmed Khan
Associate Professor, Department of Computer Science & IT
NED University of Engineering & Tech., University Road, Karachi

39. Dr. Nayyar Masood
Associate Professor, Department of Computer Sciences
Capital University of Science & Technology, Kahuta Road, Islamabad

40. Dr. Neelam Gohar
Assistant Professor, Department of Computer Sciences
Shaheed Benazir Bhutto Women University, Charsadda Road, Peshawar

41. Dr. Niaz Hussain Arijo
Associate Professor, Department of Software Engineering
University of Sindh, Jamshoro

42. Dr. Pardeep Kumar
HoD / Associate Professor, Department of Computer System Engineering
Quaid-e-Awam University of Engg, Science & Technology, Nawabshah

43. Dr. Rehan Inam Qureshi
Associate Professor, Department of Computer Software Engineering
Bahria University, Karachi

44. Dr. Riaz ul Amin
Chairman / Associate Professor, Department of Computer Sciences
BUITEMS, Airport Rd, Takatu Campus, Quetta

45. Dr. Saleem Ullah
HoD / Assistant Professor, Department of Information Technology
Khawaja Fareed UEIT,Abu Dhabi Rd, Rahim Yar Khan

46. Dr. Shaikh Muhammad Munaf Rashid,
Chairman Software Engineering, Faculty of Engg Science & Technology
Ziauddin University, Block-B, North Nazimabad, Karachi

47. Dr. Sheeraz Memon
Chairman/Associate Professor, Department of Computer System Engg.
Mehran University of Engg & Technology, Jamshoro

48. Dr. Sibt ul Hussain
Assistant Professor, Department of Computer Sciences
NUCES-FAST, H-11, Islamabad

49. Dr. Suleman Mazhar
Assistant Professor, Department of Computer Sciences
Information Tech. University, Arfa Software Technology Park, Lahore

50. Dr. Syed Asad Raza Kazmi
Assistant Professor, Department of Computer Sciences
GC University, Lahore

51. Dr. Syed Fawad Hussain
Associate Professor, Faculty of Computer Science & Engg
GIK Institute of Engineering Science & Technology, Topi, Swabi, KPK

52. Dr. Syed Tahir Qasim
Assistant Professor, Department of Computer Sciences
NUCES-FAST, Karachi

53. Dr. Tamim Ahmed Khan
HoD / Associate Professor, Department of Software Engineering
Bahria University, E-8, Shangrilla Rd, Islamabad

54. Dr. Tauseef Jamal
Associate Professor, Department of Computer & Information Sciences
Pakistan Institute of Engineering & Applied Sciences, Islamabad

55. Dr. Waqar Aslam
Assistant Professor, Department of Computer Sciences
The Islamia University of Bahawalpur, Baghdad Campus, Bahawalpur

56. Dr. Yasir Arfat Malkani
Associate Professor, Institute of Computer Sciences & IT
University of Sindh, Jamshoro

57. Dr. Zahid Hussain Abro
Dean/Professor, Department of Information Technology
Quaid-e-Awam University of Engg, Science & Technology, Nawabshah

58. Dr. Zulfiqar Ali Memon
Associate Professor, Department of Computer Science
NUCES-FAST, Karachi

59. Dr. Zulfiqar Habib
Chairman / Professor, Department of Computer Sciences
COMSATS, off Defence Road, Lahore

60. Mr. Asim Ghaffar
Vice President, R & D, LMKR
2nd Floor, Evacuee Trust Complex, F-5, Islamabad

61. Mr. Haroon Rashid Kanth
CEO and Centre Head, Teradata Global consulting Centre
TF Complex, 2nd Floor, 7 Mauve Area, G-9/4, Islamabad

62. Mr. Irfan Shahzad
Director South Asia (Public Sector), Oracle Corporation
4th floor, Ufone Tower, Blue Area, Islamabad

63. Mr. Muhammad Anwaar Saeed
Assistant Professor, Department of Computer Sciences
VU (Islamabad Campus), Sector G-10/4, Islamabad

64. Mr. Riaz-Ul-Haque
Assistant Director (Curriculum), Higher Education Commission
Sector H-8, Islamabad

65. Mr. Syed Ali
CEO, 7Vals
264-CCA, FF Block, Ground Floor, Sector V, DHA, Lahore

The meeting started with recitation of verses from the Holy *Quran* by Dr. Sharifullah Khan, Secretary of this NCRC. Mr. Riaz-ul- Haque, Assistant Director (Curriculum) and HEC Coordinator briefed the participants about the aims and objectives of the meeting and the process of curriculum printing and dissemination for adoption by the universities and DAIs of Pakistan.

3. Members of the Committee unanimously agreed to continue Dr. Mohammad Ayub Alvi, Chairman HEC NCEAC, and Dr. Sharifullah Khan, Associate Professor, SEECS, NUST as **Convener** and **Secretary** of the NCRC, respectively.

4. During the Preliminary meeting held from April 4-6, 2017 at HEC Islamabad, the house was divided in four sub-groups for revision of their respective domains. The following sub-groups were formed, which were led by a Chair Person and an Associate.

A. Core Computing Group

Chair	Dr. Nayyer Masood, CUST, Islamabad
Associate:	Dr. Junaid Haroon, LUMS, Lahore

B. Computer Science Group

Chair	Dr. Suleman Mazhar, ITU, Lahore
Associate:	Dr. Fawad Hussain, GIKI, Topi

C. Software Engineering Group

Chair	Dr. Iftikhar Azim Niaz, PEC, Islamabad
Associate:	Dr. Tamim Khan, Bahria Univ., Islamabad

D. Information Technology

Chair	Dr. Masood Raza, MY University, Islamabad
Associate:	Dr. Hannan Bin Liaqat, Univ. of Gujrat

In order to finalize the preliminary drafts of respective domains, on the first day of final meeting, the house was again divided into four groups. Following were the Chairpersons and Associates of these groups.

A. Computing Group

Chair	Dr. Nayyer Masood, CUST, Islamabad
Associate:	Dr. Zulfiqar Memon, FAST-NU, Karachi

B. Computer Science Group

Chair	Dr. Suleman Mazhar, ITU, Lahore
Associate:	Dr. Fawad Hussain, GIKI, Topi

C. Software Engineering Group

Chair	Dr. Iftikhar Niaz, CIIT, Islamabad
Associate:	Dr. Tamim Khan, Bahria Univ., Islamabad

D. Information Technology

Chair	Dr. Sheeraz Memon, MUET, Jamshoro
Associate:	Dr. Waqar Aslam, IUB, Bahawalpur

5. The Committee during the proceedings of the meeting, considered the inputs given by the members and incorporated their suggestions in the curriculum document as deemed necessary. After thorough discussion and having three days deliberations, the committee achieved the following objectives:-

i. Finalized the revision process of the draft curriculum in the discipline of Computer Science, Software Engineering, and Information Technology in order to bring it at par with international standards.
ii. Revised Vision, Mission, and Scope of the discipline.
iii. Revised /developed objectives / learning outcomes, list of contents and assessment criteria (formative & summative) aligned with undergraduate programs (vertical approach) and other graduate level programs (horizontal approach).
iv. Incorporated/suggested latest reading materials/references (local & international) against each course.
v. Made recommendations for promotion/development of the discipline, keeping in view the futuristic needs of the society and revival of our values and culture.
vi. Finalized the intake criteria for BS/MS programs.

6. The Convener thanked the NCRC members for their inputs in finalizing the revision of draft curriculum of Computer Science, Software Engineering, and Information Technology by keeping in view the requirements of the country and to make it more practical, competitive and effective.

7. The committee highly appreciated the hospitality shown by officials of HEC Regional Centre, Lahore and Assistant Director and his Aide from HEC Islamabad for making proper arrangements to facilitate the members of committee. Committee members applauded the kind patronage of Dr. Muhammad Ayub Alvi, the **Convener** and Dr. Sharifullah Khan, **Secretary** during the proceedings of the NCRC meeting.

The meeting ended with the vote of thanks to and from the chair.

Curricula Consideration

Association of Computing Machinery (ACM), USA is the largest body in the world for computer scientists. Its membership is spread over the entire globe. It has a pool of highly reputed professionals which meet after a few years to assess the directions being taken by the computing discipline. In view of its assessment, it identifies knowledge areas and also their relative importance in the years to come. Thus, ACM shows the path to follow to the computing academia and professionals all over the world.

The committee kept the latest approved ACM recommendations in view, which are for Computer Science (2013) and Software Engineering (2014). Another consideration was to aim for a curriculum, which meets the current market requirements. The committee also approved common eligibility criteria for admission for all Bachelor degree programs in Computing.

Bachelor of Science Programs

Curriculum for Bachelor Degrees in Computing

Introduction

Computing is emerging as (need to write a paragraph)

Bachelor Degree Programs in Computing

Computer Science (BS-CS)
Information Technology (BS-IT)
Software Engineering (BS-SE)

Eligibility Criteria

The minimum requirements for admission in a Bachelor degree program in Computer Science/ Information Technology/ Software Engineering, is **at least 50% marks in Intermediate (HSSC) examination with Mathematics or equivalent qualification with Mathematics certified by IBCC**.

Duration

The **minimum duration** for completion of BS degree is four years. The HEC allows a **maximum period of seven years** to complete BS degree requirements.

Degree Completion Requirements

To become eligible for award of BS degree, a student must satisfy the following requirements:

a) Must have studied and passed the **prescribed courses, totaling at least 130 credit hours**.
b) Must have earned **CGPA** (Cumulative Grade Point Average) **of at least 2.0 on a scale of 4.0**.

Program Learning Outcomes (PLOs)

Computing programs prepare students to attain educational objectives by ensuring that students demonstrate achievement of the following outcomes (derived from Graduate Attributes define by Seoul Accord www.seoulaccord.org).

Program Learning Outcomes (PLOs)	**Computing Professional Graduate**
1. **Academic Education**	To prepare graduates as computing professionals
2. **Knowledge for Solving Computing Problems**	Apply knowledge of computing fundamentals, knowledge of a computing specialization, and mathematics, science, and domain knowledge appropriate for the computing specialization to the

	abstraction and conceptualization of computing models from defined problems and requirements
3. **Problem Analysis**	Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines
4. **Design/ Development of Solutions**	Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations
5. **Modern Tool Usage**	Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations
6. **Individual and Team Work**	Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary settings
7. **Communication**	Communicate effectively with the computing community and with society at large about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions
8. **Computing Professionalism and Society**	Understand and assess societal, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice
9. **Ethics**	Understand and commit to professional ethics, responsibilities, and norms of professional computing practice
10. **Life-long Learning**	Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional

BS Curriculum Design

The combined structure of BS Programs in Computing is proposed to meet the needs of students through theory and practical computing experience. The students are expected to learn theoretical and practical understanding of the respective field of Computing.

The proposed structure is dynamic and provides basis for various options including Breadth-Based, Depth-Based, and Integrated Breadth & Depth-Based specializations. Student may choose a particular option, which is most appropriate to their planned future career. The following are some relevant details:

- Minimum credit hours shall be 130 for BS (CS, SE, IT) programs.
- Each program comprises eight semesters spread over four years.
- The following table gives the distribution of credit hours in different domains of knowledge.

Table 1.2: Areas Covered in BS programs

Course Group	Credit hours	% age
General Education	19	15%
University Electives	12	9%
Math & Science Foundation	12	9%
Computing – Core	39	30%
Common courses	**82**	**63%**
Domain (CS/ IT/SE)		
Domain Core (CS/IT/SE)	24	18%
Domain Electives (CS/IT/SE)	15	12%
Domain Supporting (CS/IT/SE)	9	7%
Domain courses	**48**	**37%**
TOTAL	**130**	**100%**

COURSES COMMON for BS (CS/ IT/ SE) – 82 Credits

Computing Core Courses

Course Title	Credit hours
Programming Fundamentals	3-1
Object Oriented Programming	3-1
Data Structures & Algorithms	3-1
Discrete Structures	3-0
Operating Systems	3-1
Database Systems	3-1
Software Engineering	3-0
Computer Networks	3-1
Information Security	3-0
Final Year Project	0-6
Total	**39 (27-12)**

General Education Courses

Course Title	Credit hours
English Composition & Comprehension	3
Technical & Business Writing	3
Communication & Presentation Skills	3
Professional Practices	3
Intro. to Info. & Comm. Technologies	2-1
Pakistan Studies	2
Islamic Studies/ Ethics	2
Total	**18-1**

University Elective Courses

(Not limited to the areas listed below, Institutions may add more courses)

Course Title	Credit hours
Foreign Language	2-0
Social Service	1-0
Management Related	3-0
Social Science Related	3-0
Economy Related	3-0
Total	**12**-0

Mathematics and Science Foundation Courses

Course Title	Credit Hours
Calculus & Analytical Geometry	3-0
Probability & Statistics	3-0
Linear Algebra	3-0
Applied Physics	3-0
Total	**12-0**

BS Computer Science

Computer Science Program BS (CS)

A complete detail of BS Program in CS involving Program structure and distribution of credits among various components of Program are discussed in the following pages.

Development in Computer Science

Recent developments in computer hardware, software and communication technologies have offered new exciting opportunities and challenges for creation of innovative learning environments for Computer Science and its curricula design. One of the key elements here is to prepare the graduates for the future. The challenge of getting all newly emerging technologies incorporated in to the curriculum is becoming pivotal for the effectiveness of curricula. There is a need for curricula structures that are really able to grow as we put new demands on them. The curriculum is required to provide integration of all components and the foundations that allow accessing all of the new knowledge and technology to fulfil the vision of future.

The basic intention of an academic Program in Computer Science is to develop the student's critical professional thinking and intuition. The curriculum must be structured to provide a balanced mixture of theory and practical experiences at foundation and advance levels to make the graduate capable of sound professional decisions. As a result the graduate should be able to assume responsible positions in business, government, and education at the research, development, and planning levels. The Program should also provide an excellent foundation for further formal learning and training. The Computer Science curriculum is expected to provide environments to put into practice, the principles and techniques learnt during the course of implementation of academic Program.

The following summarizes some key characteristics for consideration as a basis of a successful academic Program in Computer Science:

1. The Program should provide a broad understanding of the field via introducing concepts, theory, and techniques.
2. Intensive education/training in focused areas of Computer Science is desirable.
3. The Program may encourage students to develop and use abstract models in addition to apply respective technology in practical situations.
4. Computer Science graduates require special communication skills both orally and in writing. They must be able to produce well-organized reports, which clearly delineate objectives, methods of solution, results, and conclusions for a complex task.
5. The Program should provide formal foundations for higher learning.
6. The Program should be dynamic and flexible enough to maintain currency with the latest scientific and technological developments in the field.
7. The Program should provide professional orientation to prepare students for industry.

Program Structure:

BS Computer Science

Computer science is the study of the theory, experimentation, and engineering that form the basis for the design and use of computers. It is the scientific and practical approach to computation and its applications and the systematic study of the feasibility, structure, expression, and mechanization of the methodical procedures (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to information [ref WordNet Princeton definition].

Computer Science is the application of a systematic, disciplined and quantifiable approach to the design, development, operation, and maintenance of software systems. It is in fact the practice of designing and implementing large, reliable, efficient and economical software by applying the principles and practices of engineering. The program aims to train students in all aspects of software life cycle from specification through analysis and design to testing, maintenance and evaluation of software product.

Coverage of ACM Knowledge Areas

Computer Science curriculum is designed keeping in view following identified knowledge areas of ACM [ref # ACM 2013 curriculum report]. It has been tried to reasonably cover all knowledge areas without compromising the flexibility needed for a national model curriculum.

• AL -	Algorithms and Complexity
• AR -	Architecture and Organization
• CN -	Computational Science
• DS -	Discrete Structures
• GV -	Graphics and Visual Computing
• HCI -	Human-Computer Interaction
• IAS -	Information Assurance and Security
• IM -	Information Management
• IS -	Intelligent Systems
• NC -	Networking and Communications
• OS -	Operating Systems
• PBD -	Platform-based Development
• PD -	Parallel and Distributed Computing
• PL -	Programming Languages
• SDF -	Software Development Fundamentals
• SE -	Software Engineering
• SF -	Systems Fundamentals
• SP -	Social Issues and Professional Issues

Proposed Curriculum for BS-CS

Table 1.2: Areas Covered in BS programs

Course Group	Credit hours	% age
General Education	19	15%
University Electives	12	9%
Mathematics & Science Foundation	12	9%
Computing – Core	39	30%
Common courses	**82**	**63%**
Domain CS		
Domain CS Core	24	18%
Domain CS Electives	15	12%
Domain CS Supporting	9	7%
Domain courses	**48**	**37%**
TOTAL	**130**	**100%**

Courses common for all computing BS programs – 82 Credits

Computing Core Courses

Course Title	Credit hours
Programming Fundamentals	3-1
Object Oriented Programming	3-1
Data Structures & Algorithms	3-1
Discrete Structures	3-0
Operating Systems	3-1
Database Systems	3-1
Software Engineering	3-0
Computer Networks	3-1
Information Security	3-0
Final Year Project	0-6
Total	**39 (27-12)**

General Education Courses

Course Title	Credit hours
English Composition & Comprehension	3
Technical & Business Writing	3
Communication & Presentation Skills	3
Professional Practices	3
Intro to Info. & Comm. Technologies	2-1
Pakistan Studies	2
Islamic Studies/ Ethics	2
Total	**18-1**

University Elective Courses

(Not limited to the list below, Universities may add more courses)

Course Title	Credit hours
Foreign Language	2-0
Social Service	1-0
Management Related	3-0
Social Science Related	3-0
Economy Related	3-0
Total	**12-**0

Mathematics and Science Foundation Courses

Course Title	Credit hours
Calculus & Analytical Geometry	3-0
Probability & Statistics	3-0
Linear Algebra	3-0
Applied Physics	3-0
Total	**12-0**

Domain Courses for BS (COMPUTER SCIENCE)

Computer Science CORE (Compulsory) courses

Course Title	Credit hours
Compiler Construction	3-0
Comp. Organization & Assembly Language	3-1
Digital Logic Design	3-1
Design & Analysis of Algorithms	3-0
Parallel & Distributed Computing	3-0
Artificial Intelligence	3-1
Theory of Automata	3-0
Total	**24 (21-3)**

Computer Science SUPPORTING courses (ANY 3 from following list)

Coverage of relevant pre-requisite must be ensured while offering any of the following courses from this category

Course Title	Credit hours
Differential Equations	3-0
Multi-variate Calculus	3-0
Graph Theory	3-0
Theory of Programming Languages	3-0
Numerical Computing	3-0
Total (Any three of the above)	**9-**0

Computer Science ELECTIVE courses

Course Title	Credit hours
CS Elective – 1	3
CS Elective – 2	3
CS Elective – 3	3
CS Elective – 4	3
CS Elective – 5	3
Total	**15**

Proposed Study Plan for BS (Computer Science)

4-Year Program (8 Regular Semesters of 18 weeks each)

Course Codes have been assigned as an example only. The purpose is to indicate pre-requisite courses for studying advanced courses.

Semester - I

Code	Course Title	Credit Hours	Pre-requisite
CS 1x1	Introduction to ICT	3-0	
CS 1x2	Programming Fundamentals	3-1	
HU 1x1	English Composition & Comprehension	3-0	
MT 1x1	Calculus & Analytical Geometry	3-0	
NS 1x1	Applied Physics	3-0	
	Total	15-1	

Semester - II

Code	Course Title	Credit Hours	Pre-requisite
CS 1x3	Digital Logic Design	3-1	Applied Physics
CS 1x4	Object Oriented Programming	3-1	Programming Fundamentals
HU 1x2	Communication & Presentation Skills	3-0	English Composition & Comprehension
MT 1x2	Probability & Statistics	3-0	
UE 1x1	University Elective – 1	3-0	
	Total	15-2	

Semester - III

Code	Course Title	Credit Hours	Pre-requisite
CS 2x1	Comp Organization & Assembly Lang.	3-1	
CS 2x2	Data Structures & Algorithms	3-1	Object-Oriented Programming
CS 2x3	Discrete Structures	3-0	
HU 2x1	Professional Practices	3-0	
SC 2x1	CS Supporting – 1	3-0	
	Total	15-2	

Semester - IV

Code	Course Title	Credit Hours	Pre-requisite
CS 2x4	Design & Analysis of Algorithms	3-0	Data Structures & Algorithms
CS 2x5	Theory of Automata	3-0	
CS 2x6	Database Systems	3-1	Data Structures & Algorithms
MT 2x1	Linear Algebra	3-0	
UE 2x1	University Elective – 2	3-0	
	Total	15-1	

Semester - V

Code	Course Title	Credit Hours	Pre-requisite
CS 3x1	Compiler Construction	3-0	Theory of Automata
SC 3x1	CS Supporting – 2	3-0	
CS 3x2	Operating Systems	3-1	Data Structures and Algorithms
CS 3x3	Software Engineering	3-0	
SC 3x2	CS Supporting – 3	3-0	
	Total	15-1	

Semester - VI

Code	Course Title	Credit Hours	Pre-requisite
CS 3x4	Artificial Intelligence	3-1	Discrete Structures
CS 3x5	Computer Networks	3-1	
CS 3x6	CS Elective – 1	3-0	
CS 3x7	CS Elective – 2	3-0	
HU 3x1	Technical & Business Writing	3-0	
	Total	15-2	

Semester - VII

Code	Course Title	Credit Hours	Pre-requisite
CS 4x1	CS Elective – 3	3-0	
CS 4x2	CS Elective – 4	3-0	
CS 4x3	Final Year Project – I	0-3	
UE 4x1	University Elective – 3	3-0	
CS 4x4	Parallel & Distributed Computing	3-0	Operating Systems
HU 4x1	Pakistan Studies	2-0	
	Total	14-3	

Semester - VIII

Code	Course Title	Credit Hours	Pre-requisite
CS 4x5	CS Elective – 5	3-0	
UE 4x2	University Elective – 4	3-0	
CS 4x6	Final Year Project – II	0-3	
CS 4x7	Information Security	3-0	
HU 4x2	Islamic Studies/ Ethics	2-0	
	Total	11-3	

(Universities may use their own course coding scheme)

BS Information Technology

Proposed Curriculum for BS-IT

Table 1.2: Areas Covered in BS programs

Course Group	**Credit hours**	**% age**
General Education	19	15%
University Electives	12	9%
Mathematics & Science Foundation	12	9%
Computing – Core	39	30%
Common courses	**82**	**63%**
Domain IT		
Domain IT Core	24	18%
Domain IT Electives	15	12%
Domain IT Supporting	9	7%
Domain courses	**48**	**37%**
	130	**100%**
TOTAL		

COURSES COMMON to all computing bachelor programs – 82 Credits

Computing Core Courses

Course Title	**Credit hours**
Programming Fundamentals	3-1
Object Oriented Programming	3-1
Data Structures & Algorithms	3-1
Discrete Structures	3-0
Operating Systems	3-1
Database Systems	3-1
Software Engineering	3-0
Computer Networks	3-1
Information Security	3-0
Final Year Project	0-6
Total	**39 (27-12)**

General Education Courses

Course Title	**Credit hours**
English Composition & Comprehension	3-0
Technical & Business Writing	3-0
Communication & Presentation Skills	3-0
Professional Practices	3-0
Intro to Info. & Comm. Technologies	2-1
Pakistan Studies	2-0
Islamic Studies/ Ethics	2-0
Total	**19**-0

University Elective Courses

(Not limited to the list below, Institutions may add more courses)

Course Title	Credit hours
Foreign Language	2-0
Social Service	1-0
Management Related	3-0
Social Science Related	3-0
Economy Related	3-0
Total	**12-**0

Mathematics and Science Foundation Courses

Course Title	Credit hours
Calculus & Analytical Geometry	3-0
Probability & Statistics	3-0
Linear Algebra	3-0
Applied Physics	3-0
Total	**12-0**

Domain Courses for BS-IT

BS-IT CORE (Compulsory) courses

Course Title	Credit hours
Cyber Security	3-0
Database Administration and Management	3-1
Information Technology Project Management	3-0
Information Technology Infrastructure	3-0
System and Network Administration	3-1
Virtual Systems and Services	3-1
Web Technologies	3-0
Total	**24 (21-3)**

BS-IT SUPPORTING courses (ANY 3 from the following list)

(Coverage of relevant pre-requisite must be ensured while offering any of the following courses from this category)

Course Title	Credit hours
Enterprise Systems	3-0
Modeling and Simulation	3-0
Formal Methods	3-0
Operations Research	3-0
Software Requirements Engineering	3-0
Total (Any three of the above)	**9-**0

BS-IT ELECTIVE courses

Course Title	Credit hours
IT Elective – 1	3
IT Elective – 2	3
IT Elective – 3	3
IT Elective – 4	3
IT Elective – 5	3
Total	**15**

Proposed Study Plan for BS (Information Technology)

4-Year Program (8 Regular Semesters of 18 weeks each)

Course Codes have been assigned as an example only. The purpose is to indicate pre-requisite courses for studying advanced courses.

Semester - I

Code	Course Title	Credit Hours	Pre-requisite
CS 1x1	Introduction to ICT	2-1	
CS 1x2	Programming Fundamentals	3-1	
HU 1x1	English Composition & Comprehension	3-0	
MT 1x1	Calculus & Analytical Geometry	3-0	
NS 1x1	Applied Physics	3-0	
	Total	14-2	

Semester - II

Code	Course Title	Credit Hours	Pre-requisite
CS 1x4	Object Oriented Programming	3-1	Programming Fundamentals
HU 1x2	Communication & Presentation Skills	3-0	
IT xxx	IT Supporting Course – I	3-0	
MT 1x2	Probability & Statistics	3-0	
UE 1xx	University Elective – 1	3-0	
UE 1xx	University Elective – 2	3-0	
	Total	18-1	

Semester - III

Code	Course Title	Credit Hours	Pre-requisite
CS 2x2	Data Structures & Algorithms	3-1	Object-Oriented Programming
CS 2x3	Discrete Structures	3-0	
HU 2x1	Professional Practices	3-0	
IT xxx	IT Supporting Course – II	3-0	
MT 2x2	Linear Algebra	3-0	
	Total	15-1	

Semester - IV

Code	Course Title	Credit Hours	Pre-requisite
CS 2x5	Operating Systems	3-1	Data Structures & Algorithms
CS 4x4	Information Security	3-0	
CS 3x5	Computer Networks	3-1	
IT 1xx	IT Project Management	3-0	
UE 2xx	University Elective – 3	3-0	
	Total	15-2	

Semester - V

Code	Course Title	Credit Hours	Pre-requisite
CS 3x2	Database Systems	3-1	Data Structures & Algorithms
CS 3x3	Software Engineering	3-0	
IT xxx	IT Supporting Course – III	3-0	
IT 2xx	System and Network Administration	3-1	Operating Systems
UE 3xx	University Elective – 4	3-0	
	Total	15-2	

Semester - VI

Code	Course Title	Credit Hours	Pre-requisite
IT 3xx	Web Technologies	3-0	
IT xxx	IT Elective – 1	3-0	
IT xxx	IT Elective – 2	3-0	
IT 4xx	IT Infrastructure	3-0	
HU 3x1	Technical & Business Writing	3-0	
	Total	15-0	

Semester - VII

Code	Course Title	Credit Hours	Pre-requisite
IT 5xx	Virtual Systems and Services	3-1	
IT 4x1	Final Year Project – I	0-3	
IT 4xx	IT Elective – 3	3-0	
IT 4xx	IT Elective – 4	3-0	
HU 1x3	Pakistan Studies	2-0	
	Total	11-4	

Semester - VIII

Code	Course Title	Credit Hours	Pre-requisite
IT 4x2	Final Year Project – II	0-3	
IT 6xx	Cyber Security	3-0	
IT 4xx	IT Elective – 5	3-0	
IT 7xx	Database Administration and Management	3-1	
HU 1x4	Islamic Studies/ Ethics	2-0	
	Total	11-4	

BS Software Engineering

BS Software Engineering

Software plays a central and underpinning role in almost all aspects of daily life: communications, government, manufacturing, banking and finance, education, transportation, entertainment, medicine, agriculture, and law. The number, size, and application domains of computer programs have grown dramatically; as a result, huge sums are being spent on software development. Most people's lives and livelihoods depend on this development's effectiveness. Software products help us to be more efficient and productive. They provide information, make us more effective problem solvers, and provide us with safer, more flexible, and less confining work, entertainment, and recreation environments.

Software Engineering is the application of a systematic, disciplined and quantifiable approach to the design, development, operation, and maintenance of software systems. It is in fact the practice of designing and implementing large, reliable, efficient and economical software by applying the principles and practices of engineering. The department aims to train students in all aspects of software life cycle from specification through analysis and design to testing, maintenance and evolution of software product.

Program Learning Outcomes (PLOs)

Program learning outcomes are the narrower statements that describe what students are expected to know and be able to do by the time of graduation. These relate to the knowledge, skills and attitude that the students acquire while progressing through the program.

The program must demonstrate that by the time of graduation the students have attained a certain set of knowledge, skills and behavioral traits, at least to some acceptable minimum level. Specifically, it is to be demonstrated that the students have acquired the following graduate attributes (GAs)

GA1 **Engineering Knowledge:** An ability to apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

GA2 **Problem Analysis:** An ability to identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

GA3 **Design/Development of Solutions:** An ability to design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

GA4 **Investigation:** An ability to investigate complex engineering problems in a methodical way including literature survey, design and conduct of experiments, analysis and interpretation of experimental data, and synthesis of information to derive valid conclusions.

GA5 **Modern Tool Usage:** An ability to create, select and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling, to complex engineering activities, with an understanding of the limitations.

GA6 **The Engineer and Society:** An ability to apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice and solution to complex engineering problems.

GA7 **Environment and Sustainability:** An ability to understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

GA8 **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

GA9 **Individual and Team Work:** An ability to work effectively, as an individual or in a team, on multifaceted and /or multidisciplinary settings.

GA10 **Communication:** An ability to communicate effectively, orally as well as in writing, on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

GA11 **Project Management:** An ability to demonstrate management skills and apply engineering principles to one's own work, as a member and/or leader in a team, to manage projects in a multidisciplinary environment.

GA12 **Lifelong Learning:** An ability to recognize importance of, and pursue lifelong learning in the broader context of innovation and technological developments.

Proposed Curriculum for BS-SE

Table 1.2: Areas Covered in BS programs

Course Group	**Credit hours**	**% age**
General Education	19	15%
University Electives	12	9%
Mathematics & Science Foundation	12	9%
Computing – Core	39	30%
Common courses	**82**	**63%**
Domain SE		
Domain SE Core	24	18%
Domain SE Electives	15	12%
Domain SE Supporting	9	7%
Domain courses	**48**	**37%**
TOTAL	**130**	**100%**

COURSES COMMON to all computing bachelor programs – 82 Credits

Computing Core Courses

Course Title	**Credit hours**
Programming Fundamentals	3-1
Object Oriented Programming	3-1
Data Structures & Algorithms	3-1
Discrete Structures	3-0
Operating Systems	3-1
Database Systems	3-1
Software Engineering	3-0
Computer Networks	3-1
Information Security	3-0
Final Year Project	0-6
Total	**39 (27-12)**

General Education Courses

Course Title	**Credit hours**
English Composition & Comprehension	3-0
Technical & Business Writing	3-0
Communication & Presentation Skills	3-0
Professional Practices	3-0
Intro to Info. & Comm. Technologies	2-1
Pakistan Studies	2-0
Islamic Studies/ Ethics	2-0
Total	**18-1**

University Elective Courses

(Not limited to the list below, Institutions may add more courses)

Course Title	Credit hours
Economy Related	3-0
Foreign Language	2-0
Management Related	3-0
Social Science Related	3-0
Social Service	1-0
Total	**12**-0

Mathematics and Science Foundation Courses

Course Title	Credit hours
Applied Physics	3-0
Calculus & Analytical Geometry	3-0
Linear Algebra	3-0
Probability & Statistics	3-0
Total	**12-0**

Domain Courses for BS-SE

Software Engineering CORE (Compulsory) courses

Course Title	Credit hours
Human Computer Interaction	3-0
Software Construction & Development	2-1
Software Design & Architecture	2-1
Software Project Management	3-0
Software Quality Engineering	3-0
Software Re-Engineering	3-0
Software Requirements Engineering	3-0
Web Engineering	3-0
Total	**22-2**

Software Engineering SUPPORTING courses

Course Title	Credit hours
Business Process Engineering	3-0
Formal Methods in Software Engineering	3-0
Operations Research	3-0
Simulation and Modeling	3-0
Stochastic Processes	3-0
Total (Any THREE of the above)	**9**-0

Software Engineering ELECTIVE courses

(Select any FIVE courses from the following list)
(The list is by no means exhaustive. Institutions may add new courses)

Course Title	Credit hours
Agent Based Software Engineering	3-0
Big Data Analytics	3-0
Cloud Computing	3-0
Computer Graphics	3-0
Data Encryption and Security	3-0
E-Commerce	3-0
Game Application Development	3-0
Global Software Development	3-0
Information Systems Audit	3-0
Management Information Systems	3-0
Mobile Application Development	3-0
Multimedia Communication	3-0
Natural Language Processing	3-0
Real Time Systems	3-0
Semantic Web	3-0
Software Engineering Economics	3-0
Software Metrics	3-0
Systems Programming	3-0
Topics in Software Engineering	3-0
Visual Programming	3-0
Total (Any FIVE of the Above)	**15**-0

Proposed Study Plan for BS (Software Engineering)

4-Year Program (8 Regular Semesters of 18 weeks each)

Course Codes have been assigned as an example only. The purpose is to indicate pre-requisite courses for studying advanced courses.

Semester - I

Code	Course Title	Credit Hours	Pre-requisite
	Introduction to Info. & Comm. Technologies	2-1	
	Programming Fundamentals	3-1	
	English Composition & Comprehension	3-0	
	Calculus & Analytical Geometry	3-0	
	Pakistan Studies	2-0	
	Applied Physics	3-0	
	Total	16-2	

Semester - II

Code	Course Title	Credit Hours	Pre-requisite
	Object Oriented Programming	3-1	Programming Fundamentals
	Communication & Presentation Skills	3-0	English Composition and Comprehension
	Discrete Structures	3-0	
	Software Engineering	3-0	
	Islamic Studies	2-0	
	University Elective - I	3-0	
	Total	17-1	

Semester - III

Code	Course Title	Credit Hours	Pre-requisite
	Data Structures & Algorithms	3-1	Object Oriented Programming
	Software Requirement Engineering	3-0	Software Engineering
	Human Computer Interaction	3-0	Software Engineering
	Linear Algebra	3-0	
	University Elective-II	3-0	
	Total	15-1	

Semester - IV

Code	Course Title	Credit Hours	Pre-requisite
	Operating Systems	3-1	Data Structures & Algorithms
	Database Systems	3-1	Data Structures & Algorithms
	Software Design & Architecture	2-1	Software Requirement Engineering
	Probability and Statistics	3-0	
	University Elective – III	3-0	
	Total	14-3	

Semester - V

Code	Course Title	Credit Hours	Pre-requisite
	Software Construction and Development	2-1	Software Design and Architecture
	Computer Networks	3-1	
	Technical and Business Writing	3-0	Communication & Presentation Skills
	SE Supporting –I	3-0	
	SE Supporting - II	3-0	
	Total	14-2	

Semester - VI

Code	Course Title	Credit Hours	Pre-requisite
	Software Quality Engineering	3-0	Software Engineering
	Information Security	3-0	
	Professional Practice	3-0	
	Web Engineering	3-0	
	SE Elective – I	3-0	
	SE Supporting - III	3-0	
	Total	18-0	

Semester - VII

Code	Course Title	Credit Hours	Pre-requisite
	Software Project Management	3-0	Software Engineering
	Software Re-Engineering	3-0	Software Construction & Development
	SE Elective -II	3-0	
	SE Elective - III	3-0	
	Final Year Project - I	0-3	
	Total	12-3	

Semester - VIII

Code	Course Title	Credit Hours	Pre-requisite
	SE Elective – IV	3-0	
	SE Elective – V	3-0	
	Final Year Project - II	0-3	
	University Elective - IV	3-0	
	Total	9-3	

Master of Science Programs

Curriculum for Master Degrees in Computing

Introduction

Computing is emerging as a very important and inevitable tool in modern daily life and businesses.

Master Degree Programs in Computing

Computer Science	(MS-CS)
Data Science	(MS-DS)
Information Security	(MS-IS)
Information Technology	(MS-IT)
Software Engineering	(MS-SE)
Software Project Management	(MS-SPM)

Eligibility Criteria

The minimum requirements for admission in a Master degree program are

a. A degree earned after sixteen years of education in computing or a related discipline, AND
b. At least CGPA of 2.0 (on a scale of 4.0) or 60% Marks

Duration

The **minimum duration** for completion of MS degree is two years. The HEC allows a **maximum period of four years** to complete MS degree requirements.

Degree Completion Requirements

To become eligible for award of MS degree, a student must satisfy the following requirements:

c) Must have studied and passed the **prescribed courses, totaling at least 30 credit hours**.
d) Must have earned **CGPA** (Cumulative Grade Point Average) **of at least 2.5 on a scale of 4.0**.

MS Computer Science

Program Objectives:

The MS (Computer Science) comprises of both course work as well as research component. There are four ‘core courses’ aimed at strengthening the understanding and competence of students in computer science fundamentals. The University expects its MS graduates to pursue careers either as ‘Computer Science Faculty Members’ or as ‘Software Development Managers’ in the industry.

Learning Outcomes:

1. Students will be able to possess advanced knowledge of Computer Science field
2. Students will be able to think creatively and critically; to solve non-trivial problems
3. Students will be able to use computing knowledge to develop efficient solutions for real life problems
4. Students will be able to design solutions and can conduct research related activities

Eligibility:

Degree in relevant subject, earned from a recognized university after 16 years of education with at least 60% marks or CGPA of at least 2.0 (on a scale of 4.0).

The following core courses are recommended to be completed before entering the MS (CS) program.

1. Analysis of Algorithms
2. Assembly Lang. / Computer Architecture
3. Computer Networks
4. Computer Programming
5. Data Structures
6. Database Systems
7. Operating Systems
8. Software Engineering
9. Theory of Automata

A student selected for admission having deficiency in the above stated courses may be required to study a maximum of FOUR courses, which must be passed in the first two semesters. Deficiency courses shall be determined by the Graduate Studies Committee, before admitting the student.

A student cannot register in MS courses, unless all specified deficiency courses have been passed.

A student has the option to pursue MS by undertaking either a 6 credit hour MS Thesis OR a three credit hour taught course and a three credit-hour MS Project.

Tentative Study Plan of MS (Computer Science)

Semester – I

CS 5xx Core Course – I	3
CS 5xx Core Course – II	3
CS 5xx Core Course – III	3
Total	9

Semester – II

CS 5xx Core Course – IV	3
CS 5xx Elective – I	3
CS 5xx Elective – II	3
SS 3xx Research Methodology	1
Total	10

Semester – III

CS 5xx Elective – III	3
CS 5xx MS Thesis-I	3
Total	6

Semester – IV

CS 5xx Elective-IV	3
CS 5xx MS Thesis-II	3
Total	6

Registration in “MS Thesis - I” is allowed provided the student has

a. Earned at least 18 credits
b. Passed the “Research Methodology” course; AND
c. CGPA is equal to or more than 2.5.

Core Courses for MS (Computer Science)

At least four courses must be taken from the following

CS501 Advanced Analysis of Algorithms
CS505 Advanced Operating Systems
CS507 Theory of Programming Languages
CS534 Theory of Automata – II
EE502 Advanced Computer Architecture

Award of Degree

For award of MS degree, a student must have:

a. Passed courses totaling at least 30 credit hours, including four core courses.
b. Obtained a CGPA of 2.5 or more.

MS Data Science

Curriculum for MS Data Science (MS DS)

Program Overview:

The MS (DS) program has been designed to give students the option to be part of a data science endeavor that begins with the identification of business processes, determination of data provenance and data ownership, understanding the ecosystem of the business decisions, skill sets and tools that shape the data, making data amenable to analytics, identifying sub-problems, recognizing the technology matrix required for problem resolution, creating incrementally-complex data-driven models and then maintaining them to ultimately leverage them for business growth.
Individual objectives include:

- To equip students to transform data into actionable insights to make complex business decisions.
- To enable students, understand and analyze a problem and arrive at computable solutions.
- To expose students to the set of technologies that match those solutions.
- To gain hands-on experience on data-centric tools for statistical analysis, visualization and big data applications at the same rigorous scale as in a practical data science project.
- To understand the implications of handling data in terms of data security and business ethics.

Program Scope:

The amount of data is growing so rapidly and their significance in the emerging societal set ups such as the pervasive Internet of Things. The way one imagines data is going to change in the coming years. Both Big Data Analytics and pervasive computing hinge on the principle axis of data analytics. MS (DS) program is going to be relevant in terms of job creation and artisanal smart business generation. Graduates from this program would definitely avail the early-bird advantage.

Eligibility criteria:

A degree of BS (CS) as per HEC curriculum. Students with 16 years of education in following domains (Information Technology, Software Engineering, Computer Engineering, Electrical Engineering, Statistics, or Mathematics) are eligible to apply provided that they have taken following deficiency courses.

Deficiency Courses:

1. Programming Fundamentals (Core Programming Course)
2. Data Structures & Algorithms **OR** Design & Analysis of Algorithms
3. Database Systems

Outline of the MS (DS) program:

The program would be spread over 4 semesters, with a 6-credit hour thesis being offered in the second year.

Course offering plan:

Course types	Cumulative Credits
Program Core courses (3)	9
Specialization Requirement Courses (2)	6
Electives (3)	9
Thesis	6

Proposed 3 core courses:

1. Statistical and Mathematical Methods For Data Science (3)
2. Tools and Techniques in Data Science (2+1)
3. Machine Learning (3)

Proposed 2 Specialization Core Courses **(Choose any 2)**

1. Big Data Analytics (3)
2. Deep Learning (3)
3. Natural Language Processing (3)
4. Distributed Data Processing (3)

Semester-wise course offering plan:

Semester 1	Course Title	Credits
	Tools and Techniques for Data Science	2+1[1]
	Statistical and Mathematical Methods for Data Analysis	3
	Elective-I	
Semester 2	Course Title	Credits
	Machine Learning	3
	Specialization-Elective-I	3
	Specialization Elective-II	3

[1] 2+1 means 2 hours of Lecture + 3 hours of Lab work.

Semester 3	Course Title	Credits
	Elective II	3
	MS-Thesis-I	3
Semester 4	Course Title	Credits
	Elective III	3
	MS-Thesis-II	3

Thesis:

According to the current rules of HEC, a thesis would enable students to have their degree vetted equivalent to an M.Phil. degree.

Elective courses:

Following is a non-exhaustive list of elective courses. New elective courses may be added to this list. Students may be recommended to make their choice of electives, in the light of a soft specialization within the field of data science.

- Advanced Computer Vision
- Algorithmic trading
- Bayesian Data Analysis
- Big Data Analytics
- Bioinformatics
- Cloud computing
- Computational Genomics
- Data Visualization
- Deep Learning
- Deep Reinforcement Learning
- Distributed Data Processing and Machine Learning
- Distributed Machine Learning in Apache Spark
- High performance computing
- Inference & Representation
- Natural Language Processing
- Optimization Methods for Data Science and Machine Learning
- Probabilistic Graphical Models
- Scientific Computing in Finance
- Social network analysis
- Time series Analysis and Prediction

MS Information Security

MS Information Security

The committee thoroughly discussed each and every aspect of the curriculum in the light of foreign universities courses objectives, course outline and current requirements of the industry. The complete detail regarding proposed MS (Information Security) is as follows:

Minimum credit hours = 30

The Program comprises four semesters spread over two years; with two semesters in a year. The additional major areas may be added in the list of specialization as appropriate to university keeping in view the resources available.

Credit Hours Distribution

Core courses credit hours	6
Electives courses credit hours	18
Thesis credit hours	6
Total Credit Hours	30

Program Objectives:

Today’s world rely on the Internet to conduct business and share information with their employees and customers in real time. With this reliance, however, comes an increased risk for information security breaches and critical business disruption. Now more than ever, organizations are looking to information security professionals who understand the complexity of today’s information technology infrastructures, the effect of technology on business objectives, and the importance of recognizing and managing risk to design and implement their information security and assurance strategies.

A challenging graduate program is structured on the basis of the classical objective, which is the preparation for study of doctoral level, and this remains an important aspect of such program, but it is believed that all programs should prepare the student for study beyond the master’s level.

The program aims to develop core competencies in the fields of information security. Students will have the opportunity of learning the technical aspects of information security by understanding current threats and vulnerabilities and examining ways of developing effective countermeasures. In order to cater for wide range of professional and academic interests, students have the option of selecting their course work according to their specific needs.

The main aim of this program is to fulfill the growing national need of well trained professionals to work in a wide range of roles to protect Information Systems in all types of organizations, including research and academia. Moreover, program is aimed to produce skilled people who are able to contribute towards need of protecting national information infrastructure from all kinds of threats. Skilled persons will also be able to play an effective role in international efforts to make the cyberspace safe, secure and reliable for the national and international community.

Program Structure:

The graduate program should embody sufficient flexibility to fulfill the requirements of either an “academic” degree (Breadth-Based) obtained in preparation for further graduate study or a terminal “professional” degree (Depth-Based). The discipline of Computer Information Security has matured enough that the distinction between academic and professional program is beginning to appear. However, the concept of an utterly terminal program is not widely accepted in the field. All academic programs should provide the possibility of additional study in the field. The proposed program is intended to establish an integrated breadth and depth based curriculum model to assure that the common aspects of various potential areas in Information Security are captured.

The proposed curriculum structure be implemented within four-semester time. A thesis work may be unified with student’s chosen depth oriented specialties. Generally graduate program are structured with a common core of fundamental material and wide range of options for the rest of the course work.

Eligibility:

Sixteen years of education with CGPA of at least 2.0 (on a scale of 4.0) or equivalent in science/engineering discipline preferably with 4 years degree program of BS (SE/CS/IT/EE/CE) or equivalent from HEC recognized university or degree awarding institute.

Two years of relevant work experience may be preferred.

Core Subjects:

- Advanced Analysis of Algorithms
- Information Privacy and Security
- Cryptography

MS (Information Security) Semester-wise Model Program

Semester 1

Course Code	Course Title	Credit Hours
	Advanced Analysis of Algorithms (Core Course)	3
	Information Privacy and Security(Core Course)	3
	Cryptography(Core Course)	3
	Total	**9**

Semester 2

Course Code	Course Title	Credit Hours
	*Research Methods (OR University Elective I)	3
	Elective II	3
	Elective III	3
	Total	**9**

Semester 3

Course Code	Course Title	Credit Hours
	Elective IV	3
	Thesis-I	3
	Total	**6**

Semester 4

Course Code	Course Title	Credit Hours
	Elective V	3
	Thesis-II	3
	Total	**6**
	TOTAL CREDIT HOURS	**30**

*Research Methods course should be compulsory for those students who will go for Research Thesis.

List of Elective Courses:

Below are the proposed elective courses but it does not restrict below list, it may be gone beyond this list.

- Advanced Cryptography
- Analysis of Stochastic Systems
- Applied Cryptography
- Cloud Computing Security
- Cognitive Security
- Computer Forensics
- Computer Security
- Critical Infrastructure Protection and Incident Management
- Cryptanalysis
- Cryptography and Security Protocol
- Cyber Intelligence
- Cybercrime Investigation
- Data Communication Networks & Security
- Digital Forensics and Incident Response
- Electronic Warfare – Principles and Techniques

- Forensic Tool Development
- Forensics: Open Source, Windows, Apple-device
- Information Hiding
- Information Risk Management
- Information Security Management
- Information Security Policy Development
- Information Security Project Management
- Information Technology Forensics and Investigations
- Intrusion Detection in Physical and Virtual Networks
- Intrusion Detection System
- IT Security Evaluation & Auditing
- Legal Issues in Information Security
- Mobile Security
- Multimedia Security and Information Hiding
- Network Forensics
- Network Security
- OS & File System Forensics
- Principles of Incident Response and Disaster Recovery
- Quantum Computing and Information security
- Quantum Cryptography
- Reverse Engineering and Malware Analysis
- Securing Applications, Web Services, and Software as a Service (SAAS)
- Security Audit & Assessment
- Socio-technical Systems Enabled Crime
- Software Security Testing and Code Assessment
- Steganography
- Systems / Network Security
- Vulnerability Exploitation and Defense
- Wireless Network Security

MS Information Technology

The Program comprise four semesters spread over two years, with two semesters in a year. The additional major areas may be added in the list of specialization as appropriate to university keeping in view the resources available.

Credit Hours Distribution:

Core courses credit hours	6
Electives courses credit hours	18
Thesis credit hours	6
Total Credit Hours	30

Program Objectives:

Today's world rely on the Internet to conduct business and share information with their employees and customers in real time. With this reliance, however, comes an increased risk for information security breaches and critical business disruption. Now more than ever, organizations are looking to information security professionals who understand the complexity of today's information technology infrastructures, the effect of technology on business objectives, and the importance of recognizing and managing risk to design and implement their information security and assurance strategies.

A challenging graduate program is structured on the basis of the classical objective, which is the preparation for study of doctoral level, and this remains an important aspect of such program, but it is believed that all programs should prepare the student for study beyond the master's level.

The program aims to develop core competencies in the fields of information security. Students will have the opportunity of learning the technical aspects of information security by understanding current threats and vulnerabilities and examining ways of developing effective countermeasures. In order to cater for wide range of professional and academic interests, students have the option of selecting their course work according to their specific needs.

The main aim of this program is to fulfill the growing national need of well trained professionals to work in a wide range of roles to protect Information Systems in all types of organizations, including research and academia. Moreover, program is aimed to produce skilled people who are able to contribute towards need of protecting national information infrastructure from all kinds of threats. Skilled persons will also be able to play an effective role in international efforts to make the cyberspace safe, secure and reliable for the national and international community.

Program Structure:

The graduate program should embody sufficient flexibility to fulfill the requirements of either an "academic" degree (Breadth-Based) obtained in preparation for further graduate study or a terminal "professional" degree (Depth-Based). The discipline of Computer Information Security has matured enough that the distinction between academic and professional program is beginning to appear. However, the concept of an utterly terminal program is not widely accepted in the field. All academic programs should provide the possibility of additional study in the field. The proposed program is intended to establish an integrated breadth and depth based curriculum model to assure that the common aspects of various potential areas in Information Security are captured.

The proposed curriculum structure be implemented within four-semester time. A thesis work may be unified with student's chosen depth oriented specialties. Generally graduate program are structured with a common core of fundamental material and wide range of options for the rest of the course work.

Eligibility:

Sixteen years of education with CGPA of at least 2.0 (on scale of 4.0) or equivalent in science/engineering discipline preferably with 4 years degree program of BS (SE/CS/IT/EE/CE) or equivalent from HEC recognized university or degree awarding institute.

Two years of relevant work experience may be preferred.

MS (Information Security) Semester-wise Model Program

Semester 1

Course Code	Course Title	Credit Hours
	Advanced Analysis of Algorithms (Core Course)	03
	Information Privacy and Security(Core Course)	03
	Cryptography(Core Course)	03
	Total	**09**

Semester 2

Course Code	Course Title	Credit Hours
	*Research Methods (OR University Elective I)	03
	Elective II	03
	Elective III	03
	Total	**09**

Semester 3

Course Code	Course Title	Credit Hours
	Elective IV	03
	Thesis-I	03
	Total	**06**

Semester 4

Course Code	Course Title	Credit Hours
	Elective V	03
	Thesis-II	03
	Total	06
	TOTAL CREDIT HOURS	**30**

*Research Methods course should be compulsory for those students who will go for Research Thesis.

Core Subjects:

- Advanced Analysis of Algorithms
- Information Privacy and Security
- Cryptography

List of Elective Courses:

Below are the proposed elective courses but it does not restrict below list, it may be gone beyond this list.

- Analysis of Stochastic Systems
- Applied Cryptography
- Business Continuity Planning
- Cloud Computing Security
- Cognitive Security
- Computer Forensics
- Computer Security
- Critical Infrastructure Protection and Incident Management
- Data Communication Networks & Security
- Digital Forensics and Incident Response
- Forensics: Open Source, Windows, Apple-device
- Information Hiding
- Information Risk Management
- Information Security Management
- Information Security Policy Development
- Information Security Project Management
- Information Technology Forensics and Investigations
- Intrusion Detection in Physical and Virtual Networks
- Intrusion Detection System
- IT Security Evaluation & Auditing
- Legal Issues in Information Security
- Mobile Security
- Multimedia Security and Information Hiding
- Network Forensics
- Network Security
- OS & File System Forensics
- Principles of Incident Response and Disaster Recovery
- Quantum Computing and Information security
- Quantum Cryptography
- Research Methodology
- Reverse Engineering and Malware Analysis
- Securing Applications, Web Services, and Software as a Service (SAAS)
- Security Audit & Assessment
- Socio-technical Systems Enabled Crime
- Software Security Testing and Code Assessment
- Steganography
- Systems / Network Security
- Vulnerability Exploitation and Defense
- Wireless Network Security

MS Software Engineering

Curriculum for MS Software Engineering, MS (SE)

Mission Statement

The mission of the Masters of Science (Software Engineering) program is to equip students with theoretical and applied knowledge of software for the solution of complex problems. It is aimed to prepare the students to learn independently in a constantly changing discipline.

Program Objectives

The objectives of MS (Software Engineering) program are:

1. Prepare students who can critically apply concepts, theories and practices to provide creative solutions of complex computing problems.
2. Prepare students who can define, plan, implement and test a medium-sized software project using appropriate software engineering processes, methods and techniques.
3. Prepare students to effectively communicate their ideas in written and electronic form, and prepare them to work collaboratively in a team environment.
4. Prepare students with a theoretical software engineering background and applied research needed to enter a doctorate program in software engineering.
5. Prepare students to join an appropriate and respectable level position in a computing-related field, and to maintain their professional skills in rapidly evolving field.

Eligibility Criteria:

The minimum requirements for admission in a Master degree program are:

Sixteen years education in a relevant subject with a minimum CGPA of 2.0 (on a scale of 4.0).

Note:

The university may recommend deficiency courses, after considering the educational background and knowledge of the candidate.

Duration

M**inimum duration** for completion of MS degree is two years. HEC allows a **maximum period of four years** to complete MS degree requirements.

Degree Completion Requirements

To become eligible for award of MS degree, a student must satisfy the following requirements:

a) Must have earned **CGPA** (Cumulative Grade Point Average) **of at least 2.5 on a scale of 4.0**.

b)

- Must have studied and passed the **prescribed courses, totaling at least 30 credit hours**.

OR

- Must have studied and passed the **24 credit hours** of courses from the prescribed course list and successfully completed **6 credit hours** of Thesis/Research Work.

Suggested Curriculum for MS-SE

Core Courses

Course Title	Credit hours
Advanced Requirements Engineering	3
Advanced Software System Architecture	3
Software Testing and Quality Assurance	3
Total	**9 (9-0)**

Domain Elective Courses

(Not limited to the list below, Institutions may add more courses)

Course Title	Credit hours
Software Measurement and Metrics	3
Component Based Software Engineering	3
Advanced Formal Methods	3
Advanced Human-Computer Interaction	3
Agile Software Development Methods	3
Empirical Software Engineering	3
Advanced Software Project Management	3
Total (Any 2 of the above for thesis option **OR** any 2-4 courses for non-thesis option**)**	

General Elective Courses

(Not limited to the list below, Institutions may add more courses)

Course Title	Credit hours
Software Risk Management	3
Research Methodology	3
Software Measurement and Metrics	3
Software Configuration Management	3
Reliability Engineering	3
Complex Networks	3
Agent Based Modeling	3
Total (Any 3 of the above for thesis option **OR** any 3-5 courses for non-thesis option**)**	

Proposed Study Plan for
MS (Software Engineering)

2-Year Program (4 Regular Semesters of 18 weeks each)

Semester - I

Code	Course Title	Credit Hours	Pre-requisite
	Advanced Requirements Engineering	3	
	Advanced Software System Architecture	3	
	Elective I	3	
	Total	9-0	

Semester - II

Code	Course Title	Credit Hours	Pre-requisite
	Software Testing and Quality Assurance	3	
	Elective II	3	
	Elective III	3	
	Total	9-0	

Semester - III

Code	Course Title	Credit Hours	Pre-requisite
	Elective IV	3	
	Thesis I/ (Elective V)	3	
	Total	6-0	

Semester - IV

Code	Course Title	Credit Hours	Pre-requisite
	Elective VI	3	
	Thesis II / (Elective VII)	3	
	Total	6-0	

Total Program Credit Hours: 30

MS Software Project Management

The committee thoroughly discussed each and every aspect of the curriculum in the light of foreign universities courses objectives, course outline and current requirements of the industry. The complete detail regarding proposed MS (Information Security) is as follows:

Minimum credit hours = 30

The Program shall comprise 4 semesters spread over 2 years with two semesters in a year. The additional major areas may be added in the list of specialization as appropriate to university keeping in view the resources available.

Credit Hours Distribution:

Core courses credit hours	6
Electives courses credit hours	8
Thesis credit hours	6
Total Credit Hours	30

Program Objectives:

Today's world rely on the Internet to conduct business and share information with their employees and customers in real time. With this reliance, however, comes an increased risk for information security breaches and critical business disruption. Now more than ever, organizations are looking to information security professionals who understand the complexity of today's information technology infrastructures, the effect of technology on business objectives, and the importance of recognizing and managing risk to design and implement their information security and assurance strategies.

A challenging graduate program is structured on the basis of the classical objective, which is the preparation for study of doctoral level, and this remains an important aspect of such program, but it is believed that all programs should prepare the student for study beyond the master's level.

The program aims to develop core competencies in the fields of information security. Students will have the opportunity of learning the technical aspects of information security by understanding current threats and vulnerabilities and examining ways of developing effective countermeasures. In order to cater for wide range of professional and academic interests, students have the option of selecting their course work according to their specific needs.

The main aim of this program is to fulfill the growing national need of well trained professionals to work in a wide range of roles to protect Information Systems in all types of organizations, including research and academia. Moreover, program is aimed to produce skilled people who are able to contribute towards need of protecting national information infrastructure from all kinds of threats. Skilled persons will also be able to play an effective role in international efforts to make the cyberspace safe, secure and reliable for the national and international community.

Program Structure:

The graduate program should embody sufficient flexibility to fulfill the requirements of either an "academic" degree (Breadth-Based) obtained in preparation for further graduate study or a terminal “professional” degree (Depth-Based). The discipline of Computer Information Security has matured enough that the distinction between academic and professional program is beginning to appear. However, the concept of an utterly terminal program is not widely accepted in the field. All academic programs should provide the possibility of additional study in the field. The proposed program is intended to establish an integrated breadth and depth based curriculum model to assure that the common aspects of various potential areas in Information Security are captured.

The proposed curriculum structure be implemented within four-semester time. A thesis work may be unified with student’s chosen depth oriented specialties. Generally graduate program are structured with a common core of fundamental material and wide range of options for the rest of the course work.

Eligibility:

Sixteen years of education with CGPA of at least 2.0 (on scale of 4.0) or equivalent in science/engineering discipline preferably with 4 years degree program of BS (SE/CS/IT/EE/CE) or equivalent from HEC recognized university or degree awarding institute.

Two years of relevant work experience may be preferred.

MS (Information Security) Semester-wise Model Program

Semester 1

Course Code	Course Title	Credit Hours
	Advanced Analysis of Algorithms (Core Course)	03
	Information Privacy and Security(Core Course)	03
	Cryptography(Core Course)	03
	Total	**09**

Semester 2

Course Code	Course Title	Credit Hours
	*Research Methods (OR University Elective I)	03
	Elective II	03
	Elective III	03
	Total	**09**

Semester 3

Course Code	Course Title	Credit Hours
	Elective IV	03
	Thesis-I	03
	Total	**06**

Semester 4

Course Code	Course Title	Credit Hours
	Elective V	03
	Thesis-II	03
	Total	06
	TOTAL CREDIT HOURS	**30**

*Research Methods course should be compulsory for those students who will go for Research Thesis.

Core Subjects:

- Advanced Analysis of Algorithms
- Information Privacy and Security
- Cryptography

List of Elective Courses:

Below are the proposed elective courses but it does not restrict below list, it may be gone beyond this list.

- Advanced Cryptography
- Analysis of Stochastic Systems
- Applied Cryptography
- Business Continuity Planning
- Cloud Computing Security
- Cognitive Security
- Computer Forensics
- Computer Security
- Critical Infrastructure Protection and Incident Management
- Cryptanalysis
- Cryptography and Security Protocol
- Cyber Intelligence
- Cybercrime Investigation
- Data Communication, Networks & Security
- Digital Forensics and Incident Response
- Electronic Warfare – Principles and Techniques
- Forensic Tool Development
- Forensics: Open Source, Windows, Apple-device
- Information Hiding
- Information Risk Management
- Information Security Management
- Information Security Policy Development
- Information Security Project Management
- IT Forensics and Investigations
- Intrusion Detection in Physical and Virtual Networks
- Intrusion Detection System
- IT Security Evaluation & Auditing
- Legal Issues in Information Security
- Mobile Security
- Multimedia Security and Information Hiding
- Network Forensics
- Network Security
- OS & File System Forensics
- Principles of Incident Response and Disaster Recovery
- Quantum Computing and Information security
- Quantum Cryptography
- Reverse Engineering and Malware Analysis
- Securing Applications, Web Services, and Software as a Service (SAAS)
- Security Audit & Assessment
- Socio-technical Systems Enabled Crime
- Software Security Testing and Code Assessment
- Steganography
- Systems / Network Security
- Vulnerability Exploitation and Defense
- Wireless Network Security

NCRC Computing – 2017

BS Course Outlines

Bachelor Courses' List

Course Title	Page No.
1. Agent Based Software Engineering	
2. Applied Physics	
3. Artificial Intelligence	
4. Big Data Analytics	
5. Business Process Engineering	
6. Business Process Management	
7. Calculus & Analytical Geometry	
8. Communication & Presentation Skills	
9. Compiler Construction	
10. Computer Graphics	
11. Computer Networks	
12. Computer Organization & Assembly Language	
13. Computer Vision	
14. Cyber Security	
15. Data Encryption and Security	
16. Data Structures & Algorithms	
17. Database Administration and Management	
18. Database Systems	
19. Design & Analysis of Algorithms	
20. Differential Equations	
21. Digital Image Processing	
22. Digital Logic Design	
23. Discrete Structures	
24. E-Commerce	
25. English Composition & Comprehension	
26. Enterprise Systems	
27. Formal Methods in Software Engineering	
28. Global Software Development	
29. Graph Theory	
30. Human Computer Interaction	
31. Information Security	
32. Information Systems Audit	
33. Information Technology Project Management	
34. Intro. to Info. & Comm. Technologies	
35. Introduction to Software Engineering	
36. Islamic Studies/ Ethics	
37. IT Infrastructure	
38. Linear Algebra	
39. Logical Paradigms of Computing	
40. Management Information System	
41. Mobile Application Development	
42. Multimedia Communications	
43. Multi-variate Calculus	

44. Natural Language Processing	
45. Numerical Computing	
46. Object Oriented Analysis and Design	
47. Object Oriented Programming	
48. Operating Systems	
49. Operations Research	
50. Pakistan Studies	
51. Parallel & Distributed Computing	
52. Probability & Statistics	
53. Professional Practices	
54. Programming Fundamentals	
55. Real Time Systems	
56. Semantic Web	
57. Simulation and Modeling	
58. Software Construction & Development	
59. Software Design & Architecture	
60. Software Engineering Economics	
61. Software Metrics	
62. Software Project Management	
63. Software Quality Engineering	
64. Software Re-Engineering	
65. Software Requirements Engineering	
66. Stochastic Processes	
67. System and Network Administration	
68. Systems Programming	
69. Technical & Business Writing	
70. Theory of Automata	
71. Theory of Programming Languages	
72. Virtual Systems and Services	
73. Visual Programming	
74. Web Engineering	
75. Web Technologies	

DETAIL OF COURSES

Agent Based Software Engineering			
Credit Hours:	3(3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the agent system terminology and development process of agent-based systems.	C	2
2. Understand the techniques to design agent-based system.	C	2
3. Understand how to modify architecture of the current software systems and restructure them to be agent-based	C	2
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Overview of agent-based software engineering. Methodologies for agent-based modeling, analysis and design: Agent-based Unified Modeling Language (AUML), Agent-based analysis and design, Other agent-based analysis and design methods. Agent communication and knowledge sharing: knowledge level communication among software agents, Knowledge Interchange Format (KIF), Agent-based System Architecture and Organization. FIPA: Foundation for Intelligent Physical Agents: FIPA specification, the application, abstract architecture, agent communication, agent management and agent message transport standards and guidelines.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Multi-agent Systems: A Modern Approach to Distributed Artificial Intelligence, Gerhard Weiss, Edt., 1st edition, MIT Press, 2000.
2. Agent-Oriented Methodologies, Paolo Giorgini, Idea Group Publishing, 2005.
3. Agent-Oriented Software Engineering III, Fausto Giunchiglia, James J. Odell, Gerhard Weiss, Springer Verlog - LNCS 2585 – 2002.

Applied Physics

Credit Hours:	4 (3,1)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Electric force and its applications and related problems , conservation of charge, charge quantization, Electric fields due to point charge and lines of force. Ring of charge, Disk of charge, A point charge in an electric field, Dipole in a n electric field, The flux of vector field, The flux of electric field, Gauss’ Law, Application of Gauss’ Law, Spherically symmetric charge distribution, A charge isolated conductor, Electric potential energy, Electric potentials, Calculating the potential from the field and related problem Potential due to point and continuous charge distribution, Potential due to dipole, equipotential surfaces, Calculating the field from the potential , Electric current, Current density, Resistance, Resistivity and conductivity, Ohm’s law and its applications, The Hall effect, The magnetic force on a current, The Biot- Savart law, Line of B, Two parallel conductors, Amperes’ s Law, Solenoid, Toroids, Faraday’s experiments, Faraday’s Law of Induction, Lenz’s law, Motional emf, Induced electric field, Induced electric fields, The basic equation of electromagnetism, Induced Magnetic field, The displacement current, Reflection and Refraction of light waves, Total internal reflection, Two source interference, Double Slit interference, related problems, Interference from thin films, Diffraction and the wave theory, related problems, Single-Slit Diffraction, related problems, Polarization of electromagnetic waves, Polarizing sheets, related problems.

Teaching Methodology:

Lecturing, Written Assignments, Project, Experiments, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Experiments, Final Exam

Reference Materials:

1. Fundamentals of Physics (Extended), 10th edition, Resnick and Walker
2. Narciso Garcia, Arthur Damask, Steven Schwarz., “Physics for Computer Science Students”, Springer Verlag, 1998

Artificial Intelligence

Credit Hours:	3+1	**Prerequisites:**	Data Structures and Algorithms

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand key components in the field of artificial intelligence	C	2
2. Implement classical artificial intelligence techniques	C	3
3. Analyze artificial intelligence techniques for practical problem solving	C	4

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction (Introduction, basic component of AI, Identifying AI systems, branches of AI, etc.); Reasoning and Knowledge Representation (Introduction to Reasoning and Knowledge Representation, Propositional Logic, First order Logic); Problem Solving by Searching (Informed searching, Uninformed searching, Local searching.); Constraint Satisfaction Problems; Adversarial Search (Min-max algorithm, Alpha beta pruning, Game-playing); Learning (Unsupervised learning, Supervised learning, Reinforcement learning) ;Uncertainty handling (Uncertainty in AI, Fuzzy logic); Recent trends in AI and applications of AI algorithms (trends, Case study of AI systems, Analysis of AI systems)

Teaching Methodology:

Lectures, Assignments, labs, Projects, Presentations, etc. Major component of the course should be covered using conventional lectures. Practical contact hours are compulsory (~45 hours in a semester).

Course Assessment:

Exams, Assignments, Quizzes, Project, Presentations. Course will be assessed using a combination of written examinations and project(s). Practical evaluation, using rubrics, is encouraged and suggested to make up around 20% of the course.

Reference Materials:

1. Stuart Russell and Peter Norvig, Artificial Intelligence. A Modern Approach, 3rd edition, Prentice Hall, Inc., 2010.
2. Hart, P.E., Stork, D.G. and Duda, R.O., 2001. Pattern classification. John Willey & Sons.
3. Luger, G.F. and Stubblefield, W.A., 2009. AI algorithms, data structures, and idioms in Prolog, Lisp, and Java. Pearson Addison-Wesley.

Big Data Analytics

Credit Hours:	3(2,1)	**Prerequisites:**	Probability and Statistics, Programming Fundamentals

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Provide fundamental information to get insight into the challenges with big data.	C	1
2. Understand techniques for storing and processing large amounts of structured and unstructured data	C	2
3. Application of big data concepts to get valuable information on market trends	C	3
4. Implement and deploy a sample project for extracting useful information from a mid sized dataset.	C	4

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction to Big Data Analytics, Big Data Platforms, Data Store & Processing using Hadoop, Big Data Storage and Analytics, Big Data Analytics ML Algorithms, Recommendation, Clustering, and Classification, Linked Big Data: Graph Computing and Graph Analytics, Graphical Models and Bayesian Networks, Big Data Visualization, Cognitive Mobile Analytics.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman, Jeff Ullman, 2nd edition, 2011
2. Hadoop: The Definitive Guide, Tom White, 4th edition. 2009.
3. Data-Intensive Text Processing with Map Reduce, Jimmy Lin and Chris, 2010.

Business Process Engineering

Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Business process management, Manufacturing and services processes, Modelling and charting tools, Lean processes Improvement workshop techniques, Business process outsourcing, Re-engineering and improvement cases

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Final Exam

Reference Materials:

1. Business Process Improvement; The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness, H. J. Harrington
2. Business Intelligence: A Managerial Approach by Turban, Sharda, Delen, King, 2nd Edition, Prentice Hall (2011). ISBN: 13-978-0-136-10066-9

Business Process Management

Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Discover the various technologies that support Business Process Management	C	2
2. Analyze the performance of existing processes and identify process improvement.	C	3
3. Propose business solutions in written and verbal forms for process innovation and redesign projects.	C	3
4. Create a BPM implementation strategy and implementation plan for an organization.	C	5

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction to Business Process Management, Motivation and Definitions, Business Process Lifecycle, Classification of Business Processes, Goals, Structure, and Organization. Evolution of Enterprise Systems Architectures. Business Process Modeling. Process Orchestrations. Process Choreographies. Modeling in BPMN. Properties of Business Processes. Workflow Management Architectures, Flexible Workflow Management, Web Services and their Composition, Advanced Service Composition, Data-Driven Processes. Business Process Management Methodology.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Business Process Management: Concepts, Languages, Architectures by Mathias Weske, Springer; 2nd Ed. 2012
2. Business Process Management Common Body of Knowledge by Yvonne LedererAntonucci, et. al., Create Space Independent Publishing Platform, 2009
3. Process Management: A Guide for the Design of Business Processes by Jörg Becker, Martin Kugeler and Michael Rosemann, Springer; 2nd Ed. 2011
4. BPMN Method and Style with BPMN Implementer's Guide: A structured approach for business process modeling and implementation using BPMN 2.0 by Bruce Silver, Cody Cassidy Press, 2011.

Calculus & Analytical Geometry

Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Limits and Continuity; Introduction to functions, Introduction to limits, Techniques of funding limits, Indeterminate forms of limits, Continuous and discontinuous functions and their applications, Differential calculus; Concept and idea of differentiation, Geometrical and Physical meaning of derivatives, Rules of differentiation, Techniques of differentiation, Rates of change, Tangents and Normals lines, Chain rule, implicit differentiation, linear approximation, Applications of differentiation; Extreme value functions, Mean value theorems, Maxima and Minima of a function for single-variable, Concavity, Integral calculus; Concept and idea of Integration, Indefinite Integrals, Techniques of integration, Riemann sums and Definite Integrals, Applications of definite integrals, Improper integral, Applications of Integration; Area under the curve, Analytical Geometry; Straight lines in R3, Equations for planes.

Teaching Methodology:

Lecturing, Written Assignments

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Final Exam

Reference Materials:

1. Calculus and Analytic Geometry by Kenneth W. Thomas.
2. Calculus by Stewart, James.
3. Calculus by Earl William Swokowski; Michael Olinick; Dennis Pence; Jeffery A. Cole.

Communication & Presentation Skills

Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Principles of writing good English, understanding the composition process: writing clearly; words, sentence and paragraphs; Comprehension and expression; Use of grammar and punctuation. Process of writing, observing, audience collecting, composing, drafting and revising, persuasive writing, reading skills, listening skills and comprehension, skills for taking notes in class, skills for exams; Business communications; planning messages, writing concise but with impact. Letter formats, mechanics of business, letter writing, letters, memo and applications, summaries, proposals, writing resumes, styles and formats, oral communications, verbal and non-verbal communication, conducting meetings, small group communication, taking minutes. Presentation skills; presentation strategies, defining the objective, scope and audience of the presentation, material gathering material organization strategies, time management, opening and concluding, use of audio-visual aids, delivery and presentation.

Teaching Methodology:

Lecturing, Written Assignments, Project, Presentation, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Practical Business English, Collen Vawdrey, 1993, ISBN = 0256192740
2. Effective Communication Skills: The Foundations for Change, John Nielsen, 2008, ISBN = 1453506748

Compiler Construction			
Credit Hours:	3	**Prerequisites:**	Theory of Automata

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code generation		
2. Understand the basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines		
3. Design and implement a compiler using a software engineering approach		
4. Use generators (e.g. Lex and Yacc)		
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction to interpreter and compiler. Compiler techniques and methodology; Organization of compilers; Lexical and syntax analysis; Parsing techniques. Types of parsers, top-down parsing, bottom-up parsing, Type checking, Semantic analyser, Object code generation and optimization, detection and recovery from errors.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Compilers: Principles, Techniques, and Tools, A. V. Aho, R. Sethi and J. D. Ullman, Addison-Wesley, 2nd ed., 2006
2. Modern Compiler Design, D. Grune, H. E. Bal, C. J. H. Jacobs, K. G. Langendoen, John Wiley, 2003.
3. Modern Compiler Implementation in C, A. W. Appel, M. Ginsburg, Cambridge University Press, 2004.

Computer Graphics

Credit Hours:	3	Prerequisites:	None

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Comprehend the structure of modern computer graphics systems		
2. Explain the basic principles of implementing computer graphics fundamentals		
3. Compare key algorithms for modelling and rendering graphical data		
4. Develop design and problem solving skills with applications to computer graphics		
5. Construct interactive computer graphics programs using OpenGL		

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Fundamental Concepts: forward and backward rendering (i.e., ray-casting and rasterization), applications of computer graphics: including game engines, cad, visualization, virtual reality, polygonal representation, basic radiometry, similar triangles, and projection model, use of standard graphics APIs (see HCI GUI construction); basic rendering: rendering in nature, i.e., the emission and scattering of light and its relation to numerical integration, affine and coordinate system transformations, ray tracing, visibility and occlusion, including solutions to this problem such as depth buffering, painter's algorithm, and ray tracing, the forward and backward rendering equation, simple triangle rasterization, rendering with a shader-based API, texture mapping, including minification and magnification (e.g., trilinear MIP-mapping), application of spatial data structures to rendering, sampling and anti-aliasing, scene graphs and the graphics pipeline; geometric modeling: basic geometric operations such as intersection calculation, proximity tests, polynomial curves and surfaces, approximation techniques such as polynomial curves, bezier curves, spline curves and surfaces, animation as a sequence of still images.

Teaching Methodology:

Lectures, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Computer Graphics with Open GL (4th Edition) by Donald D. Hearn, Prentice Hall, 2010, ISBN-10: 0136053580.
2. Foundations of 3D Computer Graphics by S. J. Gortler, The MIT press, 2012.
3. Fundamentals of Computer Graphics, 3rd Edition, A K Peters, 2009.
4. Computer Graphics: Principles and Practice, 3rd Edition, Addison Wesley, 2013.
5. Real-Time Rendering, 3rd Edition, A K Peters, 2008.

Computer Networks

Credit Hours:	3+1	**Prerequisites:**	None

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. **Describe** the key terminologies and technologies of computer networks	C	2
2. **Explain** the services and functions provided by each layer in the Internet protocol stack.	C	2
3. **Identify** various internetworking devices and protocols, and their functions in a network.	C	4
4. **Analyze** working and performance of key technologies, algorithms and protocols.	C	4
5. **Build** Computer Network on various Topologies	P	3
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction and protocols architecture, basic concepts of networking, network topologies, layered architecture, physical layer functionality, data link layer functionality, multiple access techniques, circuit switching and packet switching, LAN technologies, wireless networks, MAC addressing, networking devices, network layer protocols, IPv4 and IPv6, IP addressing, sub netting, CIDR, routing protocols, transport layer protocols, ports and sockets, connection establishment, flow and congestion control, application layer protocols, latest trends in computer networks.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Computer Networking: A Top-Down Approach Featuring the Internet, 6th edition by James F. Kurose and Keith W. Ross
2. Computer Networks, 5th Edition by Andrew S. Tanenbaum
3. Data and Computer Communications, 10th Edition by William Stallings
4. Data Communication and Computer Networks, 5th Edition by Behrouz A. Forouzan

Computer Organization and Assembly Language

Credit Hours:	3+1	**Prerequisites:**	Programming Fundamentals

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Acquire the basic knowledge of computer organization, computer architecture and assembly language		
2. Understand the concepts of basic computer organization, architecture, and assembly language techniques		
3. Solve the problems related to computer organization and assembly language		
BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction to computer systems: Information is bits + context, programs are translated by other programs into different forms, it pays to understand how compilation systems work, processors read and interpret instructions stored in memory, caches matter, storage devices form a hierarchy, the operating system manages the hardware, systems communicate with other systems using networks; Representing and manipulating information: information storage, integer representations, integer arithmetic, floating point; Machine-level representation of programs: a historical perspective, program encodings, data formats, accessing information, arithmetic and logical operations, control, procedures, array allocation and access, heterogeneous data structures, putting it together: understanding pointers, life in the real world: using the gdb debugger, out-of-bounds memory references and buffer overflow, x86-64: extending ia32 to 64 bits, machine-level representations of floating-point programs; Processor architecture: the Y86 instruction set architecture, logic design and the Hardware Control Language (HCL), sequential Y86 implementations, general principles of pipelining, pipelined Y86 implementations

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Computer Systems: A Programmer's Perspective, 3/E (CS:APP3e), Randal E. Bryant and David R.O' Hallaron, Carnegie Mellon University
2. Robert Britton, MIPS Assembly Language Programming, Latest Edition,
3. Computer System Architecture, M. Morris Mano, Latest Edition,
4. Assembly Language Programming for Intel- Computer, Latest Edition

Cyber Security

Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Basic security concepts, Information security terminology, Malware classifications, Types of malware. Server side web applications attacks. Cross-site scripting, SQL Injection, Cross-site request forgery, Planning and policy, Network protocols and service models. Transport layer security, Network layer security, Wireless security, Cloud & IoT security.

Teaching Methodology:

Lecturing, Written Assignments, Project

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Lab, Presentation, Final Exam

Reference Materials:

3. Security+ Guide to Network Security Fundamentals by Mark Ciampa, th Edition
4. Corporate Computer Society by Randall J.Boyle, 3rd Edition

Computer Vision

Credit Hours:	3	**Prerequisites:**	None

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level[*]**
1. Understand and explain the field of computer vision in general for different applications, etc.	C	1,2
2. Understand and implement camera calibration	C	1,2,3
3. Work under OpenCV or Matlab computer vision toolbox, etc.	C	1,2,3
4. Implement an algorithm to assemble the extracted features to develop a higher-level perception	C	3,6
5. Implement different algorithms for spatial and frequency domain filtering, feature detection, structure from motion, motion estimation, etc.	C	3
6. To detect, recognize and track different types of the objects in the scene	C	3,6
7. Develop an algorithm for context awareness or scene understanding	C	3,6

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction, Image formation, Spatial and frequency domain processing, Feature detection and extraction, Image registration, Segmentation, Camera calibration, Structure from motion, Motion estimation, Stereo vision, Object detection and recognition, Object tracking, 3D scene reconstruction, Context and scene understanding, Image stitching, Image-based and video-based rendering, High-performance computing paradigms for vision and image processing.,

Teaching Methodology:

Lectures, Written Assignments, Semester Project.

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Final Exam

Reference Materials:

1. Computer Vision - A Modern Approach, by D. Forsyth and J. Ponce, Prentice Hall, 2003.
2. Szeliski R., Computer Vision - Algorithms and Applications, Springer, 2011.
3. J. R. Parker, Algorithms for Image Processing and Computer Vision, Willey Publishing Inc. 2011.
4. Gonzalez R. C., Woods R. E., Digital Image Processing, Pearson Education, 3rd edition, 2008.

Data Encryption and Security

Credit Hours:	3	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
CLO-1:. CLO-2: CLO-3:. CLO-4:.	C	

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Principle of number theory and probability theory, Primes, random numbers, modular arithmetic and discrete logarithms. Cryptographic algorithms and design principles, including conventional and symmetric encryption (DES, IDEA, Blowfish, Rijndael, RC-4, RC-5), public key or asymmetric encryption (RSA, Diffie-Hellman), key management, hash functions (MD5, SHA-1, RIPEMD-160, HMAC), digital signatures and certificates. Authentication protocols (X.509, Kerberos), electronic mail security (S/MIME, PGP), web security and protocols for secure electronic commerce (IPSec, SSL, TLS, SET).

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Cryptography and Network Security: Principles and Practice, William Stallings, 6th edition.

Data Structures and Algorithms

Credit Hours:	3+1	**Prerequisites:**	Programming Fundamentals

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Implement various data structures and their algorithms, and **apply** them in implementing simple applications.	C	2,3
2. **Analyze** simple algorithms and determine their complexities.	C	4,5
3. **Apply** the knowledge of data structures to other application domains.	C	3
4. **Design** new data structures and algorithms to solve problems.	C	6
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Abstract data types, complexity analysis, Big Oh notation, Stacks (linked lists and array implementations), Recursion and analyzing recursive algorithms, divide and conquer algorithms, Sorting algorithms (selection, insertion, merge, quick, bubble, heap, shell, radix, bucket), queue, dequeuer, priority queues (linked and array implementations of queues), linked list & its various types, sorted linked list, searching an unsorted array, binary search for sorted arrays, hashing and indexing, open addressing and chaining, trees and tree traversals, binary search trees, heaps, M-way tress, balanced trees, graphs, breadth-first and depth-first traversal, topological order, shortest path, adjacency matrix and adjacency list implementations, memory management and garbage collection.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Data Structures and Algorithms in C++ by Adam Drozdek
2. Data Structures and Algorithm Analysis in Java by Mark A. Weiss
3. Data Structures and Abstractions with Java by Frank M. Carrano & Timothy M. Henry
4. Data Structures and Algorithm Analysis in C++ by Mark Allen Weiss
5. Java Software Structures: Designing and Using Data Structures by John Lewis and Joseph Chase

Database Administration & Management

Credit Hours:	3 (3,0)	**Prerequisites:**	Database System

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level** *
* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction to advance data models such as object relational, object oriented. File organizations concepts, Transactional processing and Concurrency control techniques, Recovery techniques, Query processing and optimization, Database Programming, Integrity and security, Database Administration, Physical database design and tuning, Distributed database systems, Emerging research trends in database systems.

Teaching Methodology:

Lecturing, Written Assignments, Project & Research

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Fundamentals of Database Systems, by Ramez Elmasri and Shamkant Navathe, Addison Wesley, 5th Edition.
2. Database System Concepts by Henry F. Korth and Abraham Silberschatz, 4th edition, McGraw Hill, 2002, ISBN: 0-07-12268-0

Database Systems			
Credit Hours:	3+1	**Prerequisites:**	None

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. **Explain** fundamental database concepts.	C	2
2. **Design** conceptual, logical and physical database schemas using different data models.	C	5
3. **Identify** functional dependencies and resolve database anomalies by normalizing database tables.	C	2
4. **Use** Structured Query Language (SQL) for database definition and manipulation in any DBMS	C	4
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Basic database concepts, Database approach vs file based system, database architecture, three level schema architecture, data independence, relational data model, attributes, schemas, tuples, domains, relation instances, keys of relations, integrity constraints, relational algebra, selection, projection, Cartesian product, types of joins, normalization, functional dependencies, normal forms, entity relationship model, entity sets, attributes, relationship, entity-relationship diagrams, Structured Query Language (SQL), Joins and sub-queries in SQL, Grouping and aggregation in SQL, concurrency control, database backup and recovery, indexes, NoSQL systems.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition by Thomas Connolly and Carolyn Begg
2. Database Systems: The Complete Book, 2nd Edition by Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom
3. Database System Concepts, 6th Edition by Avi Silberschatz, Henry F. Korth and S. Sudarshan.
4. Database Management Systems, 3rd Edition by Raghu Ramakrishnan, Johannes Gehrke

Design and Analysis of Algorithms

Credit Hours:	**3**	**Prerequisites:**	**Data Structures and Algorithms**

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm		
2. Identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors.		
3. Determine informally the time and space complexity of simple algorithms		
4. List and contrast standard complexity classes		
5. Use big O, Omega, Theta notation formally to give asymptotic upper bounds on time and space complexity of algorithms		
6. Use of the strategies(brute-force, greedy, divide-and-conquer, and dynamic programming) to solve an appropriate problem		
7. Solve problems using graph algorithms, including single-source and all-pairs shortest paths, and at least one minimum spanning tree algorithm		
8. Trace and/or implement a string-matching algorithm		

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction; role of algorithms in computing, Analysis on nature of input and size of input Asymptotic notations; Big-O, Big Ω, Big Θ, little-o, little-ω, Sorting Algorithm analysis, loop invariants, Recursion and recurrence relations; Algorithm Design Techniques, Brute Force Approach, Divide-and-conquer approach; Merge, Quick Sort, Greedy approach; Dynamic programming; Elements of Dynamic Programming, Search trees; Heaps; Hashing; Graph algorithms, shortest paths, sparse graphs, String matching; Introduction to complexity classes;

Teaching Methodology:

Lectures, Written Assignments, Semester Project.

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Final Exam

Reference Materials:

1. Introduction to Algorithms (3rd edition) by Thomas H. Corman, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein
2. Algorithm Design, (1st edition, 2013/2014), Jon Kleinberg, Eva Tardos,
3. Algorithms, (4th edition, 2011), Robert Sedgewick, Kevin Wayne

Differential Equations

Credit Hours:	**3-0**	**Prerequisites:**	**Calculus and Analytical Geometry**

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Identify, analyze and subsequently solve physical situations whose behavior can be described by ordinary differential equations.		
2. Determine solutions to first order separable differential equations.		
3. Determine solutions to first order linear differential equations.		
4. Determine solutions to first order exact differential equations.		
5. Determine solutions to second order linear homogeneous and non-homogeneous differential equations with constant coefficients.		
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Ordinary Differential Equations of the First Order: Geometrical Considerations, Isoclines, Separable Equations, Equations Reducible to Separable Form, Exact Differential Equations, Integrating Factors, Linear First-Order Differential Equations, variation of Parameters. Ordinary Linear Differential Equations; Homogeneous Linear Equations of the Second Order, Homogeneous Second-Order Equations with Constant Coefficients, General Solution, Real Roots, Complex Roots, Double Root of the Characteristic Equation, Differential Operators, Cauchy Equation, Homogeneous Linear Equations of Arbitrary Order, Homogeneous Linear Equations of Arbitrary Order with Constant Coefficients, Non- homogeneous Linear Equations. Modelling of Electrical Circuits. Systems of Differential Equations. Series Solutions of Differential Equations. Partial Differential Equations: Method of Separation of variables, wave, Heat & Laplace equations and their solutions by Fourier series method.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. *Advanced Engineering Mathematics* Michael, G.*1996*, Prentice Hall Publishers.
2. *Advanced Engineering Mathematics*, 7th edition, Erwin, K. 1993, John Wiley & Sons Inc.
3. *A First Course in Differential Equation* Zill. Prindle. Weber. Schmidt.1996. Brooks/Cole Publishing.
4. *Differential Equations with Boundary-Value Problems*, Dennis. G. Zill, Michael, R. Cullen. 1996, Brooks/Cole Publishing,
5. *Elementary Differential Equations with Applications* C. H. Edwards. David, E. 1993. Penney, Prentice Hall.

DIGITAL IMAGE PROCESSING			
Credit Hours:	3+1	**Prerequisites:**	None

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the basics, applications in general, working inside the digital camera, sampling and quantization, image representation, etc.	C	1,2
2. Implement image enhancement, image segmentation, image transformations, spatial and frequency domain processing, filtering, convolution, image registration, feature detection, pattern recognition, etc.	C	3
3. Evaluate the performance of different image processing algorithms.	C	4,5
* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

The human visual system, electromagnetic system, working and components inside digital camera, pixels, image representation, sampling, quantization, mathematics of image formation, convolution, camera projection, point-based image processing, Fourier theory, image filtering in spatial and frequency domain, wavelets, image registration, morphological operations, color models, multispectral images, feature detection, image segmentation, Pattern recognition, etc.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Midterm Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Gonzalez R. C., Woods R. E., Eddins S. L., Digital Image Processing Using Matlab, Pearson Education, 2nd edition, 2009.
2. Gonzalez R. C., Woods R. E., Digital Image Processing, Pearson Education, 3rd edition, 2008.
3. Understanding Digital Signal Processing by Richard G. Lyons, Prentice Hall; 3rd edition, 2010.

Digital Logic Design

Credit Hours:	3	**Prerequisites:**	Applied Physics

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***[*]**
1. Acquire knowledge related to the concepts, tools and techniques for the design of digital electronic circuits		
2. Demonstrate the skills to design and analyze both combinational and sequential circuits using a variety of techniques		
3. Apply the acquired knowledge to simulate and implement small-scale digital circuits		
4. Understand the relationship between abstract logic characterizations and practical electrical implementations.		

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Number Systems, Logic Gates, Boolean Algebra, Combination logic circuits and designs, Simplification Methods (K-Map, Quinn Mc-Cluskey method), Flip Flops and Latches, Asynchronous and Synchronous circuits, Counters, Shift Registers, Counters, Triggered devices & its types. Binary Arithmetic and Arithmetic Circuits, Memory Elements, State Machines. Introduction Programmable Logic Devices (CPLD, FPGA); Lab Assignments using tools such as Verilog HDL/VHDL, MultiSim

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Digital Fundamentals by Floyd, 11/e.
2. Fundamental of Digital Logic with Verilog Design, Stephen Brown, 2/e.

Discrete Structures				
Credit Hours:	3+0	**Prerequisites:**	None	
Course Learning Outcomes (CLOs):				
At the end of the course the students will be able to:			**Domain**	**BT Level***
1. **Understand** the key concepts of Discrete Structures such as Sets, Permutations, Relations, Graphs, and Trees etc.			C	2
2. **Apply** formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles.			C	3
3. **Apply** discrete structures into other computing problems such as formal specification, verification, databases, artificial intelligence, and cryptography.			C	3
4. **Differentiate** various discrete structures and their relevance within the context of computer science, in the areas of data structures and algorithms, in particular.			C	4
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain				

Course Content:

Mathematical reasoning, propositional and predicate logic, rules of inference, proof by induction, proof by contraposition, proof by contradiction, proof by implication, set theory, relations, equivalence relations and partitions, partial orderings, recurrence relations, functions, mappings, function composition, inverse functions, recursive functions, Number Theory, sequences, series, counting, inclusion and exclusion principle, pigeonhole principle, permutations and combinations, elements of graph theory, planar graphs, graph coloring, euler graph, Hamiltonian path, rooted trees, traversals.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Discrete Mathematics and Its Applications, 7th edition by Kenneth H. Rosen
2. Discrete Mathematics with Applications, 4th Edition by Susanna S. Epp
3. Discrete Mathematics, 7th edition by Richard Johnson Baugh
4. Discrete Mathematical Structures, 4th edition by Kolman, Busby & Ross
5. Discrete and Combinatorial Mathematics: An Applied Introduction by Ralph P. Grimaldi
6. Logic and Discrete Mathematics: A Computer Science Perspective by Winifred Grassman

E-Commerce			
Credit Hours:	3(3,0)	**Prerequisites:**	Web Engineering

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the concepts and standards related to the discipline of E-Commerce.	C	
2. Analyze complex real world problems found in E-Commerce	C	
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

An overview of E-Commerce & its business models and concepts, Planning an E-Commerce Framework, Managing Products and Categories, Product Variations and User Uploads, Enhancing the User Experience, The Shopping Basket, The Checkout and Order Process, Shipping and Tax, Discounts, Vouchers, and Referrals, Checkout, Taking Payment for Orders, User Account Management, Administration: Dashboard, Managing Products and Categories, Managing Orders, Customers, Refunds, Voucher Codes, Shipping, Deploying, Security, and Maintenance, Web Payment Systems, Social, Legal, and Ethical Issues of E-Commerce, Auctions, Portals, and Communities, SEO.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. E-Commerce, Kenneth Laudon and Carol Guercio Traver, 13th Edition, Pearson, 2017.
2. PHP 5 E-commerce Development, Michael Peacock, Packt Publishing, 2010.
3. Introduction to E-Commerce, Jeffrey F. Rayport, McGraw-Hill, 2nd Edition, 2007.
4. Electronic Commerce, Gary Schneider, Course Technology; 12th Edition 2016

English Composition & Comprehension			
Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Paragraph and Essay Writing, Descriptive Essays; Sentence Errors, Persuasive Writing; How to give presentations, Sentence Errors; Oral Presentations, Comparison and Contrast Essays, Dialogue Writing, Short Story Writing, Review Writing, Narrative Essays, Letter Writing

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. College Writing Skills with Readings, by John Langan, McGraw-Hill, 5th Edition.
2. A Textbook of English Prose and Structure by Arif Khattak, et al, GIKI Institute, 2000

Enterprise Systems			
Credit Hours:	3 (3,0)	**Prerequisites:**	Database System

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level**[*]
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Fundamentals of an Enterprise and Industries artifacts. Introduction to Enterprise Resource Planning (ERP). ERP Implementation life cycle methodologies and strategy. Business processes, architecture, User Interface Designs and their modeling. ERP Security, workflows, data integration, applications migration and data migration. Study of business modules Human Resource, Procurement, Sales and Distribution, Material Management, and Manufacturing. Concepts and tools of designing and implementing an ERP system. Emerging trends in ERP and special topics such as Supply Chain Management (SCM), Customer Relationship Management (CRM), Business Intelligence (BI).

Teaching Methodology:

Lecturing, Written Assignments, Project & Lab Work

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentation, Final Exam

Reference Materials:

1. Enterprise Resource Planning by Rajesh Ray, Tata McGraw Hill Education Private Limited, New Delhi, 2011
2. Design of Industrial Information Systems by Thomas O. Boucher, Ali Yalcin, Elsevier AP Printer, 2006
3. Enterprise Application Integration by David S. Linthicum, Addison Wesley Information Technology Series, 2000

Formal Methods in Software Engineering

Credit Hours:	3 (3,0)	Prerequisites:	Discrete Structures

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level**[*]
1. Describe the costs and benefits of formal methods'	C	1
2. Construct formal models of sequential software systems	C	2
3. Implement sequential software systems based on formal models	C	3
4. Verify attributes of formal models	C	3
5. Demonstrate formal correctness of simple procedure	C	4

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction to the use of mathematical models for specification and validation, Finite state machine models, models of concurrent systems, verification of models, and limitations. Analyzing well-formedness (e.g. completeness, consistency, robustness, etc.), Analyzing correctness (e.g. static analysis, simulation, model checking, etc.), Formal analysis, An introduction to VDM-SL, Sets, Sequences, Composite objects, Maps, VDM-SL, Comparative Formal Methods, Proofs, Introduction to Z

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Modern Formal Methods and Applications, Hossam A. Gabbar, Springer-Verlag 2006.
2. Formal Software Development: From VDM to Java, Charatan, Quentin, and Aaron Kans. Palgrave Macmillan, 2003.
3. Understanding Z: a Specification Language and its Formal Semantics. J. M. Spivey. 1988. Cambridge University Press, New York, NY, USA.

Global Software Development

Credit Hours:	3(3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the principles of the software engineering in context of global software development.	C	2
2. Evaluate and discuss the issues around global software development and techniques for managing distributed projects.	C	4
3. Understand Configuration management systems, release management and task assignments in context of distributed projects.	C	2
4. Acquire strategies for effectively dividing tasks among teams, controlling the communication among teams, planning tasks and collaborating on modular project with the help of realistic examples.	C	3

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction to Global Software Development. Global Teams and Organization. Guideline for making the virtual team. The Geography of Coordination. Dealing with Distance. Architectures and Coordination: Reconfiguration of Existing Product Technologies, Identification of Coordination Requirements. Distributed Development Environments: Software configuration management, Awareness among Configuration Management. Challenges of Culture: Managing distances and differences in geographically distributed work groups. The Outsourcing Relationship. Facilitating Cross-site Trust, Cooperation, and Social Capital: Communication and Trust in Global Virtual Teams. Social Networks and Knowledge Networks. Communication and Awareness: dealing with distance. Assessing Coordination Risk.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Global Software and IT: A Guide to Distributed Development, Projects”, Christof Ebert, Wiley 2011.
2. Global Software Teams: Collaborating Across Borders and Time Zones”, Erran Carmel. Prentice Hall, 1999.

Graph Theory

Credit Hours:	3	**Prerequisites:**	Nil

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	Domain	**BT Level***
1. To introduce the fundamental concepts of Graph Theory.	C	**1**
2. To provide knowledge for application of Graph Theory in subsequent courses in the design and analysis of algorithms, computability theory, software engineering, and computer systems.	C	2

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction to Graph Theory, Basic definitions, computer representations and properties of Graph, Data structure for representing Graphs, Fundamental theorem of Graph Theory, Isomorphic and Special Graphs, Properties of Trees and Forests, Binary tree, Balanced binary tree, Directed and Undirected rooted tree, Minimum Spanning Tree algorithms and implementation, Path and Distance in graphs, Shortest path algorithms and implementation, Cycle and distance in weighted graph and digraphs, Distance algorithms and implementation, Eulerian graphs and Hamiltonians graphs with applications, Flow networks, Max-flow Min-cut Theorem, Graph coloring, Edge coloring, Planar graphs, Four color theorem, Deadlock of computer system, Matching Algorithms, Dominance & Ramsey theory.

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. *Graph Theory & Applications* (1st Edition) by Fournier. Published by Wiley-ISTE, 2011.
2. *Applied Algorithmic Graph Theory* (1st Edition) by Chartrand. Published by McGraw-Hill College, 1995.
3. *Handbook of Graph Theory* (Series Edition) by Jonathan Published by CRC Press, 2004.
4. *Graph Theory with Applications* (8th Edition) by J. A. Bondy, Published Elsevier USA, 1982.

Human Computer Interaction

Credit Hours:	3 (3,0)	**Prerequisites:**	Software Engineering

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain context of HCI and different measures for evaluation.	C	2
2. Apply the principles of good design for people from the perspective of age and disabilities.	C	3
3. Analyze techniques for user centered design for a medium sized software.	C	4
4. Evaluate the usability of a medium size software user interface.	C	5
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Contexts for HCI, Psychology of usable things, Processes for User-Centered Design, Metrics and Measures for Evaluation, Usability heuristics and principles of Usability testing, Physical capabilities, Cognitive and social models for interaction design, Principles of good interaction design, Accessibility, Principles of GUI, Visual design elements, Data gathering, Task analysis, Prototyping, Help and user documentation, Internationalization, Usability inspection methods, Usability testing methods, New Interaction Technologies, Usability in practice, Visual Design and Typography, Icon Design, Ubiquitous, Augmented and Virtual Reality.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Designing the User Interface: Strategies for Effective Human-Computer Interaction, Ben Shneiderman and Catherine Plaisant, 6th Ed, Pearson Inc, 2016.
2. Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction Design, Benyon, D. 3rd Ed., Pearson. 2013
3. About Face: The Essentials of Interaction Design, Alan Cooper, Robert Reimann, David Cronin, Christopher Noessel, 4th Ed, Wiley, 2014

Information Security

Credit Hours:	3+0	**Prerequisites:**	None

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. **Explain** key concepts of information security such as design principles, cryptography, risk management, and ethics	C	2
2. **Discuss** legal, ethical, and professional issues in information security.	A	2
3. **Apply** various security and risk management tools for achieving information security and privacy.	C	3
4. **Identify** appropriate techniques to tackle and solve problems in the discipline of information security.	C	4

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Information security foundations, security design principles; security mechanisms, symmetric and asymmetric cryptography, encryption, hash functions, digital signatures, key management, authentication and access control; software security, vulnerabilities and protections, malware, database security; network security, firewalls, intrusion detection; security policies, policy formation and enforcement, risk assessment, cybercrime, law and ethics in information security, privacy and anonymity of data.

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Computer Security: Principles and Practice, 3rd edition by William Stallings
2. Principles of Information Security, 6th edition by M. Whitman and H. Mattord
3. Computer Security, 3rd edition by Dieter Gollmann
4. Computer Security Fundamentals, 3rd edition by William Easttom
5. Official (ISC)2 Guide to the CISSP CBK, 3rd edition

Information Systems Audit

Credit Hours:	3	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the concepts and standards related to the discipline of Information System Audit.	C	1
2. Analyze and Audit Information Systems	C	4
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction to Auditing, IS Audit charter, Polices, Procedures, The Audit Process, Audit computer networks and communication, Auditing software development, Acquisition, Maintenance, Auditing IT infrastructure, Auditing Management and Organization, Business process re-engineering: IS audit proposal, report, evidence and follow-up, complaint to standard, Enterprise service agreement, IP pro count policies and process, Backup and procedures, Overview of Computer-Assisted Audit Tools and Techniques.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Auditing Information Systems: Enhancing Performance of the Enterprise, Abraham Nyirongo, Trafford, 2015.
2. Information Systems Control and Audit, Ron Weber, Dorling Kindesley Pearson Education, 2014
3. CISA® Certified Information Systems Auditor All-in-One Exam Guide, Peter Gregory, 3rd Edition, McGraw-Hill Education, 2016

Information Technology Project Management			
Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level[*]**
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction to Project Management. The Project Management and Information Technology Context. The Project Management Process Groups. Project Integration Management. Project Scope Management. Project Time Management. Project Cost Management. Project Quality Management. Project Human Resource Management. Project Communications Management. Project Risk Management. Project Procurement Management. Project Management Tools.

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

1. Information Technology Project Management by Kathy Schwalbe, Course Technology; 6th Edition (July 22, 2010). ISBN-10: 1111221758
2. A Guide to the Project Management Body of Knowledge, 3rd Edition (PMBOK Guides), ISBN-13: 978-1930699458
3. IT Project Management: On Track from Start to Finish by Joseph Phillips, McGraw-Hill Osborne Media; 3rd Edition (February 25, 2010). ISBN-10: 0071700439
4. Information Technology Project Management by Jack T. Marche, Wiley; 3rd Edition (January 6, 2009). ISBN-10: 0470371935

Introduction to Information and Communication Technologies

Credit Hours:	3 (2,1)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Principles of writing good English, understanding the composition process: writing clearly; words, sentence and paragraphs; Comprehension and expression; Use of grammar and punctuation. Process of writing, observing, audience collecting, composing, drafting and revising, persuasive writing, reading skills, listening skills and comprehension, skills for taking notes in class, skills for exams; Business communications; planning messages, writing concise but with impact. Letter formats, mechanics of business, letter writing, letters, memo and applications, summaries, proposals, writing resumes, styles and formats, oral communications, verbal and non-verbal communication, conducting meetings, small group communication, taking minutes. Presentation skills; presentation strategies, defining the objective, scope and audience of the presentation, material gathering material organization strategies, time management, opening and concluding, use of audio-visual aids, delivery and presentation.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Practical Business English, Collen Vawdrey, 1993, ISBN = 0256192740
2. Effective Communication Skills: The Foundations for Change, John Nielsen, 2008, ISBN = 1453506748

Introduction to Software Engineering

Credit Hours:	3 (3,0)	Prerequisites:	

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	Domain	BT Level*
1. Describe various software engineering processes and activities	C	1
2. Apply the system modeling techniques to model a medium size software system	C	3
3. Apply software quality assurance and testing principles to medium size software system.	C	4
4. Discuss key principles and common methods for software project management such as scheduling, size estimation, cost estimation and risk analysis	C	2

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Nature of Software, Overview of Software Engineering, Professional software development, Software engineering practice, Software process structure, Software process models, Agile software Development, Agile process models, Agile development techniques, Requirements engineering process, Functional and non-functional requirements, Context models, Interaction models, Structural models, behavioral models, model driven engineering, Architectural design, Design and implementation, UML diagrams, Design patterns, Software testing and quality assurance, Software evolution, Project management and project planning, configuration management, Software Process improvement.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Software Engineering, Sommerville I., 10th Edition, Pearson Inc., 2014
2. Software Engineering, A Practitioner’s Approach, Pressman R. S.& Maxim B. R., 8th Edition, McGraw-Hill, 2015.

Islamic Studies

Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Basic Themes of Quran, Introduction to Sciences of Hadith, Introduction to Islamic Jurisprudence, Primary & Secondary Sources of Islamic Law, Makken & Madnian life of the Prophet, Islamic Economic System, Political theories, Social System of Islam

Teaching Methodology:

Lecturing, Written Assignments, Project

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Introduction to Islam by Dr Hamidullah, Papular Library Publishers Lahore
2. Principles of Islamic Jurisprudence by Ahmad Hassan, Islamic Research Institute, IIUI
3. Muslim Jurisprudence and the Quranic Law of Crimes, By Mir Waliullah, Islamic Books Services

IT Infrastructure

Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Definition of IT Infrastructure, Non-functional Attributes, Availability Concepts, Sources of Unavailability, Availability Patterns. Performance. Security Concepts. Data centres. Servers: Availability, Performance, Security. Networking: Building Blocks, Availability, Performance, Security. Storage: Availability, Performance, Security. Virtualization: Availability, Performance, Security. Operating Systems: Building Blocks, Implementing Various OSs, OS availability, OS Performance, OS Security. End User Devises: Building Blocks, Device Availability, Performance, Security. IT Infrastructure Management. Service Delivery Processes. Service Support Processes. Ethics, Trends, organizational and technical issues related to IT infrastructure.

Teaching Methodology:

Lecturing, Written Assignments, Project, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. IT Infrastructure Architecture: Infrastructure building blocks and concepts by Sjaak Laan, Lulu.com (November 5, 2011). ISBN-10: 1447881281
2. IT Infrastructure and its Management by Prof Phalguni Gupta, Tata McGraw Hill Education Private Limited (October 6, 2009). ISBN-10: 0070699798
3. IT Architecture for Dummies by Kalani Kirk Hausman and Susan Cook, For Dummies; 1st Edition (November 9, 2010). ISBN-10: 0470554231

Linear Algebra

Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level**[*]
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Algebra of linear transformations and matrices. determinants, rank, systems of equations, vector spaces, orthogonal transformations, linear dependence, linear Independence and bases, eigenvalues and eigenvectors ,characteristic equations, Inner product space and quadratic forms

Teaching Methodology:

Lecturing, Written Assignments

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Final Exam

Reference Materials:

1. Elementary Linear Algebra by Howard Anton
2. Linear Algebra and its Applications by Gibert Strang

Logical Paradigm of Computing

Credit Hours:	3	**Prerequisites:**	Discrete Structures

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	Domain	**BT Level***
1. Understand how formal methods (FM) help produce high-quality software	C	1
2. Write and understand formal requirement specifications	C	2
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Propositional logic, Declarative sentences, Natural deduction, Propositional logic as a formal language, Semantics of propositional logic, Normal forms. Predicate logic, The need for a richer language, Predicate logic as a formal language, Proof theory of predicate logic, Semantics of predicate logic, Un-decidability of predicate logic, Expressiveness of predicate logic, Micro models of software, Verification by model checking, Motivation for verification, Linear-time temporal logic LTL Model checking: systems, tools, properties, Branching-time logic CTL, CTL* and the expressive powers of LTL and CTL, Model-checking algorithms. The fixed-point characterization of CTL, Program verification, Why should we specify and verify code? A framework for software verification, Proof calculus for partial correctness, Proof calculus for total correctness, Programming by contract, Modal logics and agents, Modes of truth, Basic modal logic, Logic engineering. Natural deduction, Reasoning about knowledge in a multi-agent system, Binary decision diagrams, Representing Boolean functions, Algorithms for reduced OBDDs, Symbolic model checking. A relational mu-calculus. Introduction to Process Algebra, Modelling Communication, Synchronization, Action and Transition Internal Actions.

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Logic in Computer Science Modelling and Reasoning about Systems 2nd Edition Michael Huth, Mark Ryan, University of Birmingham, 2004
2. Principles Of Model Checking by Christel Baier and Joost-Pieter Katoen MIT Press, 2008
3. Software Reliability Methods Doron Peled, Springer, 2001
4. Communication and Concurrency, R. Milner (1989), Prentice Hall

Management Information System

Credit Hours:	3(3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand and articulate concepts of information technology management.	C	2
2. Assess and apply IT to solve common business problems.	C	2
3. Suggest and defend effective solutions to business problems, and design a database application to solve a business problem.	C	3
4. Explain in details the ethical aspects of information technology use in the organization and its governance issues.	C	2

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction to Information Systems in Organizations; Business Process and Decision Making; Productivity, Innovation and Strategy; Database and Content Management; Decision Making and Business Intelligence; Competitive Advantage and Business Processes; Networks and Collaboration; ERP and E-commerce, Social Networking, and Web 3.0; Acquiring Information Systems Through Projects; Structure, Governance, and Ethics; Managing Information Security and Privacy

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Experiencing MIS, D. M. Kroenke, A. Gemino and P. Tingling. P. 4th Edition. Toronto: Pearson.2016.
2. Business driven information systems, P. Baltzan, B. Detlor, and C. Welsh, 4th Ed., McGraw Hill Ryerson Press, 2015..

Mobile Application Development

Credit Hours:	3(3,0)	**Prerequisites:**	Object Oriented Programming

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Discuss different architectures & framework for Mobile Application development.	C	1
2. Develop mobile applications using current software development environments.	C	3
3. Compare the different performance tradeoffs in mobile application development.	C	3
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Mobiles Application Development Platform; HTML5 for Mobiles; Android OS: Architecture, Framework and Application Development; iOS: Architecture, Framework; Application Development with Windows Mobile; Eclipse; Fragments; Calling Built-in Applications using Intents; Displaying Notifications; Components of a Screen; Adapting to Display Orientation; Managing Changes to Screen Orientation; Utilizing the Action Bar; Creating the User Interface; Listening for UI Notifications; Views; User Preferences; Persisting Data; Sharing Data; Sending SMS Messages; Getting Feedback; Sending E-mail; Displaying Maps; Consuming Web Services Using HTTP; Web Services: Accessing and Creating; Threading; Publishing, Android Applications; Deployment on App Stores; Mobile Programming Languages; Challenges with Mobility and Wireless Communication; Location-aware Applications; Performance/Power Tradeoffs; Mobile Platform Constraints; Emerging Technologies..

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Professional Android application development, Reto Meier, Wrox Programmer to Programmer, 2015.
2. iOS Programming: The Big Nerd Ranch Guide, Conway, J., Hillegass, A., & Keur, C., 5th Edition, 2014.
3. Android Programming: The Big Nerd Ranch Guides, Phillips, B. & Hardy, B., 2nd Edition, 2014.

Simulation and Modeling

Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain the model classification at different levels.	C	1
2. Analyze complex engineering systems and associated issues (using systems thinking and modelling techniques)	C	3
3. Apply advanced theory-based understanding of engineering fundamentals and specialist bodies of knowledge in the selected discipline area to predict the effect of engineering activities.	C	4
4. Analyze the simulation results of a medium sized engineering problem.	C	4
* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction to modelling and simulation, System analysis, Classification of systems, System theory basics, its relation to simulation, Model classification at conceptual, abstract, and simulation models levels, Methodology of model building, Simulation systems and languages, Means for model and experiment description, Principles of simulation system design, Parallel process modeling using Petri nets and finite automata in simulation, Models of queuing systems, Discrete simulation models, Model time, Simulation experiment control, Overview of numerical methods used for continuous simulation. System Dymola/ Modelica, Combined simulation, Special model classes, Models of heterogeneous systems, Cellular automata and simulation, Checking model validity, Verification of models, Analysis of simulation results, simulation results visualization, model optimization, generating, transformation, and testing of pseudorandom numbers with overview of commonly used simulation systems.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Modeling and Simulation, Bungartz, H.-J., Zimmer, S., Buchholz, M., Pflüger, D., Springer-Verlag, 2014.
2. Simulation Modeling Handbook, A Practical Approach, Christopher A. Chung, CRC Press, 2004.
3. System design, modeling and simulation using Ptolemy II, Claudius Ptolemaeus, , Ver 2.0, Creative Commons Attribution-ShareAlike 3.0 Unported, 2014
4. Applied Simulation Modeling, Andrew F. Seila, Vlatko Ceric, Pandu Tadikamalla, Thomson Learning Inc., 2003.

Multimedia Communications

Credit Hours:	3(3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
CLO-1: CLO-2: CLO-3:. CLO-4:	C	

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Overview of multimedia systems, Audio/Video fundamentals (representation, human perception, equipment and applications). Audio and video compression (e.g., JPEG, MPEG, H.26X, etc.), scalable coding, perceptual audio encoders. Performance comparison of coding algorithms, Algorithms for image and video processing, multimedia programming.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Fred Halsall, "Multimedia Communications: Applications, Networks, Protocols, and Standards", Latest Ed.
2. Puri, "Multimedia Systems, Standards and Networks", Marcel Dekker, Latest Ed.
3. Steve Heath, "Multimedia and Communication Technology", Focal Press, Latest Ed.
4. Bill Whyte, "Multimedia Telecommunication", Chapman and Hall, Latest Ed.

Multivariate Calculus

Credit Hours:	3-0	Prerequisites:	Calculus and Analytical Geometry

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the basic concepts and know the basic techniques of differential and integral calculus of functions of several variables;		
2. Apply the theory to calculate the gradients, directional derivatives, arc length of curves, area of surfaces, and volume of solids;		
3. Solve problems involving maxima and minima, line integral and surface integral, and vector calculus;		
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Functions of Several Variables and Partial Differentiation. Multiple Integrals, Line and Surface Integrals. Green's and Stoke's Theorem. Fourier Series: periodic functions, Functions of any period P-2L, Even & odd functions, Half Range expansions, Fourier Transform; Laplace Transform, Z-Transform.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. *Multivariable Calculus*, 6th edition James, Stewart 2007 Cengage Learning publishers.
2. *Calculus and Analytical Geometry*, 6th edition. Swokowski, Olinick and Pence.1994.Thomson Learning EMEA, Ltd.
3. ***Multivariable Calculus***, 5th edition Howard, A. Albert, H. 1995, John Wiley.

Natural Language Processing

Credit Hours:	3(3,0)	Prerequisites:	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Identify techniques for information retrieval, language translation, and text classification.	C	1
2. List the advantages of using standard corpora. Identify examples of current corpora for a variety of NLP tasks.	C	2
3. Define and contrast deterministic and stochastic grammars, providing examples to show the adequacy of each.	C	3
4. Simulate, apply, or implement classic and stochastic algorithms for parsing natural language.	C	4
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Deterministic and stochastic grammars, Parsing algorithms, CFGs, Representing meaning / Semantics, Semantic roles, Temporal representations, Corpus-based methods, N-grams and HMMs, Smoothing and backoff, POS tagging and morphology, Information retrieval, Vector space model, Precision and recall, Information extraction, Language translation, Text classification, categorization, Bag of words model.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Python Machine Learning, Sebastian Raschka. Publisher: Packt Publishing, 2015.
2. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit Latest Edition, Steven Bird, Ewan Klein and Edward Loper Publisher: O'Reilly Media, 2009.
3. Speech and Language Processing, Latest Edition, Daniel Jurafsky and James H. Martin Publisher: Prentice Hall, 2000.

Numerical Computing			
Credit Hours:	3	**Prerequisites:**	Calculus and Analytical Geometry

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. The student would understand the fundamental concepts of Scientific Programming using programing Language(s)	C	1
2. Use a computer algebra system to investigate and solve mathematical problems relating to integration, differential equations and approximation.	C	2
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Mathematical preliminaries and error analysis, round-off errors and computer arithmetic, Calculate Divided Differences. Use Divided-difference Table. Find Newton's Interpolation Polynomial. Calculate Interpolation with Equally Spaced Data. Find the Difference Table. Calculate, Newton's Forward & Backward Difference Formulae. Use Gauss Formulae. Use Stirling's Interpolation Formula. Use Bessel's Interpolation Formula. Use Everett's Interpolation Formula. Solve Nonlinear Equations. Solve Equations by Bisection Method. Solve Equations by Regula Falsi Method. Solve Equations by Secant Method. Solve Equations by Newton-Raphson Method. Find Fixed Point Iteration. Solve Equations by Jacobi Iterative Methods. Solve Equations by Gauss Seidel Method Calculate Numerical Differentiation. Find Numerical Differentiation Formulae Based on Equally Spaced Data. Find Numerical Differentiation Based on Newton's Forward Differences. Find Numerical Differentiation Based on Newton's Backward Differences. Find Numerical Differentiation Based on Stirling's Formula. Find Numerical Differentiation Based on Bessel's Formula. Find Numerical Differentiation Based on

Lagrange's Formula. Calculate Error Analysis of Differentiation Formulae. Solve Richardson Extrapolation. Calculate Numerical Integration. Use Trapezoidal Rule with Error Term. Use Simpson's 1/3 Rule with Error Term. Use Simpson's 3/8 Rule with Error Term. Use Composite Numerical Integration. Use Composite Trapezoidal Rule. Use Composite Simpson's Rule. Find Richardson's Extrapolation. Find Newton-Cotes Closed Quadrature Formulae.

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Lab Assignments, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. *Numerical Analysis* (9th Edition) by Richard L. Burden, J. Douglas Faires by Brooks/Cole Boston USA, 2011
2. *Numerical Methods for Scientific Computing* by J.H. Heinbockel Trafford Publishing USA, 2006

Object Oriented Analysis & Design

Credit Hours:	3 (3,0)	**Prerequisites:**	Programming Fundamentals

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level[*]**

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Principles of Object Technology. OOP Review. Principles of Modeling. OOA&D Overview. OO Development Process. Requirements Engineering, Analysis, and Specification: Requirements Engineering, Use Cases, Prototyping, Class Models. Interaction Diagrams. Verification and Validation. Architectural and Detailed Design. Class Diagrams. Interaction Diagrams. State Machines and Diagrams. Implementation, Package Diagrams. Activity Diagrams. OO Patterns, Verification and Validation. Note: Students may also be introduced to Object Diagram, Component Diagram, Package Diagram, Deployment Diagram, Network Diagram.

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

1. Applying UML and patterns: An introduction to Object-Oriented Analysis and Design and Iterative Development by Craig Larman, Prentice Hall; 3rd Edition (October 30, 2004). ISBN-10: 0131489062
2. Using UML: Software Engineering with Objects and Components by Perdita Stevens, Addison-Wesley; 2nd Edition (February 13, 2006). ISBN-10: 0321269675
3. Fundamental of Object-Oriented Design in UML by Meiler Page-Jones, Addison Wesley, 2000. ISBN: 020169946X.
4. The Unified Modeling Language User Guide by G. Booch, J. Rambaugh and I. Jakobson, Addison-Wesley Professional; 2nd Edition (2005). ISBN- 10: 0321267974.

Object Oriented Programming			
Credit Hours:	3+1	**Prerequisites:**	Programming Fundamentals

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand principles of object oriented paradigm.	C	2
2. Identify the objects & their relationships to build object oriented solution	C	3
3. Model a solution for a given problem using object oriented principles	C	3
4. Examine an object oriented solution.	C	4
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction to object oriented design, history and advantages of object oriented design, introduction to object oriented programming concepts, classes, objects, data encapsulation, constructors, destructors, access modifiers, const vs non-const functions, static data members & functions, function overloading, operator overloading, identification of classes and their relationships, composition, aggregation, inheritance, multiple inheritance, polymorphism, abstract classes and interfaces, generic programming concepts, function & class templates, standard template library, object streams, data and object serialization using object streams, exception handling.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Starting Out with C++ from Control Structures to Objects, 9th Edition, Tony Gaddis
2. C++ How to Program, 10th Edition, Deitel & Deitel.
3. Object Oriented Programming in C++, 3rd Edition by Robert Lafore
4. Java: How to Program, 9th Edition by Paul Deitel
5. Beginning Java 2, 7th Edition by Ivor Horton
6. An Introduction to Object Oriented Programming with Java, 5th Edition by C. Thomas Wu

Operating Systems

Credit Hours:	3+1	**Prerequisites:**	Data Structures and Algorithms

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. **Understand** the characteristics of different structures of the Operating Systems and identify the core functions of the Operating Systems.	C	2
2. **Analyze** and **evaluate** the algorithms of the core functions of the Operating Systems and explain the major performance issues with regard to the core functions.	C	4,5
3. **Demonstrate** the knowledge in applying system software and tools available in modern operating systems.	C	3

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Operating systems basics, system calls, process concept and scheduling, inter-process communication, multithreaded programming, multithreading models, threading issues, process scheduling algorithms, thread scheduling, multiple-processor scheduling, synchronization, critical section, synchronization hardware, synchronization problems, deadlocks, detecting and recovering from deadlocks, memory management, swapping, contiguous memory allocation, segmentation & paging, virtual memory management, demand paging, thrashing, memory-mapped files, file systems, file concept, directory and disk structure, directory implementation, free space management, disk structure and scheduling, swap space management, system protection, virtual machines, operating system security

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Operating Systems Concepts, 9th edition by Abraham Silberschatz
2. Modern Operating Systems, 4th edition by Andrew S. Tanenbaum
3. Operating Systems, Internals and Design Principles, 9th edition by William Stallings

Operations Research

Credit Hours:	3	Prerequisites:	None

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	Domain	BT Level*
1. Learn the characteristics of different types of decision-making environments, appropriate decision making approaches and tools to be used in each type.		
2. Solve the Transportation Models and Assignment Models.		
3. Understand the basic methodology for the solution of linear programs and integer programs.		

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction to operations research, History of operations research, Applications, Modeling the linear programming, Linear programming, Geometry, Solving the linear programming, the Simplex method, Shadow price, Theory of the simplex method, Duality, Dual theory, Sensitivity analysis, Other algorithms for linear programming, The dual simple method, Big – M method, The tow phase method, The transportation and assignment problems, The transportation problem, A streamlined simplex method for transportation problem, The assignment problem, A special algorithm for the assignment problem, Dynamic programming, Characteristic of dynamic programming, Deterministic dynamic programming, Integer programming, Prototype examples, BIP applications and formulation examples, Some perspectives on solving integer programming problems, The branch-and-cut approach to solve BIP problems, The incorporation of constraint programming.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Frederick S. Hiller, Gerald J. Lieberman, Introduction to Operations Research, 9th Edition, English, McGraw-Hill, 2010.
2. W. Winston, Operations Research, Duxbury Press.
3. Operations Research: Applications and Algorithms, Wayne L Winston, Indian University, 4th edition, 2004

Pakistan Studies

Credit Hours:	3	Prerequisites:	None

Course Learning Outcomes (CLOs):		
	Domain	BT Level*

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Historical background of Pakistan: Muslim society in Indo-Pakistan, the movement led by the societies, the downfall of Islamic society, the establishment of British Raj- Causes and consequences. Political evolution of Muslims in the twentieth century: Sir Syed Ahmed Khan; Muslim League; Nehru; Allama Iqbal: Independence Movement; Lahore Resolution; Pakistan culture and society, Constitutional and Administrative issues, Pakistan and its geo-political dimension, Pakistan and International Affairs, Pakistan and the challenges ahead.

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. The Emergence of Pakistan, Chaudary M., 1967
2. The making of Pakistan, Aziz. 1976
3. A Short History of Pakistan, I. H. Qureshi, ed., Karachi, 1988

Parallel and Distributed Computing

Credit Hours:	3	**Prerequisites:**	Operating Systems

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Learn about parallel and distributed computers.		
2. Write portable programs for parallel or distributed architectures using Message-Passing Interface (MPI) library		
3. Analytical modelling and performance of parallel programs.		
4. Analyze complex problems with shared memory programming with openMP.		

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Asynchronous/synchronous computation/communication, concurrency control, fault tolerance, GPU architecture and programming, heterogeneity, interconnection topologies, load balancing, memory consistency model, memory hierarchies, Message passing interface (MPI), MIMD/SIMD, multithreaded programming, parallel algorithms & architectures, parallel I/O, performance analysis and tuning, power, programming models (data parallel, task parallel, process-centric, shared/distributed memory), scalability and performance studies, scheduling, storage systems, synchronization, and tools (Cuda, Swift, Globus, Condor, Amazon AWS, OpenStack, Cilk, gdb, threads, MPICH, OpenMP, Hadoop, FUSE).

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Distributed Systems: Principles and Paradigms, A. S. Tanenbaum and M. V. Steen, Prentice Hall, 2nd Edition, 2007
2. Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet, K Hwang, J Dongarra and GC. C. Fox, Elsevier, 1st Ed.

Probability & Statistics

Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction to Statistics and Data Analysis, Statistical Inference, Samples, Populations, and the Role of Probability. Sampling Procedures. Discrete and Continuous Data. Statistical Modeling. Types of Statistical Studies. Probability: Sample Space, Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability, Independence, and the Product Rule, Bayes' Rule. Random Variables and Probability Distributions. Mathematical Expectation: Mean of a Random Variable, Variance and Covariance of Random Variables, Means and Variances of Linear Combinations of Random Variables, Chebyshev's Theorem. Discrete Probability Distributions. Continuous Probability Distributions. Fundamental Sampling Distributions and Data Descriptions: Random Sampling, Sampling Distributions, Sampling Distribution of Means and the Central Limit Theorem. Sampling Distribution of S2, t-Distribution, F-Quantile and Probability Plots. Single Sample & One- and Two-Sample Estimation Problems. Single Sample & One- and Two-Sample Tests of Hypotheses. The Use of P-Values for Decision Making in Testing Hypotheses (Single Sample & One- and Two-Sample Tests), Linear Regression and Correlation. Least Squares and the Fitted Model, Multiple Linear Regression and Certain, Nonlinear Regression Models, Linear Regression Model Using Matrices, Properties of the Least Squares Estimators.

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

1. Probability and Statistics for Engineers and Scientists by Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying E. Ye, Pearson; 9th Edition (January 6, 2011). ISBN-10: 0321629116
2. Probability and Statistics for Engineers and Scientists by Anthony J. Hayter, Duxbury Press; 3rd Edition (February 3, 2006), ISBN-10:0495107573
3. Schaum's Outline of Probability and Statistics, by John Schiller, R. Alu Srinivasan and Murray Spiegel, McGraw-Hill; 3rd Edition (2008). ISBN-10:0071544259

Professional Practices

Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Computing Profession, Computing Ethics, Philosophy of Ethics. The Structure of Organizations, Finance and Accounting, Anatomy of a Software House, Computer Contracts, Intellectual Property Rights, The Framework of Employee Relations Law and Changing Management Practices, Human Resource Management and IT, Health and Safety at Work, Software Liability, Liability and Practice, Computer Misuse and the Criminal Law, Regulation and Control of Personal Information. Overview of the British Computer Society Code of Conduct, IEEE Code of Ethics, ACM Code of Ethics and Professional Conduct, ACM/IEEE Software Engineering Code of Ethics and Professional Practice. Accountability and Auditing, Social Application of Ethics.

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

1. Professional Issues in Software Engineering by Frank Bott, Allison Coleman, Jack Eaton and Diane Rowland, CRC Press; 3rd Edition (2000). ISBN-10: 0748409513
2. Computer Ethics by Deborah G. Johnson, Pearson; 4th Edition (January 3, 2009). ISBN-10: 0131112414
3. A Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet (3rd Edition) by Sara Baase, Prentice Hall; 3rd Edition (2008). ISBN-10: 0136008488
4. Applied Professional Ethics by Gregory R. Beabout, University Press of America (1993). ISBN-10: 0819193747.

Programming Fundamentals

Credit Hours:	3+1	**Prerequisites:**	None

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. **Understand** basic problem solving steps and logic constructs	C	2
2. **Apply** basic programing concepts	C	3
3. Design and implement algorithms to **solve** real world problems.	C	3

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction to problem solving, a brief review of Von-Neumann architecture, Introduction to programming, role of compiler and linker, introduction to algorithms, basic data types and variables, input/output constructs, arithmetic, comparison and logical operators, conditional statements and execution flow for conditional statements, repetitive statements and execution flow for repetitive statements, lists and their memory organization, multi-dimensional lists, introduction to modular programming, function definition and calling, stack rolling and unrolling, string and string operations, pointers/references, static and dynamic memory allocation, File I/O operations

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Starting out with Python, 4th Edition, Tony Gaddis.
2. Starting out with Programming Logic & Degins, 4th Edition, Tony Gaddis,
3. The C Programming Language, 2nd Edition by Brian W. Kernighan, Dennis M. Ritchie
4. Object Oriented Programming in C++ by Robert Lafore
5. Introduction to Computation and Programming Using Python: With Application to Understanding Data, 2nd Edition by Guttag, John
6. Practice of Computing Using Python, 3rd Edition by William Punch & Richard Enbody
7. C How to Program, 7th Edition by Paul Deitel & Harvey Deitel
8. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman

Real Time Systems

Credit Hours:	3(3,0)	**Prerequisites:**	Software Engineering

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the issues and basic concepts of real-time software development.	C	1
2. Demonstrate the ability to develop embedded real-time software using appropriate software methods and tools.	C	2
3. Analyze the timing performance of a real-time software design using real-time analysis tools.	C	4
4. Apply real-time software engineering knowledge in developing a medium to complex program	C	3

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction to Real-Time Systems, Categories, Characteristics and challenges, Requirement Specification and Design, Design fundamentals, Elements of modular design, Concurrency, Real-time & other application areas, Real-Time Operating Systems, Memory management, Fundamental of microprocessor based systems, Input-output interfacing technique, Real-time programming, Real-Time Analysis, Schedulability analysis, Scheduling policies, Designing with rate-monotonic analysis

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Software Engineering for Real-Time Systems, Cooling J., Addison-Wesley.
2. Real-time Systems and Programming Languages, 2nd Edition, Burns A., Wellings A. J., Addison Wesley, UK.
3. Principles of Concurrent and Distributed Programming. Ben-Ari M., Addison-Wesley, 2006.

Semantic Web

Credit Hours:	3(3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the concept structure of the semantic web technology and how this technology revolutionizes the World Wide Web and its uses.	C	1
2. Understand the concepts of metadata, semantics of knowledge and resource, ontology, and their descriptions in XML-based syntax and web ontology language (OWL).	C	2
3. Describe logic semantics and inference with OWL.	C	2
4. Use ontology engineering approaches in semantic applications program semantic applications with Java API.	C	4

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction to the semantic web, introduction to ontologies, ontology languages for the semantic web, Resource Description Framework (RDF), lightweight ontologies: RDF Schema, Web Ontology Language (OWL), query language for RDF: SPARQL, Ontology Engineering, Semantic web and Web 2.0 and applications of Semantic Web.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Build Flexible Applications with Graph Data, Toby Segaran, Colin Evans, Jamie Taylor,302 pages O'Reilly Media, 2009
2. Foundations of Semantic Web Technologies, Pascal Hitzler, Markus Krotzsch, Sebastian Rudolph,
3. Introduction to the Semantic Web and Semantic Web Services, Liyang Yu, Chapman and Hall/CRC, 2007

Software Construction and Development

Credit Hours:	3 (2-1)	**Prerequisites:**	Software Design and Architecture

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	Domain	BT Level*
1. Understand the role of design and its major activities within the OO software development process, with focus on the Unified process	C	1
2. Develop Object-oriented design models and refine them to reflect implementation details	C	3
3. Evaluate different architectures for a medium size software.	C	4
4. Implement design model using an object-oriented programming language.	C	4

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Software development process, Software engineering process infrastructure, Software engineering process improvement, Systems engineering life cycle models, Process implementation, Levels of process definition, Life cycle model characteristics, Individual and team software process, Lehman’s Laws, code salvaging, and configuration management. Martin Fowler’s refactoring concepts and their application to small projects. Apply Michael Feathers’ “legacy code” concepts. Exception handling, making methods robust by having them check their inputs sent from calling objects. Software configuration management, Release management, Software configuration management processes, Software deployment processes, Distribution and backup, Evolution processes and activities, Basic concepts of evolution and maintenance, Working with legacy systems, Refactoring, Error handling, exception handling, and fault tolerance. Personal reviews (design, code, etc.), Peer reviews (inspections, walkthroughs, etc.).

Teaching Methodology:

Lecturing, Written and Lab Assignments, Project, Report Writing

Course Assessment:

Reference Materials:

1. Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, Prentice Hall, 2008.
2. The Pragmatic Programmer: From Journeyman to Master, Andrew Hunt and David Thomas, Addison-Wesley Professional, 1999.
3. Working Effectively with Legacy Code, Michael C. Feathers. Pearson Education, Prentice-Hall, 2004.
4. Refactoring: Improving the Design of Existing Code, Martin Fowler, Addison-Wesley Professional. 1999.

Software Design and Architecture			
Credit Hours:	3 (2-1)	**Prerequisites:**	Software Requirement Engineering

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the role of design and its major activities within the OO software development process, with focus on the Unified process.	C	1
2. Comprehend the advantages of consistent and reliable software design.	C	2
3. Design OOD models and refine them to reflect implementation details	C	3
4. Apply and use UML to visualize and document the design of software systems.	C	4
5. Implement the design model using an object-oriented programming language.	C	5

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Software Design Concepts, Design principles, Object-Oriented Design with UML, System design and software architecture, Object design, Mapping design to code, User interface design, Persistent layer design, Web applications design, State machine diagrams and modeling, Agile software engineering, Design Patterns, Exploring inheritance, Interactive systems with MVC architecture, Software reuse. Architectural design issues, , Software Architecture, Architectural Structures & Styles-, Architectural Patterns, Architectural & Design Qualities, Quality Tactics, Architecture documentation, Architectural Evaluation, Model driven development.

Teaching Methodology:

Lecturing, Written and Lab Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home and Lab Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Software Engineering: A Practitioner's Approach, Roger S. Pressman, Bruce R. Maxim, 8th Ed, McGraw-Hill Education, 2015.
2. Object-Oriented Analysis, Design and Implementation, Brahma Dathan, Sarnath Ramnath, 2nd Ed, Universities Press, India, 2014.
3. Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures, Hassan Gomaa, Cambridge University Press, 2011.
4. Head First Design Patterns, Eric Freeman, Elisabeth Freeman, Kathy Sierra and Bert Bates, O'Reilly Media, Inc. 2004.

Software Engineering Economics

Credit Hours:	3(3,0)	**Prerequisites:**	Software Engineering

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Overview economic analysis techniques and their applicability to software engineering	C	2
2. Develop software cost estimation skills using industry standards.	C	3
3. Critically evaluate and discuss the issues in cost estimation of different applications in the real world with course participants and learners.	C	3

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Programming aspects, economic aspects, human relations aspects, software trends: cost, social impact, the plurality of SE Means, The GOALS Approach to Software Engineering, The Software Work Breakdown Structure (WBS), Software Maintenance, introduction to COCOMO, definitions and assumptions, development effort and schedule, phase distribution, The Raylaigh Distribution, interpolation, basic software maintenance effort estimation. Performance Models, Optimal Performance, Sensitivity Analysis, Cost-Effectiveness Models. Cost Drivers: Project Attributes–Modern Programming Practices, Use of Software Tools, Schedule Constraint.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Software Engineering Economics and Declining Budgets by Pamela T. Geriner, Thomas R. Gulledge, William P. Hutzler, Springer Verlag, 2012
2. Estimating Software Costs: Bringing Realism to Estimating, Capers Jones, McGraw-Hill Osborne Media; 2nd Edition, 2007.
3. Software Cost Estimation and Sizing Methods, Issues, and Guidelines, Shari Lawrence Pfleeger, Rand Publishing, 2005.

Software Metrics

Credit Hours:	3(3,0)	**Prerequisites:**	Software Engineering

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explains how quantitative and empirical methods are applied to software engineering problems	C	2
2. Presents the fundamentals of measurement, experimentation, data collection and analysis	C	3
3. Critically evaluate and discuss different software matrices of different applications in the real world with course participants and learners	C	3
4. Have a working knowledge of software size measurement (Function Point counting, etc.)	C	4

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Overview of software metrics; Basics of measurements; Goal-based framework for software measurement; Software measure classification; Empirical investigation, principles and techniques; Formal experiments: Planning, principles, types and selection; Measuring internal product attributes: size and structure; Measuring cost and effort; Measuring external product attributes: quality and reliability; Software test metrics; Object-oriented metrics

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Software Metrics: A Rigorous and Practical Approach, (3rd ed.), N.E. Fenton and J. Bieman, CRC Press, 2014,
2. Software Metrics: A Guide to Planning, Analysis, and Application, C. Ravindranath Pandian, Auerbach Publications, CRC Press Company, 2004.
3. Metrics and Models in Software Quality Engineering, Stephen H. Kan, 2nd ed., Addison-Wesley Professional, 2002.

Software Project Management			
Credit Hours:	3 (3,0)	**Prerequisites:**	Software Engineering

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain principles of the project lifecycle and how to identify opportunities to work with learners on relevant and appropriate project scenarios to share this understanding	C	2
2. Critically evaluate and discuss the issues around project management and its application in the real world with course participants and learners	C	3
3. Choose project management techniques for IT projects to initiate, plan, execute and evaluate a project and work in teams to create a project plan for a project scenario that includes key tasks, critical path, dependencies and a realistic timeline.	C	4
4. Present strategies for gaining confidence in managing projects through simple project planning examples.	C	4
* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction to Software Project Management, Project Management concepts, Project Management Tools, PMI‘s Knowledge areas, PMI Framework, PMI Process Groups. Understanding Organizations. Project Planning, Project Evaluation, Selection of an Appropriate Approach in Project, Software Effort Estimation, Activity Planning, Risk Management, Evaluating the Risks to the Schedule, Risk Control, Configuration Management and Maintenance, Environment for Configuration Control, Resource Allocation, Monitoring & Control, Review and Evaluation, Challenges of Outsourcing in Project Management

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Software Project Management, Bob Hughes and Mike Cotterell, McGraw-Hill Education; 5th Edition (2009).
2. A Guide to the Project Management Body of Knowledge, 5th Edition (PMBOK Guides),
3. Mastering Software Project Management: Best Practices, Tools and Techniques, Murali K. Chemuturi and Thomas M. Cagley Jr., J. Ross Publishing, 2010
4. Effective Project Management: Traditional, Agile, Extreme, Robert K. Wysocki, Wiley; 6th Edition, 2011

Software Quality Engineering

Credit Hours:	3 (3,0)	**Prerequisites:**	Software Engineering

Course Learning Outcomes (CLOs):

At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Outline software testing and software quality assurance principles.	C	1
2. Prepare test case and test suites for completely testing all aspects of a system under test (SUT)	C	3
3. Analyze which of the software testing techniques are relevant for a particular case and know software reliability analysis tools and techniques.	C	4
4. Compile findings of a quality assurance cycle.	C	5

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Software Quality, Software Quality Attributes, Quality Engineering., Testing: Concepts, Issues, and Techniques, Software testing lifecycle., Testing Scopes., Testing Approaches., Testing Concepts., Test Planning Process, Introduction to testing process, Requirement of software test planning, Testing documentation, Reporting and historical data recording., Software testing techniques, Testing philosophies , Testing strategies, Model based testing, Software testing techniques, Testing using models, Domain and combinatorial testing, Unit and integration testing, Acceptance testing, Test automation, Slicing, Software reliability models and engineering, Introduction, Exponential model., Reliability growth models, Modeling process, Software inspections, Software reviews, Inspection checks and metrics, Quality Models, Models for quality assessment, Product quality metrics, Quality Measurements, In-Process metrics for software testing, In-Process quality management, Effort/outcome models, System testing, Introduction to sub-system testing, From functional to system aspects of testing, System testing, Introduction to system testing, Scenarios development, System testing, Use-cases for testing, Specification-based testing, Open issues on software testing

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

3. Paul Jorgensen, Software Testing, A Craftsman's Approach, 4th Ed. CRC Press, Taylor and Francis Group, 2015
4. Bernard Homes, Fundamentals of Software Testing, ISTE, Wiley, 2012
5. Software Engineering, “Ian Sommerville, 9th Edition, Addison Wesley, 2011

Software Re-Engineering

Credit Hours:	3 (3,0)	**Prerequisites:**	Software Construction and Development

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain the concepts and technique of software re-engineering.	C	1
2. Apply reengineering techniques to maintain and modify software systems	C	3
3. Analyze and understand maintenance related problems associated with object oriented software systems.	C	4
4. Able to perform complex design reengineering and reverse engineering problems.	C	5

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Salient topics include the terminology and the processes pertaining to software evolution, fundamental re-engineering techniques to modernize legacy systems including source code analysis, architecture recovery, and code restructuring, software refactoring strategies, migration to Object Oriented platforms, quality issues in re-engineering processes, migration to network-centric environments, and software integration, reverse engineering, program comprehension, source code transformation and refactoring strategies, software maintenance and re-engineering economics.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Re-engineering legacy software, David Lorge Parnas, Chris Birchall, Safari Books, Shelter Island, NY, 2016
2. Reengineering, Priyadarshi Tripathy and Kshirasagar Naik, John Wiley & Sons, Inc.2015
3. Software Maintenance and Evolution: a Roadmap, K.H.Bennett and V.T Rajlich, The Future of Software Engineering, ACM Press 2000.

Software Requirements Engineering

Credit Hours:	3 (3,0)	**Prerequisites:**	Software Engineering

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Describe the requirements engineering process	C	1
2. Effectively analyze software requirements for the development of cost-effective and efficient technical solutions.	C	4
3. Prepare both functional and non-functional requirements along with validation for a medium-size software system.	C	3
4. Document effective requirements in Software Requirements Specification (SRS) using clear, unambiguous requirements.	C	3

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction to Requirements Engineering, Software Requirements, classification of requirements, Requirements process, Levels/layers of requirements, Requirement characteristics, Analyzing quality requirements, Software requirements in the context of systems engineering, Requirement evolution, requirement traceability, requirement prioritization, trade-off analysis, risk analysis and impact analysis, Requirement management, interaction between requirement and architecture, Requirement elicitation, elicitation sources and techniques, Requirement specification and documentation, specification sources and techniques, Requirements validation and techniques, Management of Requirements, Introduction to Management, Requirements Management Problems , Managing Requirements in an Acquisition Organization, Supplier Organizations, Product Organizations, Requirements engineering for agile methods.

Teaching Methodology:

Lecturing, Written and Lab Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home and Lab Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Software Requirements, Wiegers K. &Beatty J., 3rd Ed. Microsoft Press, 2013
2. Requirements Engineering, Elizabeth Hull, Ken Jackson and Jeremy Dick. 3rd Ed, Springer-Verlag London Limited, 2011.
3. Requirements Engineering and Management for Software Development Projects, Chemuturi M., Springer New York, 2013.

Stochastic Processes			
Credit Hours:	3 (3,0)	**Prerequisites:**	Probability and Statistics

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Define basic concepts from the theory of Markov chains and present proofs for the most important theorems.	C	1
2. Compute probabilities of transition between states and return to the initial state after long time intervals in Markov chains.	C	2
3. Derive differential equations for time continuous Markov processes with a discrete state space.	C	3
4. Solve differential equations for distributions and expectations in time continuous processes and determine corresponding limit distributions.	C	4
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Discrete Markov chains, classification of states, first passage and recurrence times, absorption problems, stationary and limiting distributions. Chapman-Kolmogorov equations, Long run behavior of Markov chains, Absorption probabilities and expected times to absorption, Statistical aspects of Markov chains, The mover-stayer model, Application of a Markov chain and mover-stayer model to modeling repayment behavior of bank loans' grantees. Markov Processes in continuous time: Poisson processes, birth-death processes. Poisson process The Kolmogorov differential equations, Limiting behavior of continuous time Markov chains The Q matrix, forward and backward differential equations, imbedded Markov Chain, stationary distribution. renewal theory, Brownian Motion and its generalizations, Discrete time martingales, Conditional expectation, Definition of a martingale and examples, Optional stopping theorem, Stochastic calculus

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Introduction to Probability Models, 11th Ed, Sheldon M. Ross, Academic Press 2014.
2. Essentials of stochastic processes, Durrett, Richard. Springer Science & Business Media, 2nd Ed, 2012.
3. Introduction to Stochastic Processes, 2nd Ed, G.F. Lawler, Chapman and Hall, Probability Series, 2006

System and Network Administration

Credit Hours:	3 (3,0)	**Prerequisites:**	Operating System

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction To System Administration. SA Components. Server Environment (Microsoft and Linux). Reliable Products, Server Hardware Costing, Maintenance Contracts and Spare Parts, Maintaining Data Integrity, Client Server OS Configuration, Providing Remote Console Access. Comparative Analysis of OS: Important Attributes, Key Features, Pros and Cons. Linux Installation and Verification, Configuring Local Services and Managing Basic System Issues. Administer Users and Groups. Software Management. Managing Network Services and Network Monitoring Tools. Boot Management and Process Management. IP Tables and Filtering. Securing Network Traffic. Advanced File Systems and Logs. Bash Shell Scripting. Configuring Servers (FTP, NFS, Samba, DHCP, DNS and Apache).

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

1. The Practice of System and Network Administration, Second Edition by Thomas Limoncelli, Christina Hogan and Strata Chalup, Addison-Wesley Professional; 2nd Edition (2007). ISBN-10: 0321492668
2. Red Hat Enterprise Linux 6 Bible: Administering Enterprise Linux Systems by William vonHagen, 2011
3. Studyguide for Practice of System and Network Administration by Thomas A. Limoncelli, Cram101; 2nd Edition (2011). ISBN-10: 1428851755
4. Networking Systems Design and Development by Lee Chao, CRC Press; 1st Edition (December 21, 2009). ISBN-10: 142009159X (TB2)

Systems Programming

Credit Hours:	3(3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
CLO-1: CLO-2: CLO-3:. CLO-4:	C	
* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction to the Microsoft Windows ® Operating System, File Processing, Memory Management, Memory Mapped Files and DLLs, Process management, Threads and scheduling, Thread synchronization, Inter-process Communication, Input/Output, Device Drivers (USB or Parallel Port), File System Drivers, Filter Drivers

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Windows System Programming 3rd edition, Johnson M. Hart, Addison Wesley
2. The Windows NT Device driver book 2nd edition, Art Baker, Prentice Hall.

Technical & Business Writing			
Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Overview of technical reporting, use of library and information gathering, administering questionnaires, reviewing the gathered information; Technical exposition; topical arrangement, exemplification, definition, classification and division, casual analysis, effective exposition, technical narration, description and argumentation, persuasive strategy, Organizing information and generation solution: brainstorming, organizing material, construction of the formal outline, outlining conventions, electronic communication, generation solutions. Polishing style: paragraphs, listening sentence structure, clarity, length and order, pomposity, empty words, pompous vocabulary, document design: document structure, preamble, summaries, abstracts, table of contents, footnotes, glossaries, cross-referencing, plagiarism, citation and bibliography, glossaries, index, appendices, typesetting systems, creating the professional report; elements, mechanical elements and graphical elements. Reports: Proposals, progress reports, Leaflets, brochures, handbooks, magazines articles, research papers, feasibility reports, project reports, technical research reports, manuals and documentation, thesis. Electronic documents, Linear verses hierarchical structure documents.

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Report Writing, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

1. Technical Report Writing, by Pauley and Riordan, Houghton Mifflin Company, 8th Edition.
2. Effective Technical Communication by Ashraf Rizvi, Tata McGraw-Hill.

Theory of Automata

Credit Hours:	3	**Prerequisites:**	None

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain and manipulate the different concepts in automata theory and formal languages such as formal proofs, automata, regular expressions, Turing machines etc;		
2. Prove properties of languages, grammars and automata with rigorously formal mathematical methods		
3. Design of automata, RE and CFG		
4. Transform between equivalent NFAs, DFAs and REs		
5. Define Turing machines performing simple tasks.		
6. Differentiate and manipulate formal descriptions of languages, automata and grammars with focus on regular and context-free languages, finite automata and regular expressions.		
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Finite State Models: Language definitions preliminaries, Regular expressions/Regular languages, Finite automata (FAs), Transition graphs (TGs), NFAs, Kleene's theorem, Transducers (automata with output), Pumping lemma and non-regular language

Grammars and PDA: CFGs, Derivations, derivation trees and ambiguity, Simplifying CFLs, Normal form grammars and parsing, Decidability, Context sensitive languages, grammars and linear bounded automata (LBA), Chomsky's hierarchy of grammars

Turing Machines Theory: Turing machines, Post machine, Variations on TM, TM encoding, Universal Turing Machine, Defining Computers by TMs.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Introduction to computer theory, Daniel I. A. Cohen, 2nd Edition
2. Automata, Computability and Complexity: Theory and Applications, by Elaine Rich, 2011
3. An Introduction to Formal Languages and Automata, by Peter Linz, 4th edition, Jones & Bartlett Publishers, 2006
4. Theory of Automata, Formal Languages and Computation, by S. P. Eugene, Kavier, 2005, New Age Publishers

Theory of Programing Languages

Credit Hours:	3	**Prerequisites:**	Programming Fundamentals

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. The better understating the underlying theory of programming languages	C	1
2. Enable a student to choose the appropriate Language for a Project	C	2
3. Learning of formal semantics design for a programming Languages	C	2
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction: Models of Computation, Syntax and Semantics, Pragmatics, Language Design Principles. Syntax and Semantics: Context-Free Grammars, Regular Expressions, Attribute Grammars and Static Semantics, Algebraic Semantics, Axiomatic Semantics, Denotational Semantics. BNF grammars and Syntax, Operational Equivalence, Abstraction and Generalization, Expressions, Assignment Statement, and Control Structures, Functional Programming: The Lambda Calculus, Operational Semantics, Reduction Order, Recursive Functions, Logic Programming, Inference Engine, Concurrency.

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Lab Assignments, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Concepts of Programming Languages, Robert W. Sebesta, 10th edition, 2012
2. Scott, Michael L., Programming Language Pragmatics, 2nd edition, 2006
3. Theory Introduction to Programming Languages, by Anthony A. Aaby, 2004
4. Principles of Programming Languages by Mike Grant Zachary Palmer Scott Smith, John Hopkins University 2016.

Virtual Systems and Services

Credit Hours:	3	**Prerequisites:**	Programming Fundamentals

Course Learning Outcomes (CLOs):

	Domain	**BT Level***

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

This course will investigate the current state of virtualization in computing systems. Virtualization at both the hardware and software levels will be examined, with emphasis on the hypervisor configurations of systems such as Vmware, Zen and Hyper-V. The features and limitations of virtual environments will be considered, along with several case studies used to demonstrate the configuration and management of such systems. Para-virtualized software components will be analyzed and their pros and cons discussed. Processor and peripheral support for virtualization will also be examined, with a focus on emerging hardware features and the future of virtualization.

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Lab Assignments, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

Handbook of Virtual Environments: Design, Implementation, and Applications (Human Factors and Ergonomics), Edited by Kay M Stanney, Lawrence Erlbaum Associates
Virtual Reality Technology by GRIGORE

Visual Programming

Credit Hours:	3(3,0)	**Prerequisites:**	Object Oriented Programming

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Use the different elements of a visual programming language as building blocks to develop correct, coherent programs.	C	1
2. Program using the fundamental software development process, including design, coding, documentation, testing, and debugging.	C	3
3. Analyze problems, develop conceptual designs that solve those problems, and transform those designs to Visual Programs.	C	4

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Visual Programming Basics; Introduction to Events; Fundamentals of Event-driven Programming, message handling, user interfaces, graphics device interface, painting and drawing, windows management, input devices, resources, string and menu resource, dialogs and windows controls, common controls, dynamic link libraries, threads and synchronization, network programming, Building Class Libraries at the Command Line, Class Libraries, Using References, Assemblies, Private Assembly Deployment, Shared Assembly Deployment, Configuration Overview, Configuration Files, Programmatic Access to Configuration, Using SDK Tools for Signing and Deployment, Metadata, Reflection, Late Binding, Directories, Files, Serialization, Attributes, Memory Management and Garbage Collection, Threading and Synchronization, Asynchronous Delegates, Application Domains, Marshal by Value, Marshal by Reference, Authentication and Authorization, Configuring Security, Code Access Security, Code Groups, Evidence, Permissions, Role-Based Security, Principals and Identities, Using Data Readers, Using Data Sets, Interacting with XML Data, Tracing Event Logs, Using the Boolean Switch and Trace Switch Classes, Print Debugging Information with the Debug Class, Instrumenting Release Builds with the Trace Class, Using Listeners, and Implementing Custom Listeners.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Visual C#: How to Program, Deitel and Deitel, 6/e Edition, Prentice Hall / Pearson Education, 2017.
2. Programming in C# .NET, J.C. Bradley, A.C. Millspaugh, McGraw-Hill, 2014
3. Microsoft Visual C# 2013 Step by Step (Step by Step Developer), Sharp, J., 1st Edition (2013), Microsoft Press.

Web Engineering

Credit Hours:	3 (3,0)	**Prerequisites:**	Programming Fundamentals

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level**[*]
CLO-1: Discuss how web standards impact software development.	C	1
CLO-2: Describe the constraints that the web puts on developers.	C	2
CLO-3: Design and Implement a simple web application.	C	4
CLO-4: Review an existing web application against a current web standard.	C	4

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Web programming languages (e.g., HTML5, CSS 3, Java Script, PHP/JSP/ASP.Net), Design principles of Web based applications, Web platform constraints, Software as a Service (SaaS), Web standards, Responsive Web Design, Web Applications, Browser/Server Communication, Storage Tier, Cookies and Sessions, Input Validation, Full stack state management, Web App Security - Browser Isolation, Network Attacks, Session Attacks, Large scale applications, Performance of Web Applications, Data Centers, Web Testing and Web Maintenance.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Web Engineering, Rajiv Chopra, Prentice-Hall of India, 2016
2. Web Engineering, Emilia Mendes and Nile Mosley, Springer Verlag, 2010.
3. Web Engineering: A Practitioners’ Approach, Roger S. Pressman, McGraw Hill, 2008.
4. Dynamic HTML: The Definitive Reference: A Comprehensive Resource for XHTML, CSS, DOM, JavaScript 3rd Edition, O'Reilly Media 2007.
5. JavaScript: The Definitive Guide, 8th Edition, David Flanagan. O'Reilly Media. 2014.

Web Technologies			
Credit Hours:	4 (3,1)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

Introduction to Web Applications, TCP/IP Application Services. Web Servers: Basic Operation, Virtual hosting, Chunked transfers, Caching support, Extensibility. SGML, HTML5, CSS3. XML Languages and Applications: Core XML, XHTML, XHTM MP. Web Service: SOAP, REST, WML, XSL. Web Services: Operations, Processing HTTP Requests, Processing HTTP Responses, Cookie Coordination, Privacy and P3P, Complex HTTP Interactions, Dynamic Content Delivery. Server Configuration. Server Security. Web Browsers Architecture and Processes. Active Browser Pages: JavaScript, DHTML, AJAX. JSON, Approaches to Web Application Development. Programing in any Scripting language. Search Technologies. Search Engine Optimization. XML Query Language, Semantic Web, Future Web Application Framework.

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

1. Web Application Architecture: Principles, protocols and practices by Leon Shklar and Richard Rosen, Wiley; 2nd Edition (May 5, 2009). ISBN-10:047051860X
2. Web Technologies: A Computer Science Perspective by Jeffrey C. Jackson, Prentice Hall; 1st Edition (August 27, 2006). ISBN-10:0131856030

NCRC Computing – 2017

MS Course Outlines

Master Courses’ List

Course Title	**Page No.**
Advanced Algorithm Analysis	
Advanced Formal Methods	
Advanced Human-Computer Interaction	
Advanced Requirements Engineering	
Advanced Software Project Management	
Advanced Software System Architecture	
Advanced Topics in Applied Cryptography	
Agent Based Modeling	
Agile Software Development	
Applied Cryptography	
Big Data Analytics	
Complex Networks	
Component Based Software Engineering	
Cryptography	
Database Security	
Deep Learning	
Distributed Data Processing	
Empirical Software Engineering	
Information Privacy and Security	
Machine Learning	
Management & Organizational Behavior	
Natural Language Processing	
Quantum Computing and Information security	
Quantum Cryptography	
Reliability Engineering	
Requirements Engineering	
Research Methodology	
Research Methods	
Securing the Internet of Things	
Security Management	
Security Testing	
Software Configuration Management	
Software Measurement and Metrics	
Software Process Management & Metrics	
Software Project Management	
Software Quality Assurance	
Software Risk Management	
Software Testing and Quality Assurance	
Statistical and Mathematical Methods For Data Science	
Tools and Techniques in Data Science	
Trusted Computing	
Wireless Security	

DETAIL OF COURSES

Advanced Algorithm Analysis

Credit Hours:	3	**Prerequisites:**	Data Structures and Algorithms

Course Content:

Advanced algorithm analysis including the introduction of formal techniques and the underlying mathematical theory. NP-completeness; Search Techniques; Randomized Algorithms. Heuristic and Approximation Algorithms; Topics include asymptotic analysis of upper and average complexity bounds using big-O, little-o, and theta notation. Fundamental algorithmic strategies (brute -force, greedy, divide-and-conquer, backtracking, branch-and-bound, pattern matching, and numerical approximations) are covered. Also included are standard graph and tree algorithms. Additional topics include standard complexity classes, time and space tradeoffs in algorithms, using recurrence relations to analyze recursive algorithms, non-computable functions, the halting problem, and the implications of non-computability. Algorithmic animation is used to reinforce theoretical results. Upon completion of the course, students should be able to explain the mathematical concepts used in describing the complexity of an algorithm, and select and apply algorithms appropriate to a particular situation.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Approximation Algorithms, By Vijay V. Vazirani, Springer, 2004.
2. Introduction to Algorithms, By Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, 2nd edition, Published by MIT Press, 2001.
3. Algorithms and Theory of Computation Handbook, By Mikhail J. Atallah Contributor Mikhail J. Atallah, CRC Press, 1998.

Advanced Formal Methods

Credit Hours:	**3**	**Prerequisites:**	None

Course Content:

Introduction to formal methods and specification. State-Based Formal Methods. Transformational systems. Traditional approaches. Z specification. Formal development cycle. Temporal Specification: reactive systems, syntax and semantics of temporal logic, temporal specification of reactive systems (safety, aliveness, fairness). Model Checking: Generating finite models, Analysis of a simple model checking algorithm. Symbolic model checking. Overview of reduction methods. Spin and Promela. Case study and practical verification of properties. Current research topics based on Formal Methods.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Paper

Reference Materials:

1. *Z: An Introduction to Formal Methods* by Antoni Diller, 2nd Edition, John Wiley & Sons, Inc.,1994

Advanced Human-Computer Interaction

Credit Hours:	**3**	**Prerequisites:**	None

Course Content:

Introduction to HCI. Importance of usable and useful software products. The theories of HCI. How to evaluate/develop software products. How to apply theoretical results from HCI research to software products. How to conduct their own research about aspects of usability and user experience. Concepts of Human Computer Interaction. The psychology of usable things. Usability Engineering. Prototypes. Usability inspection methods. Usability testing methods. Usability in practice. User Experience (UX). Web Usability. Mobile Usability. Mobile User Experience. Site objectives and user needs. Information architecture. Information and navigation design. Implementation and optimization. Experiments and HCI guidelines. Current research topics in Human-Computer Interaction.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Paper

Reference Materials:

1. *About Face: The Essentials of Interaction Design*, Alan Cooper, Robert Reimann, David Cronin, Christopher Noessel, Wiley, 4th Edition, 2014.
2. *Designing the User Interface*, Ben Shneiderman and Catherine Plaisant, Pearson, 5th Edition, 2013.
3. *Research Methods in Human-Computer Interaction*, Lazar, Feng, Hochheiser, Wiley, 2010.

Advanced Requirements Engineering

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Software Requirements Fundamentals: Product and process requirements, Functional and non-functional requirements, Emergent properties, Quantifiable requirements, System and software requirements. Requirements Process: Process models, Process actors, Process support and management, Process quality and improvement. Requirements Analysis: Requirements sources, Elicitation techniques. Requirements Analysis: Requirements classification, Conceptual modeling, Architectural design and requirements allocation, Requirements negotiation, Formal analysis. Requirements Specification: System definition document, System requirements document, Software requirements specification. Requirements Validation*:* Requirements reviews, Prototyping, Model validation, Acceptance tests. Practical Considerations: Iterative nature of the requirements process, Change management, Requirements attributes, Requirements tracing, Measuring requirements. Software Requirements Tools. Current research topics in requirement engineering.

Teaching Methodology:

Group project, Industry Visit/Case study, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Papers

Reference Materials:

1. *Software Engineering: A Practitioner's Approach*, Roger S. Pressman, Bruce R. Maxim, 8th Ed, McGraw-Hill Education, 2015.
2. *Object-Oriented Analysis, Design and Implementation*, Brahma Dathan, Sarnath Ramnath, 2nd Ed, Universities Press, India, 2014.
3. *Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures*, Hassan Gomaa, Cambridge University Press, 2011.
4. *Applying UML & Patterns: An Introduction to Object-Oriented Analysis & Design and Iterative Development*, Craig Larmen, 3rd Edition.
5. *Head First Design Patterns*, Eric Freeman, Elisabeth Freeman, Kathy Sierra and Bert Bates, O'Reilly Media, Inc., 2004.

Advanced Software Project Management

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Introduction to project management. Algorithmic cost estimation models. Advanced cost estimation models. Function points estimation Risk assessment. Life cycle models. Prototyping. Management of software reuse. Software maintenance. Software maturity framework. An Overview of Project Planning. Program Management and Project Evaluation. Software Effort Estimation. Activity Planning. Risk Analysis and Management. Resource Allocation. Project tracking and Control. Contract Management. Software Quality Assurance. Configuration Management. Various tools of Software Project Management. Project Cost Management. Project Human Resource Management. Project Communications Management. Project Procurement Management. Case studies, Current research topics in Software Project Management.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Paper

Reference Materials:

1. *Software Project Management*, Bob Hughes & Mike Cotterell, 3rd Ed., McGraw-Hill Publication, 2003, ISBN: 0707709834X
2. *Software Project Management in Practice*, Pankaj Jalote, Addison-Wesley, 2002, ISBN 0-201-73721-3

Advanced Software System Architecture

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Quality attributes in the context of architecting. Qualitative and quantitative assessment of architectures. Architectural modeling through Architecture Description Languages. System modeling its relation to software architecting. Architecting for evolution and variability. Partitioned and layered architectures. System-of-Systems and Ultra-Large Scale Systems. Software Product Lines and Configurable Software. Self-Adaptive Software. Architectural Description Languages. Feature Modeling. Architecture and Model-Based Testing. Current research topics in software system architecture.

Teaching Methodology:

Case Study, Project, Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Papers

Reference Materials:

1. ***Designing Software Architectures: A Practical Approach (SEI Series in Software Engineering***), Humberto Cervantes, Rick Kazman, 1st Edition, Addison-Wesley Professional, 2016.
2. *Software Product Lines: Practices and Patterns,* P. Clements and L. Northrup, Addison-Wesley, 2002.
3. *Software Architecture : Foundations, Theory, and Practice,* R. Taylor, N. Medvidović and E.M. Dashofy, John Wiley, 2010

Advanced Topics in Applied Cryptography

Credit Hours:	3	**Prerequisites:**	Information Security

Course Content:

The course covers (but is not limited to) the following topics:

Privacy-Enhancing Technologies: Privacy-Preserving Data Collection and Data Publishing, Privacy-Preserving Data Mining, K-Anonymity, Anonymous communications, Anonymous credentials, Group signatures, Privacy and anonymity in peer-to-peer architectures, Privacy-enhanced access control or authentication or certification;

Advanced Crypto Algorithms and Protocols: Zero-knowledge proof, Oblivious Transfer, Secure Multiparty Computation, Digital Cash, Secret Sharing, Threshold Cryptography, Identity-Based Encryption, Attribute-Based Encryption

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Term Paper, Presentations, Final Exam

Reference Materials:

Current research papers on the selected topic.

Agent Based Modeling

Credit Hours:	**3**	**Prerequisites:**	None

Course Content:

Introduction to agent based modeling. Introduction to Net Logo. Complexity in Social Worlds. Net Logo Commands. Net Logo Procedures. Model properties (Why agent-based objects? Agents, environments, and timescales).Biological systems: fireflies, flocking, slime mold, bees, ants (flocking behavior slime mold).Biological systems: predator/prey, debugging (Verification and validation).Social systems: segregation, Schelling, Micro motives and Macro behavior. A self-forming neighborhood model. Cellular automata. Critical phenomena. Sand piles. Current research topics in Agent Based Modeling.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Paper

Reference Materials:

1. *Agent-Based Models,* Nigel Gilbert, SAGE Publications, 2008

Agile Software Development

Credit Hours:	**3**	**Prerequisites:**	None

Course Content:

Agile values and principles. Agile Practices. Pair programming Refactoring. Test-driven development. Continuous integration and delivery. Automated build. Coding standards simplicity. SMART user stories and release and deployment. Applying Agile methods: Integration, XP+SCRUM, SCRUM +Kanban, Agile methods +User-Centered Design. Distributed Agile teams. Current research topics in **Agile Software Development.**

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Paper

Reference Materials:

1. *Agile Software Development, Principles, Patterns, and Practices*, Robert C. Martin, Pearson, 2002.
2. *Extreme Programming Explained*, Kent Back & Cynthia Andres, 2nd Edition, Addison-Wesley Professional 2005.
3. *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*, Andrew Stallman and Jennifer Greene, O'Reilly Media, 2014.

Applied Cryptography

Credit Hours:	3	**Prerequisites:**	Information Security

Course Content:

Smart Cards, Hardware Security. Security engineering. Mobile phone security. RFID systems, access control, user/data authentication. Key sizes. Random number and key generation. Symmetric cryptography engineering, key derivation and key management. Bank cards and terminals, history, EMV specs, different forms of security, fraud, attacks. Public key crypto engineering, best practices, standardized algorithms and padding methods. PGP vs. smart cards. PKI vs. alternatives. Applications of digital signatures. Legal/regulatory aspects, qualified certificates, times tamping. More applications of PK crypto. Electronic passports and ID cards vs. SDA/DDA/CDA in bank cards. Electronic commerce, SSL/TLS, Forward Security, standard methods of encoding of digital signatures and certificates (X.509). Financial cryptography, payment systems, crypto currencies, bit-coin. Side-channel attacks (timing, SPA, DPA and DFA). Side-channel attack countermeasures.

Tutorial and Labs: Writing programs with standard crypto libraries (Open SSL, NTL, GMP) and developing efficient and secure implementations of cryptography in C++/Java.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Term Paper, Presentations, Final Exam

Reference Materials:

Current research papers on the selected topic.

Big Data Analytics

Credit Hours:	3	**Prerequisites:**	None

Course Contents:

Introduction Hadoop and Map Reduce, Association Rules: Frequent item sets and association rule mining, Similar item sets and LSH, Near Neighbor Search in High Dimensional Data, Recommender systems, Link analysis: Personalized PageRank, Hubs and Authorities, Web spam and Trust Rank, Clustering, Descriptive analytics -- clustering, Dimensionality reduction: SVD a, Machine learning with massive datasets, Mining streaming data, Analysis of very large graphs, Time series data and streaming, Other application areas, Proximity search on Graphs: Random Walks with Restarts, Web Advertising,

Teaching Methodology:

Lectures, Problem based learning

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Material:

Books:

1. Mining of Massive Datasets, 1st Edition, Anand Rajaraman and Jeffrey Ullman

Complex Networks

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Introduction to complex networks. What is a complex system? Basic metrics. Degree distribution (DD).Clustering coefficient (CC). Centrality. Page Rank. Hubs and authorities. Bib-coupling. Co-citation index. Edge reciprocity. Rich club phenomenon. Social Network. Homophily. Cohesiveness. Equivalence of ties. Ego-centric networks. Community Structures. Hierarchical Agglomerative. Linear algebra techniques and spectral methods. Citation Networks, Rise and fall of CS fields. Inter-disciplinarily of CS fields. Temporal structures of citation profiles. Citation count prediction. Co-authorship circles. Economic and financial network analytics. Graph mining. Measuring user engagement. Basic definitions and metrics: walks, paths, cycles, connectedness, trees. The clustering coefficient. The World Wide Web. Scale-free networks. Random graphs with a given degree sequence. The Barabasi-Albert model and other models of growing graphs. Degree correlations. The Internet and other assortative and dissertated networks. Community structures: spectral bisection and hierarchical clustering methods. The modularity and Girvan-Newman algorithm. Current research topics in Complex Networks.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Paper

Reference Materials:

1. *Complex networks*, Ronaldo Menezes, Alexandre Evsukoff, Marta C. González, Springer-Verlag Berlin Heidelberg, 2013.

Component Based Software Engineering

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Introduction to Software Component (Component. Definition and Essentials, What is CBSE? Why CBSE? The Anatomy of Components: internals, application interfaces, platform interfaces, middleware, Component Characteristics: Properties of Software Component in CBSE). Basic Concepts in CBSE (Improving SW through Software Process Improvement (SPI)).Component-Based Software Development (CBSD).Approach. Component Patterns & Abstraction. Challenges of CBSE. Technical Issues and Objectives of Component Based Software Engineering. Reuse Dimensions. Software Components Types: open, closed, COTS, in house. Challenges in Software

Reuse. Software Component Specification. Specification Techniques. Specifying the Semantics of Components. Specifying Extra-Functional Properties. Architecting component based systems (Software Architecture Parts, The Roles of Software Architecture, Designing Software Architectures, Architectural Styles, Architecture-Driven Component Development, Components and Component Models, Component Model Implementation, Component Frameworks, Black-Box and White-Box Frameworks, How do we use Framework in CBSE?, Component Interface Specification). Component Engineering Process: Domain Engineering, Domain Engineering pattern based design. Domain Engineering: Component Repositories, Overview of Existing Component Techniques, Component testing in CBSE. Current research topics in Component Based Software Engineering.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Paper

Reference Materials:

1. *Software Engineering: A Practitioner's Approach*, Roger S. Pressman, 8th Edition, McGraw-Hill Higher Education, 2015
2. *Building Reliable Component Based Software Systems*, Ivica Crnkovic and Magnus Larsson, Artech House Publishers; 1st edition, 2002
3. *Component-Based Development: Principles and Planning for Business Systems*, **Katharine Whitehead**, Addison Wilsey, 2010

Cryptography

Credit Hours:	3	**Prerequisites:**	Information Security, Mathematics

Course Content:

Elementary number theory: Prime numbers, Factoring, Modular arithmetic, Fermat's & Euler's theorems, gcd, Euclid's algorithm, Discrete logarithm problem
Public key encryption: Public key crypto systems, RSA algorithm, Elliptic Curve cryptography
Hash digests: Properties of cryptographic hash functions, Merkle Damgard construction, md family, sha family, Digital signatures, sha3
Block ciphers: Block cipher principles, Feistel networks, S boxes and P boxes, Block cipher modes of operation, DES, 3DES, AES
Interactive Proofs, Zero-Knowledge Proofs, Zero-Knowledge Proofs of Knowledge, Non-Interactive Zero-Knowledge Proofs, Secure Protocols, Two-Party Secure Computation, Multiparty Secure Computation, Chosen Cipher text Security

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Term Paper, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Term Paper, Presentations, Final Exam

Reference Materials:

1. The course materials will consist of research papers related to each topic.

Database Security

Credit Hours:	3	**Prerequisites:**	Database Management Systems

Course Content:

Transaction Processing, Serialisability Theory, Two Phase Locking, Centralised Recovery, Distributed Recovery, Security and Security Models, Relational Database Security, Statistical Database Security, Concurrency Control and Multi-Level Security, Oracle Security

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Concurrency Control and Recovery in Database Systems, P.A. Bernstein, V. Hadzilacos and N. Goodman
2. Database Security, S. Castano, M. Fugini, G. Martella, P. Samarati
3. Computer Security, D. Gollmann

Deep Learning

Credit Hours:	3	**Prerequisites:**	Machine Learning

Course Content:

Introduction to Deep learning, Review of Linear classification (Multi-class Support Vector Machines, Soft max) and Regularization, Gradient Descent & Stochastic Gradient Descent (SGD), Back propagation (Intuitions, back propogation as flow graph), Introduction to Neural Networks (model of a biological neuron, activation functions, neural net architecture, representational power, etc.), Building Neural Networks (data preprocessing, loss functions, weight initialization, regularization, dropout, batch normalization), Learning Neural Networks (Learning and Evaluation gradient checks, sanity checks), Variants of SGD (momentum, Adagrad/RMSprop, ADAM), Introduction to Convolutional Neural Networks (CNN) and its components (Convolution and Pooling Layers), Convolutional Neural Network case studies (AlexNet/ZFNet/VGGNet), Understanding and Visualizing Convolutional Neural Networks, Convolutional networks for other visual Recognition Tasks (Localization, Detection, Segmentation, etc.), Transfer Learning and Fine-tuning Convolutional Neural Networks, Introduction to Natural Language Processing (NLP), Learning word and sentences embedding (wordvec, glove, sentvec), Introduction to recurrent networks (RNNs, LSTMS, etc.), Applications of Recurrent neural networks to different NLP tasks (e.g. sentiment analysis, parsing, NER tagging, etc.), Introduction to Reinforcement Learning and Q-Learning, Deep Q-Networks (DQN) and Game playing using DQN, Introduction to Policy gradients and their applications.,

Teaching Methodology:

Lectures, Problem based learning

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Material:

Books

1. Deep Learning, 1st Edition, Yoshua Bengio, Ian Goodfellow, Aaron Courville, Neural networks and deep learning, 1st Edition, Michael A. Nielsen
2. Hands-On Machine Learning with Scikit-Learn and Tensor Flow, 1st Edition, Aurélien Géron

Distributed Data Processing

Credit Hours:	3	**Prerequisites:**	None

Course Contents:

Introduction to distributed data Processing, Introduction to Spark, The Spark Programming Model, RDD Fundamentals, Programming With RDDs and Key-Value Pairs, File Formats, Spark SQL and data frames, Spark Job Execution, Intro to Spark Streaming, Building systems using Spark Streaming, Extract-Transform-Load operations (ETL) and Exploratory Data Analysis (EDA) using Spark, Machine Learning With MLLib, Machine learning models building, Hyper-parameter search, Cross validation and evaluation using MLLib, Distributed deep learning using Spark.,

Teaching Methodology:

Lectures, Problem based learning

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam.

Reference Material:

Books

1. Learning Spark for Lightning-Fast Big Data Analysis, 1st Edition, Matei Zaharia, Holden Karau, Andy Konwinski, Patrick Wendell
2. Advanced Analytics with Spark: Patterns for Learning from Data at Scale, Sandy Ryza, Uri Laserson, Josh Wills, Sean Owen
3. Machine Learning with Spark – Tackle Big Data with Powerful Spark Machine Learning Algorithms, Nick Pentreath

Empirical Software Engineering

Credit Hours:	**3**	**Prerequisites:**	None

Course Content:

Quantitative study design. Qualitative study designs. Measurement and data collection. State-of-the practice. Archival data analysis. Human variation & impact of experience. Evidence-based software engineering. Simulation of software process. Current research techniques in Empirical Software Engineering.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Paper

Reference Materials:

1. *Experimentation in Software Engineering* by C. Wohlin , Kluwer, 2000. ISBN 0-7923-8682-5.
2. *Research Methods Knowledge Base*, by William M.K., 2002

Information Privacy and Security

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Overview of e-security: Threats, risks, consequences, Sources of threats, Attacks classification, Preventive measures, remedial measures
Cryptography for e-security: Historical perspective, Confusion vs. diffusion, Stream ciphers vs. block ciphers, Keys and key management, Key exchange (peer to peer, peer – key server – peer), Diffie Helman key sharing scheme, Symmetric key cryptography vs asymmetric key cryptography, Trapdoor functions
GPG: Overview of GPG, Commands and CLI, GPG trust model, GUI – KGPG, Seahorse, Frontends – Kleopatra, enigmail 2
Practical applications: PKI, CA. X509 certificates, SSL/TLS, HTTPS, IPV6 and IPSEC, Proxies and Firewalls
Misc. techniques: Encryption using non-cryptographic tools (vi, zip), Authentication principles and methods, Passwords, two-factor authentication, One-way encryption, Steganography, Hamming, Chaffing and Winnowing,
Management aspects: System Administration policies, Security audit, Penetration testing and ethical hacking, Mandatory Access control, Discretionary Access Control, Monitoring and logging tools, Legal aspects

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. William Stallings, Cryptography and network security, Pearson Education.
2. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone , Handbook of Applied Cryptography, CRC Press.
3. Margaret Cozzens, Steven J Miller, The mathematics of encryption, American Mathematical Society

Machine Learning

Credit Hours:	3	**Prerequisites:**	None

Course Learning Outcomes (CLOs):		
The core objectives of this course are	**Domain**	**BT Level***

* BT= Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Contents:

Introduction to machine learning and statistical pattern recognition. Supervised learning: Part I (Graphical models (full Bayes, Naïve Bayes), Decision trees for classification & regression for both categorical & numerical data, Ensemble methods, Random forests, Boosting (Adaboost and Xgboost), Stacking; Part II (Four Components of Machine Learning Algorithm (Hypothesis, Loss Functions, Derivatives and Optimization Algorithms), Gradient Descent, Stochastic Gradient Descent, Linear Regression, Nonlinear Regression, Perceptron, Support vector machines, Kernel Methods, Logistic Regression, Softmax, Neural networks); Unsupervised learning: K-means, Density Based Clustering Methods (DBSCAN, etc.), Gaussian mixture models, EM algorithm, etc.; Reinforcement learning; Tuning model complexity; Bias-Variance Tradeoff; Grid Search, Random Search; Evaluation Metrics; Reporting predictive performance

Teaching Methodology:

Lectures, Problem based learning

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Material:

Books

1. Elements of Statistical Learning
2. Pattern Recognition & Machine Learning, 1st Edition, Chris Bishop
3. Machine Learning: A Probabilistic Perspective, 1st Edition, Kevin R Murphy
4. Applied Machine Learning, online Edition, David Forsyth, http://luthuli.cs.uiuc.edu/~daf/courses/LearningCourse17/learning-book-6-April-nn-revision.pdf

Management & Organizational Behavior

Credit Hours:	3	**Prerequisites:**	None

Course Learning Outcomes (CLOs):		
The core objectives of this course are	**Domain**	**BT Level***

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Contents:

Organizational Behavior (OB) is an interdisciplinary field geared to satisfy managers' quest to know why people behave as they do in relation to their jobs, their work groups and their organizations. Drawing on numerous disciplines including psychology, sociology, anthropology and economics, OB identifies and explores factors that influence individual and group behavior in organizations. This knowledge of individuals' perceptions, motivational attitudes and behavior enables managers to not only understand themselves better, but also to adopt appropriate managerial policies and leadership styles to increase their effectiveness. Students will be able to demonstrate analytical and problem solving skills in the application of this knowledge to work-related situations.

Teaching Methodology:

Lectures, Problem based learning

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Material:

Books

1.

Natural Language Processing

Credit Hours:	3	**Prerequisites:**	

Course Content:

Introduction to NLP, linguistics and NLP tasks, Python libraries for NLP, Text preprocessing and N-grams, Softmax / MAXENT (sequence) classifiers, sequence classifiers for POS & NER, Deep learning based word representations & deep networks for NER, Recurrent networks and language modeling, Statistical machine translation, Machine translation: word alignment, parallel corpora, decoding, evaluation, Modern deep learning machine translation systems (phrase-based, syntactic), Syntax and parsing, co-reference resolution, Tree recursive neural networks for POS tagging, Computational semantics, Question answering, Text summarization, Dialogue systems

Teaching Methodology:

Lectures, Problem based learning

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Material:

Books

1. Speech and Language Processing, 2nd Edition, Daniel Jurafsky and James Martin.
2. Foundations of Statistical Natural Language Processing, 2nd Edition, Chris Manning and Hinrich Schuetze,
3. Neural Network Methods for Natural Language Processing, 1st Edition, Yoav Goldberg

Quantum Computing and Information Security

Credit Hours:	3	**Prerequisites:**	Information Security

Course Content:

The course covers (but is not limited to) the following topics:
Introduction to quantum mechanics: Hilbert space, Unitary and stochastic dynamics, Probabilities and measurements, Entanglement, Density operators and correlations; **Introduction to quantum information:** Classical information theory, Quantum information types and quantum channels, Dense coding, Teleportation, No cloning, Quantum cryptography; **Quantum algorithms:** Classical computation, Shor factorization, Grover search, Measurement-based computation; **Physical realizations:** Optical lattices; **Noise and error correction:** Quantum operations, Graph states and codes, Quantum error correction, Fault-tolerant computation

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Term Paper, Presentations, Final Exam

Reference Materials:

Current research papers on the selected topic.

Quantum Cryptography

Credit Hours:	3	**Prerequisites:**	Information Security

Course Content:

From essential tools to the first quantum protocol, The power of entanglement, Quantifying information, From imperfect information to (near) perfect security, Distributing keys, Quantum key distribution protocols, Quantum cryptography using untrusted devices, Quantum cryptography beyond key-distribution, Perfect security from physical assumptions, Further topics

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Term Paper, Presentations, Final Exam

Reference Materials:

Current research papers on the selected topic.

Reliability Engineering

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Introduction to Reliability Engineering. The Need for Reliable Software. Software Reliability Engineering Concepts. Basic Definitions. Software Reliability and System Reliability. The Dependability Concept. Reliability Modeling. Availability Modeling. Statistical Reliability Models for Software Reliability. Best Current Practices of software Reliability Engineering. Software Metrics for Reliability Assessment. Software Testing and Reliability. Software Reliability Tools. Review of Reliability Theory. Analytical Techniques and Basic Statistics for Reliability Engineering. Current research topics in Reliability Engineering.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Papers

Reference Materials:

1. *An Introduction to Reliability and Maintainability Engineering*, Ebeling, C. E., Waveland Press, Inc., 2nd edition. 2009 (ISBN 1-57766-625-9)

2. *IEEE Recommended Practice in Software Reliability Handbook of Software Reliability Engineering* by Michael R. Lyu. Published by IEEE Computer Society Press and McGraw-Hill Book Company, 2008

Requirements Engineering

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Definition of requirements engineering and role in system development, Fundamental concepts and activities of requirements engineering, Information elicitation techniques, Modeling scenarios Fundamentals of goal oriented requirements engineering, Modeling behavioral goals, Modeling quality goals, Goal modeling heuristics, Object modeling for requirements engineering, Object modeling notations, Object modeling heuristics, Identifying objects from goals, Modeling use cases and state machines, Deriving operational requirements from goals, Requirements Specification, Requirements negotiation, Requirements verification and validation Management of inconsistency and conflict, requirements engineering risks, the role of quality goals in the requirements selection process, Techniques for requirements evaluation, selection and prioritization; Requirements management; Requirements traceability and impact analysis.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Papers

Reference Materials:

Research Methodology

Credit Hours:	**3**	**Prerequisites:**	None

Course Content:

Introduction to Research. Objectives of Research. Importance of Research Methodology in Research Study. Types of Research. Steps in Conducting Research. What is Literature Review? Why need for Literature Review. Types of Literature Review. Systematic Literature Review Protocol. Problem Statement and Problem formulation. Criteria for selecting a problem. Identifying Types of variables in Research. Types of hypothesis. Identifying Target Population. Types of Sampling. Sampling Techniques. Quantitative Research Methods. Scientific Methods. Design of Quantitative Surveys. Techniques to Conduct Quantitative Methods. Introduction to Qualitative Research. Qualitative Research Methods. Data Analysis and Theory in Qualitative Research Articles. Introduction to Mixed Methods Research. Design of Mixed Methods Research. Evaluation of Mixed Methods Research. Case Study. How to Conduct a Case Study. Case Study Protocol. Importance and Benefits of Case Study. Types of Statistical Tests to Conduct Data Analysis. Data Analysis Tools. Introduction to SPSS. Hands on Practice of SPSS. How to Define variables in SPSS. How to Record Collected Data in SPSS. Types of Tests via SPSS including Regression. Correlation. Cross tabulation and others. How to write Good Research Proposal. Contents of Thesis. Important Elements of Research Thesis.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Paper

Reference Materials:

1. Research design: Qualitative, quantitative and mixed methods approaches, Creswell, J. W. Thousand Oaks, CA: Sage,4th Ed. 2014.

Research Methods

Credit Hours:	3	**Prerequisites:**	Probability and Statistics

Course Content:

Research: introduction to the nature of research, and types of Research;
Research questions, and the nature of evidence: deciding what type of question to ask, and how to handle the various types of answer; Mud pits and how to avoid them: things that go wrong; Isms: necessary assumptions, dubious assumptions, and being caught in crossfire; Searching the literature: why, where, what for and how; Research in society agendas, context and the like: things we take for granted, and things that can cause you trouble; Research design: Types of design: which to use and how to use them; Surveys and sampling; Field experiments: doing research in the world. Controlled experiments: changing things systematically and seeing what happens; Summary and technical terms; Generic advice; arranging a study: subjects, equipment, procedures, things to remember, things to beware; Handling subjects; Recording; Data collection; Data collection methods: the methods, and choosing and using the appropriate method; Reports: getting respondents to talk about how things happen; Observation: watching what happens; Card sorts: getting respondents to categorize things; Laddering: unpacking the respondents’ concepts systematically; Repertory grids: a systematic representation for respondents’ knowledge interviews: asking people questions; Face-to -face interactions with respondents: the nuts and bolts of asking questions; Questionnaires: when to use, when not to use, which questions to ask, what format to use; Data analysis; Content analysis: what is said in a text, how it is said, and how often it’s said; Discourse analysis: who says what, about what, to whom, in what format. Knowledge representation: formats, structures and concepts for making sense of knowledge; Statistics: describing things with numbers, and assessing the odds; Descriptive statistics: giving a systematic description of the numbers you’ve found; Measurement theory: types of measurement and their implications; Inferential statistics: what are the odds against your findings being due to random chance? Conclusion: the end game; Writing up: demonstrating your excellence efficiently, and practical points to remember; References and referencing: using and citing the right texts to demonstrate your excellence; what next; thinking forward about what you really want your life to be?

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. *A Gentle Guide to Research*, Gordon Rugg & Marian Petre, Open University Press McGraw-Hill Education, 2007

2. *Practical Research Methods*, CATHERINE DAWSON, How To Books Ltd,3 Newtec Place, 2002.

Securing the Internet of Things

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Introduction of Internet of Things (IoT), need of IoT Security, Requirement and Basic Properties, Main Challenges, Confidentiality, Integrity, Availability, Non-Repudiation, IoT Architectures (Device, Cloud, Gateway, Backend, Applications), Security Classification & Access Control, Data classification (Public and Private), Privacy issues in IoT, IoT Authentication and Authorization, IoT Data Integrity, Web Based Attacks and Implementation in IoT, Denial of Service, Sniffing, Phishing, DNS Hijacking, Pharming, Defacement etc., Cryptology Cipher –Symmetric Key Algorithms (AES and DES), Asymmetric Key Algorithm (RSA) Attacks– Dictionary and Brute Force, Lookup Tables, Reverse Lookup Tables, Rainbow Tables, Attack Surface in IoT and Threat Assessment, Embedded Device – UART, SPI, I2C, JTAG, Software and Cloud components– Firmware of the device, Web Application Dashboard, Mobile Application used to Control, Configure and Monitor the devices, Radio Communication– WIFI, BLE, Cellular, ZigBee*, ZWave*, 6LoWPAN, IoT Protocol inbuilt Security Features On Transport Layer and Application Layer, Security Management, Identity and Access Management, Key Management, Model Discussions (Smart Home, Smart Agriculture, Smart Retail Supply, Smart Healthcare, Smart Grid, Smart Cities).

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

Recommended Books:

1. Fei Hu, Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations, 1st Edition, ISBN-13:978-1498723183.
2. Brian Russell, Drew Van Duren, Practical Internet of Things Security, 2016.

Security Management

Credit Hours:	3	**Prerequisites:**	None

Course Content:

- Fundamentals and need of information security management
- The role of standards in information security management
- Internal control, audit and security

- The role of risk in information security management
- Information security, governance and law
- Case studies in information security management

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Information Security Management Principles, Andy Taylor, David Alexander, Amanda Finch and David Sutton, 2nd Ed.
2. A Practical Guide to Managing Information Security, Steve Purser

Security Testing

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Security testing frameworks and methodologies.
Legal aspects of performing penetration testing.
Network security and its vulnerabilities, including how these vulnerabilities may be exploited.
Computer security covering operating systems and access control vulnerabilities, and how to exploit and mitigate these vulnerabilities.
Internet based applications, web services, protocols, languages (e.g. SQL) and how these may be exploited using for example SQL injection and cross-site scripting; how to exploit these vulnerabilities, and how to mitigate these vulnerabilities.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Professional Penetration Testing, Thomas Wilhelm, 2nd Ed.
2. Kali Linux: Assuring Security by Penetration Testing, Lee Allen, Tedi Heriyanto, and Shakeel Ali.
3. Gray Hat Hacking, Branko Spasojevic, 3rd Ed.
4. The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws, Dafydd Stuttard and Marcus Pinto, 2nd Ed.

Software Configuration Management

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Management of the SCM Process. Organizational Context for SCM. Constraints and Guidance for the SCM Process. Planning for SCM. SCM Plan. Surveillance of Software Configuration Management. Software Configuration Identification. Identifying Items to Be Controlled. Software Library. Software Configuration Control. Requesting, Evaluating, and Approving Software Changes. Implementing Software Changes. Deviations and Waivers. Software Configuration Status Accounting. Software Configuration Status Information. Software Configuration Status Reporting. Software Configuration Auditing. Software Functional Configuration Audit. Software Physical Configuration Audit. In-process Audits of a Software Baseline. Software Release Management and Delivery. Software Building. Software Release Management. Software Configuration Management Tools. Current research topics in Software Configuration Management.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Paper

Reference Materials:

1. *Software Configuration Management Patterns: Effective Teamwork, Practical Integration* by Stephen P. Berczuk, Brad Appleton, 2003

Software Measurement and Metrics

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Introduction to quality control and planning needs (Measurement Concepts, Measurement as a support process, Review Metrics Models and Standards). Measurement goals (Formulating problem and goal statement, Prioritize information needs and objectives, Formalize measurement goals). Specify Measures (Identify questions and indicators, Identify data elements, Operational definitions for measures). Specify Data Collection and Storage Procedures. Sources of data. How to collect and store the measurement data? Specify Analysis Procedures. Potential data analyses. Methods and tools for measuring software. Develop software measurement reporting. Current research topics in Software Measurement and Metrics.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Papers

Reference Materials:

1. *Metrics and Models in Software Quality Engineering*, Stephen H. Kan, Addison Wesley, 2003
2. *Measuring the Software Process*, Anita Carleton, William A. Florac, Addison-Wesley 1999
3. *The Big Book of Six Sigma training Games*, Chris Chen and Hadley Roth, McGraw-Hill, 2005

Software Process Management & Metrics

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Introduction to software processes and their significance. Process Models: Object-Oriented Software Process Model, Unified Process, SOMA; Implications of Software development methodology on software processes. Process planning: resource allocation, SDLC and deliverable definition, role and responsibility definition, measurement planning: process metrics and process training; Process implementation: training, process prototyping, social issues (e.g., resistance, buy-in), other issues (e.g. time and risk management for software projects using processes); Process monitoring: process measurement; Process improvement. Process Standardization: TQM, ISO, CMM and others. Advanced issues: Process change management, Process Patterns, organizational and personal software processes.

The course begins with the importance of **software metrics**; metrics parameters are highlighted; role of software metrics in SDLC is discussed; particular emphasis is placed in Process Metrics and the corresponding issues discussed are: key responsibilities of process management, perspective of process measurement (performance, stability, compliance, capability, improvement), planning measures for process management, applying measures to process management (data collection, analyzing data, acting on the results); software cost estimation techniques (manual, automated) are discussed like SLOC, COCOMO, FP; object oriented design metrics are explored; software quality metrics issues are taken up; finally studies for software assessment & bench marks are taken up along with software best & worst practices.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Papers

Reference Materials:

Software Project Management

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Software Project planning (including scope and time management), execution, and monitoring. Project risk management, resource estimation and assignment. Cost estimation.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Papers

Reference Materials:

Software Quality Assurance

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Basic software quality assurance and testing concepts, SQA management & planning, software inspections and walkthroughs, software reliability engineering, white-box testing, black-box testing, testing object-oriented system, advanced testing topics.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Papers

Reference Materials:

Software Risk Management

Credit Hours:	3	**Prerequisites:**	None

Course Content:

What is risk and risk management?. Motivation for risk management. Reasons we don’t do risk management. SEI’s Risk Management paradigm. Identifying and recording software risk. Risk Taxonomy. Tools and methods for identifying and recording risks. Analyzing and classifying risks. Complex project management theory. Software Risk Identification. Software Risk Analysis. Software Risk Planning. Software Risk Monitoring. Software Qualitative Risk Analysis. Quantitative Risk Analysis. Risk management and the SDLC. Risk management in CMM. Other useful tools for successful risk management. Current research topics in Software Risk Management.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment:

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Paper

Reference Materials:

1. *Software Engineering Risk Management* by Dale Walter Karolak, 1995, ISBN9780818671944
2. *Applied Software Risk Management: A Guide for Software Project Managers* by C. Ravindranath Pandian, 2006, ISBN 9780849305245
3. *Software Risk Management* by Boehm, Barry, W. IEEE Computer Society Press, ISBN 10: 0818689064

Software Testing and Quality Assurance

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Testing techniques. Black Box testing, White Box and Grey Box testing techniques. Quality Assurance planning and execution. Automated testing topics include constructing a framework, scripting techniques, generating a test data, generating test architecture, pre/post-processing, test maintenance, and job specific metrics. Current research topics in Software Testing and Quality Assurance.

Teaching Methodology:

Lectures, Problem based learning, Research Papers

Course Assessment

Sessional Exam, Assignments, Quizzes, Project, Presentations, Final Exam, Term Papers

Reference Materials:

1. ***Software Quality Assurance: Integrating Testing, Security, and Audit*** (Internal Audit and IT Audit), Abu Sayed Mahfuz, Auerbach Publications, 2016.
2. *Practical Model-Based Testing: A Tools Approach*, Mark Utting and Bruno Legeard, Morgan Kaufmann Publishers Inc., San Francisco, CA, 2006.
3. *Software Quality Engineering, Testing, Quality Assurance, and Quantifiable improvements*, Jeff Tian, IEEE Computer Society, 2005.
4. *Introduction to Software Engineering*, P Ammann and J Offutt, Cambridge University Press, 2008.

Statistical and Mathematical Methods for Data Science

Credit Hours:	3	**Prerequisites:**	None

Course Contents:

Probability: Probability basics (axioms of probability, conditional probability, random variables, expectation, independence, etc.), multivariate distributions, Maximum a posteriori and maximum likelihood estimation; **Statistics:** introduction to concentration bounds, laws of large numbers, central limit theorem, minimum mean-squared error estimation, confidence intervals; **Linear algebra:** Vector spaces, Projections (will also cover the least regression), linear transformations, singular value decomposition (this substitute for PCA), eigen decomposition, power method; **Optimization:** Matrix calculus with Lagrange Multipliers, gradient descent, coordinate descent, introduction to convex optimization.

Teaching Methodology:

Lectures, Problem based learning

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials

Books:

1. Probability and Statistics for Computer Scientists, 2nd Edition, Michael Baron
2. Linear Algebra and Its Applications, 5th Edition, David C. Lay and Steven R. Lay
3. Introduction to Linear Algebra, 5th Edition, Gilbert Strang
4. Probability for Computer Scientists, online Edition, David Forsyth.

Tools and Techniques in Data science:

Credit Hours:	3	**Prerequisites:**	None

Course Contents:

Introduction to Data Science, Data Science Life cycle & Process (Asking Right Questions, Obtaining Data, Understanding Data, Building Predictive Models, Generating Visualizations) For Building Data Products, Introduction to Data (Types of Data and Datasets), Data Quality (Measurement and Data Collection Issues), Data pre-processing Stages (Aggregation, Sampling, Dimensionality Reduction, Feature subset selection, Feature creation etc.), Algebraic & Probabilistic View of Data, Introduction to Python Data Science Stack (Python, Numpy, Pandas, Matplotlib), Relational Algebra & SQL, Scraping & Data Wrangling (assessing, structuring, cleaning & munging of data), Basic Descriptive & Exploratory Data Analysis, Introduction to Text Analysis (Stemming, Lemmatization, Bag of Words, TF-IDF), Introduction to Prediction and Inference (Supervised & Unsupervised) Algorithms, Introduction to Scikit Learn, Bias-Variance Tradeoff, Model Evaluation & Performance Metrics (Accuracy, Contingency Matrix, Precision-Recall, F-1 Score, Lift, etc.), Introduction to Map-Reduce paradigm

Teaching Methodology:

Lectures, Problem based learning

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

Books:

1. Python for Data Analysis, 1st Edition, William McKinney
2. An Introduction to Statistical Learning with Applications in R, 1st Edition, G. James, D. Witten, T. Hastie and R. Tibshirani
3. Computational and Inferential Thinking: The Foundations of Data Science, 1st Edition, A. Adhikari and J. DeNero
4. Data Mining and Analysis: Fundamental Concepts and Algorithms, 1st Edition, M. Zaki & W. Meira,
5. Data Science from Scratch, 1st Edition, Joel Grus
6. Doing Data Science, 1st Edition, Cathy O'Neil and Rachel Schutt
7. Introduction to Data Science. A Python Approach to Concepts, Techniques and Applications, 1st Edition, Laura Igual.

Trusted Computing

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Introduction, Security issues in Industry, Elements of Trusted Computing, Trusted Platform Module (TPM), Trusted Computing Applications, Digital Rights Management, Trusted Network Connect (TNC), Trusted Servers, Storage, Trusted Input and Output Devices, Mobile Phones, Authentication, Remote Attestation, Network attestation and platform measurement, Application and Content Protection, TPM Keys management schemes, TPM Programming

1. Thunderbird integration: TPM protection of key store,
2. tboot: GRUB (boot loader) version with extra TPM compatibility, features,
3. Trusted Software Stack (Highlevel API for TPM and TrouSerS on Linux)
4. Driver Level Coding
5. Drive Encryption (BitLocker Technology)

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1- David Challener, Kent Yoder, Ryan Catherman, David Safford , Leendert Van Doorn, A Practical Guide to Trusted Computing, 1st edition, ISBN-13:978-0132398428

2- Chris Mitchell (editor).Trusted Computing, IEE, Hertfordshire, UK, 2005. ISBN 0-86341-535-3.

3- Mihir Bellare and Phillip Rogaway. Introduction to Modern Cryptography, (2005)

Wireless Security

Credit Hours:	3	**Prerequisites:**	None

Course Content:

Vulnerabilities of Wired and Wireless Networks, Attacks in wireless networks: Passive and Active Attacks, DOS and DDoS attacks, TCP attack, Trojan Attacks, Xhole attacks etc. Securing neighborhood discovery. Securing route in multi-hop networks. 802.11 Security and authentication mechanism. Security in Ad-Hoc networks, Reactive, hybrid and Proactive routing security. Data modification and tunnel attacks, intrusion detection and intrusion tolerance in various networks. WSN security for real time applications. Key agreements in 5G networks. Security measures in L2 and L1 of 802x protocols. Trust assumptions in cooperative networks. Trust management in relay networks. Selfish behavior at MAC layers of CSMA/CA, Selfishness in packet forwarding.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Latest research papers in the area
2. Forsberg et al., LTE Security, John Wiley & Sons, 2010.
3. Edney, Arbaugh: Real 802.11 Security, Addison-Wesley 2004.
4. Wireless and Mobile Network Security Basics, Edited by Hakima Chaouchi Maryline Laurent-Maknavicius (WILEY Edition).
5. Nicholos Lekkas, Wireless Security, McGraw-Hill, 2000.
6. Kaveh Pahlavan and Prashant Krishnamurthy, Principles of Wireless Networks, Prentice Hall, 2006.

In addition there will be lecture notes and selected articles.