



ہائیر ایجوکیشن کمیشن

HIGHER EDUCATION COMMISSION

Government of Pakistan, Islamabad

Sector H-9
Islamabad, Pakistan
Phone : +92-51-90402111
www.hec.gov.pk
hkasi@hec.gov.pk

Deputy Director (Curriculum)

No. HEC/NCRC/CS&IT/2025/8163

Academics Division

October 14, 2025

SUBJECT: REVISED CURRICULUM FOR DEGREE PROGRAMS IN COMPUTER SCIENCE

The Higher Education Commission (HEC) of Pakistan, as mandated by its law, provides guidance to Higher Education Institutions (HEIs) on curricula for tertiary education levels in alignment with the National Qualifications Framework (NQF). To address evolving academic trends and market demands, HEC has revised the curriculum standards for Computer Science degree programs at NQF levels 5 & 6 in collaboration with EU TVET Sector Support Programme, British Council. These updated standards are aligned with HEC's Undergraduate Education Policy V 1.1 (2023) ensuring coherence with national priorities and adherence to international benchmarks.

2. The revised curriculum for disciplines under the domain of Computer Science is hereby notified. All universities offering these programs are required to align their Computer Science curriculum with these updated standards/framework as the minimum benchmark for quality and compliance. Additionally, the respective departments must develop course contents in accordance with the prescribed framework, ensuring that the programs address both national and local industry needs. The finalized course contents be submitted electronically to this office at the earliest. An electronic copy of the revised curricula is available on HEC's official website.

HIDAYATULLAH KASI

Vice Chancellors/Rectors/Heads

All Public/Private Sector Universities/DAIs

Copy for information to:

- i. ES to Chairman, Higher Education Commission, Islamabad
- ii. ES to Executive Director, Higher Education Commission, Islamabad
- iii. Chairman, National Computing Education Accreditation Council (NCEAC)
- iv. Mr. Abdul Hafeez Abbasi, Technical Advisor, EU TVET Sector Support Programme, British Council
- v. PS to Consultant, Quality Assurance, Higher Education Commission, Islamabad
- vi. PS to Managing Director, NAHE, Higher Education Commission, Islamabad
- vii. PS to Advisor, Human Resource Development Division, Higher Education Commission, Islamabad
- viii. PS to Director General, Academics Division, Higher Education Commission, Islamabad
- ix. PS to Director General, A&A Division, Higher Education Commission, Islamabad
- x. PS to Director General, Quality Assurance, Higher Education Commission, Islamabad
- xi. PS to DG, Higher Education Commission, Regional Centers in Karachi, Lahore, Peshawar & Quetta
- xii. Director, Academics Division, Higher Education Commission, Islamabad
- xiii. Director / In-charge, Higher Education Data Repository, Higher Education Commission, Islamabad

CURRICULUM
OF
BS COMPUTER SCIENCE
ASSOCIATE DEGREE COMPUTING
(2025)



HIGHER EDUCATION COMMISSION
ISLAMABAD – PAKISTAN

Prepared by:
Subject Experts as per Contribution List
Academia and Industry

Facilitated by:
Mr. Hidayatullah Kasi
Deputy Director, Academics Division
Higher Education Commission

Design, Composed and Edited by:
Mr. Tanveer Ali
IT Administrator, Academics Division
Higher Education Commission

Supported by:
EU TVET Sector Support Programme
British Council

TABLE OF CONTENTS

Contributions	iv
Preface	vii
Guiding Principles	1
Minimum standards	1
Course sequences, Titles & credits	1
Course learning outcomes	1
Course syllabus	1
General Education	1
Distribution of Courses in Semesters	1
Implementation	1
Requirements of Field Experience/Internship	2
Requirements for Capstone Project	2
Certification Requirement	2
Entry Provisions at Undergraduate Level	2
Pathway for Graduates with Associate Degree	2
BS Computer Science	2
Program Description	2
Standard Nomenclature	3
Program Learning Outcomes (PLOs)	3
Eligibility Criteria	4
Program Structure	4
General Education	5
Major Courses (Compulsory)	6
Interdisciplinary Courses	7
Scheme of Studies (Semester I – VIII)	8
Specializations in BS Computer Science	11
Guidelines for Specializations	12
Software Engineering	12
Data Science	12
Artificial Intelligence	13
Cyber Security	13
Information Technology	14
Computer Engineering	14
Computer Game Development	15
Multimedia and Animation	15
Robotics	16
Human Computer Interaction	16
Internet of Things (IoT)	17
Network Infrastructure & Cloud Computing	17
Quantum Computing	18
Health Informatics	18
Associate Degree in Computing	19

Standard Nomenclature	19
Eligibility Criteria	19
Program Structure	19
General Education Courses	20
Major Courses (Compulsory)	21
Pool of Electives	21
Scheme of Studies (Semester I – IV)	22
Course Learning Outcomes (CLOs)	24
Mandatory Major Courses	24-26
Specialization Courses	27-64
Certifications	65
Software Engineering Specialization Certifications	65
Human Computer Interaction	65
Certified Professional in Health Information	66
AI Specialization MOOC courses, Certifications and Recommendations	67
Certifications Recommended by PSEB and Pakistan Software Houses Association (P@SHA)	73
Annex-A (Field experience / Internship guidelines – Computing Programs)	75
Annex-B (Final Year Project (FYP) guidelines – Computing Programs)	76
Annex-C (Framework for Integration of International Certifications into BS Computer Science Program)	77

CONTRIBUTIONS

1. Dr. Shafay Shamail, Chairman, National Computing Education Accreditation Council (**Convener**)
2. Dr. Syed Mansoor Sarwar, Vice Chancellor, Qarshi University, Lahore
3. Dr. Umar Qasim, Professor, Department of Computer Science, University of Engineering & Technology, Lahore
4. Prof. Dr. Tariq Rahim Soomro, Rector, IOBM, Karachi
5. Prof. Dr. Muhammad Abbas, Dean, CCSIS, IOBM, Karachi
6. Prof. Muazzam A. Khan Khattak, Professor, Department of Computer Science, Quaid-i-Azam University, Islamabad (**Co-Convener**)
7. Dr. Muhammad Shahzad Asif, Professor, Department of Computer Science, University of Engineering & Technology, Lahore
8. Dr. Sharifullah Khan, Professor, Department of Computer Science, Pak-Austria Fachhochschule Institute of Applied Sciences and Technology, Haripur (**Convener**)
9. Dr. Ahmed Shabbar Kazmi, Professor, University of Central PUNJAB, Lahore
10. Dr. Zubair Ahmed Shaikh, President / Professor, Faculty of Computing, Department of Computer Science, Muhammad Ali Jinnah University, Karachi
11. Prof. Dr. Zunera Jalil, Dean, National Cyber Security, Air University, Islamabad
12. Prof. Ayyaz Hussain, Chairperson Department of Computer Science, Quaid-i-Azam University, Islamabad
13. Dr. Muhammad Zubair, Professor & Dean Faculty of Computer Sciences, Riphah International, Islamabad
14. Dr. Moneeb Gohar, Professor/HoD, Department of Computer Science, Bahria University, Islamabad
15. Dr. Muhammad Ali, Professor, School of CS & IT, Institute of Management Science (IMS), Peshawar
16. Dr. Awais Adnan, Professor, School of CS & IT, Institute of Management Science (IMS), Peshawar
17. Dr. Hameedur Rahman, Professor & Chair, Dept. of Computer Games Development, Air University, Islamabad
18. Prof. Dr. Arshad Aziz, Dean, FEST, Iqra University, Karachi
19. Dr. Ata ul Aziz Ikram, Professor & Head of Creative Technologies, University of Greater Manchester, Islamabad
20. Prof. Dr. Zulfiqar Ali Memon, Professor & Director, Department of CS, National University of Computer and Emerging Sciences, Karachi
21. Prof. Dr. Najeed Ahmed Khan, Professor, NED University of Engineering & Technology, Karachi
22. Dr. Sara Shahzad, Professor, Department of Computer Science, University of Peshawar, Peshawar
23. Prof. Dr. Asad Ali Sheikh, Dean, Faculty of Computer Science, The University of Modern Sciences, Tando Muhammad Khan
24. Prof. Dr. Faisal Riaz, Professor/Director ORIC, CS & IT Department, Mirpur University of Science and Technology (MUST), AJ&K
25. Prof. Dr. Arfan Jaffar, Dean Faculty of CS & IT, Superior University Lahore

26. Prof. Dr Muhammad Sadiq Ali Khan, Chairman, Department of Computer Science (UBIT), University of Karachi, Karachi
27. Prof. Dr. Muhammad Mubashir Khan, Department of Computer Science, NED University of Engineering & Technology, Karachi
28. Prof. Muhammad Shaheen, Dean / Director, Faculty of Engineering & IT, Foundation University, Islamabad
29. Dr. Salman Ahmad, Head of Department, Department of Computer Science & AI, GIKI
30. Dr. Saud Altaf, Professor and Chairperson, Department of Information Engineering Technology, National Skill University, Islamabad
31. Engr. Dr. Muhammad Hanif, Associate Professor, Department of Artificial Intelligence, University of Engineering & Technology, Lahore
32. Dr. Muhammad Shahid Farid, Chairman & Associate Professor, Department of Computer Science, University of the PUNJAB, Lahore
33. Dr. Arslan Shaukat, Tenured Associate Professor, Department of Computer & Software Engineering, National University of Sciences and Technology (NUST)
34. Dr. Momina Moetesum, Head of Software Engineering Department, SEECS-NUST, Islamabad (**Co-Convener**)
35. Dr. Mohsin Ashraf, Associate Professor / HoD, Department of Computer Science, University of Central PUNJAB, Lahore
36. Dr. Subhan Ullah, Associate Professor, Department of Cyber Security, National University of Computer and Emerging Sciences, Islamabad
37. Dr. Farhana Jabeen, Associate Professor & HOD, COMSATS University, Islamabad
38. Dr. Muhammad Rashid, Principal & HOD, National University of Technology (NUTECH), Islamabad
39. Engr. Dr. Surat khan, Professor / Dean FICT, BALOCHISTAN University of Information Technology & Management Sciences, Quetta
40. Dr. Hafiz Muhammad Faisal Shahzad, Associate Professor & Chairman, Department of CS, University of Sargodha
41. Dr. Sadaf Abdul Rauf, Chairperson, Department of Computer Science, Fatima Jinnah Women University, Rawalpindi
42. Dr. Muhammad Ashraf, Chairperson / Associate Professor, Department of Computer Engineering/FICT, BALOCHISTAN University of Information Technology & Management Sciences, Quetta
43. Dr. Zar Nawab Khan, Associate Professor, School of Computing and Emerging Technologies (SCET), Karakoram International University, Gilgit, Gilgit Baltistan
44. Dr. Nadeem Iqbal, HOD Department of Computer Science, Abdul Wali Khan University, Mardan
45. Dr. Shahid Hussain, Associate Professor and Chairperson Computer Science Department, IBA, Karachi
46. Dr. Ihsan Ullah, Associate Professor, Department of CS, University of Balochistan, Quetta
47. Dr. Raees Khan, Director QEC, Information Technology University (ITU), Lahore
48. Dr. Zohaib Iqbal, CEO Quest Ai
49. Dr. Waheed Noor, Director GIL, Balochistan
50. Dr. Moazam Maqsood, Assistant Professor, Electrical & Computer Engineering, Head of Program, BS Robotics, Pak-Austria Fachhochschule Institute of Applied Sciences and Technology, Haripur

51. Dr. Muhammad Munawar Iqbal, Assistant Professor, University of Engineering & Technology, Taxila
52. Dr. Fareeha Zafar, Assistant Professor, Computer Science, Government College University, Lahore
53. Mian Waqas Masud, Director, Fazal Software Technology Park (FSTP)
54. Prof. Dr. Engr. Muhammad Ali Nasir, Advisor, R&ID, HEC
55. Dr. Amjad Hussain, DG Academics & Curriculum, HEC
56. Mr. Nasir Shah, DG Quality Assurance, HEC
57. Mr. Abdul Hafeez Abbasi, Technical Advisor, EU TVET, British Council, Islamabad
58. Dr. Waseemullah, Department of Computer Science, NED University of Engineering & Technology, Karachi
59. Dr. Asif Aziz Memon, Assistant Professor CS, Dawood University of Engineering & Technology, Karachi
60. Dr. Muhammad Kashif Sheikh, CS/IT, Sir Syed University of Engineering & Technology, Karachi
61. Dr. Zunaira Saqib, Director, Skill Development, Pakistan Software Export Board (PSEB)
62. Mr. Raza Sukhera, Director, Skill Development, Pakistan Software Export Board (PSEB)
63. Dr. Talha Farooq Khan, Assistant professor, Faculty of CS (AI), The University of Faisalabad
64. Dr. Umair Abdullah, Chief Technology Officer, MaxRemind (Pvt) Ltd Rawalpindi
65. Mr. Aleem Abbas, Manager of Skills Development and Training, P@SHA
66. Syeda Safa Hassany, Associate, Skills Development, P@SHA
67. Mr. Hidayatullah Kasi, Deputy Director, Academics Division, HEC (**Secretary-NCRC**)
68. Mr. Waqas Haseeb Rattu, Deputy Director, IT, HEC
69. Mr. Tanveer Ali, IT Administrator, Academics Division, HEC (**Coordinator-NCRC**)
70. Mr. Hamza Khan, Project Manager, Representative Coursera, Pakistan
71. Mr. Muhammad Qureshi, Provincial Lead KPGB - British Council
72. Mr. Waseem Jan Allahwala, Manager Procurement, British Council
73. Mr. Adeel Ahmed, Media & Communication Manager, British Council

PREFACE

The revised Computer Science Curriculum (2025) has been formulated in accordance with the Higher Education Commission's Undergraduate Education Policy (UEP) Version 1.1, with the objective of ensuring alignment with international academic and professional standards in the rapidly evolving domains of computing and information technology. This curriculum represents a significant step forward in modernizing IT education in Pakistan at tertiary level, equipping graduates with the skills, knowledge, and ethical grounding necessary to meet emerging challenges in the digital era.

The revision process emphasizes the integration of cutting-edge global trends in areas such as Artificial Intelligence, Data Science, Cybersecurity, Cloud Computing, Software Engineering, and Internet of Things (IoT) etc, while fostering experiential and project-based learning aligned with industry expectations. The curriculum offers flexible specialization tracks, empowering Higher Education Institutions (HEIs) and students to tailor their learning pathways toward specific professional and research-oriented areas, thereby expanding employment and innovation opportunities.

In addition, the framework encourages HEIs to promote industry-recognized certifications through collaborations with leading national and international certification bodies. These certifications may be considered for course mapping and credit substitution, further enhancing the graduates' global competitiveness, professional mobility, and acceptance in international ICT sectors.

The development and revision of this curriculum were made possible through the collaborative efforts of the National Curriculum Revision Committee (NCRC) comprising distinguished academicians, researchers, and industry experts from across Pakistan. Their valuable insights, technical expertise, and commitment to academic excellence have been instrumental in ensuring that this curriculum remains relevant, forward-looking, and responsive to both national needs and global standards.

The Higher Education Commission extends special appreciation to the British Council, European Union TVET Sector Support Programme for their sponsorship and facilitation of the NCRC under this initiative. Their partnership and continued support in advancing curriculum reform in IT domain have been vital in enhancing the quality and global relevance of computing education in Pakistan.

The Higher Education Commission also acknowledges the significant contributions and support of the Pakistan Software Export Board (PSEB), Pakistan Software Houses Association (P@SHA), and the Ministry of Information Technology and Telecommunication (MoITT) for their valuable input and collaboration throughout the curriculum development process.

The Higher Education Commission acknowledges the dedication of the NCRC members, industry partners and all contributors who played a key role in shaping this curriculum. It is anticipated that this revised framework will serve as a comprehensive guide for HEIs, fostering innovation, research, and professional excellence in the field of Computer Science and Information Technology.

Dr. Amjad Hussain
Director General
Academics Division

GUIDING PRINCIPLES

Minimum Standards

The curriculum standards and guidelines prescribed under this policy are mandatory at minimum level. Universities or the concerned departments may however set higher standards provided that the standards prescribed herein are not reduced or compromised.

Course Sequence, Titles and Credits

For BS Computer Science, the sequence of courses prescribed under this document is logically arranged and is suggestive only. The offering department may rearrange the sequence and alter the course titles and credits provided that the essence of the courses prescribed in policy remains intact. The department may add more courses as and when required subject to approval of university's relevant statutory body.

Course Learning Outcomes

Course learning outcomes (CLOs) are the bare minimum standards of learning that students must achieve upon completing a specific course. These outcomes serve as essential benchmarks, ensuring consistency in the quality of education across institutions. The CLOs prescribed herein represent the minimum level of competency and understanding expected from students. While these standards must not be compromised, departments are encouraged to enhance the rigor of the CLOs by incorporating additional learning outcomes, provided these do not alter the essence of the prescribed standards. In this policy, CLOs are exclusively developed for major field courses within the program. For interdisciplinary courses, departments offering these courses are responsible for developing their CLOs in alignment with their respective disciplines and program requirements. For general education courses as required under the HEC Undergraduate Education Policy V 1.1., departments may adopt the CLOs prescribed in the HEC-developed model courses.

Course Syllabus

This document serves as a comprehensive guideline describing the course learning outcomes (CLOs) for each course offered in the Associate Degree, BS in Computer Science as minimum standards. The offering department is mandated to meticulously prepare, modify, and tailor the syllabus of each course, ensuring alignment with the stipulated learning outcomes. It is in this regard imperative that the department utilizes instructional, reference, and reading materials that it deems appropriate to effectively meet the CLOs.

General Education

For Associate degree and BS in Computer Science, the courses prescribed for General Education component must mandatorily be offered with the same titles and credits as prescribed under HEC Undergraduate Education Policy V 1.1. The concerned departments may adopt and follow the learning outcomes and study contents developed by HEC for these courses as available on its website.

Distribution of Courses in Semesters

Universities can redistribute courses across different semesters. This adjustment can be made based on factors such as teaching staff availability and other facilities. The course distribution outlined is not rigid and is subject to modification by the concerned universities.

Implementation

HEIs are advised to implement this revised scheme of studies and align their degree programs and degree nomenclature accordingly, with effective from Spring 2026.

Requirement of Field Experience / Internship

It is a mandatory degree award requirement of three (03) credit hours for BS in Computer Science. Internship of six (06) to eight (08) weeks must be graded by a faculty member in collaboration with the supervisor in the field. This requirement cannot be substituted with additional course work, capstone or project work. Detailed advisory of Field Experience is attached at **Annex-A**.

Requirement of Capstone Project / Final Year Project

It is a mandatory degree award requirement of six (06) credit hours for BS in Computer Science. A capstone project (Final Year Project) is multifaceted body of work that serves as a culminating academic and intellectual experience for students. The FYP must be supervised jointly by Industry Supervisor and graded by a faculty member as per the protocols given at **Annex-B**. This requirement cannot be substituted with additional course work or internship.

Requirement of Certifications

It is a mandatory requirement of three (3) credit hours for BS Computer Science students to earn industry-relevant certifications, preferably through recognized national or international certification bodies, as outlined in **Annex-C**. These certifications are aimed at enhancing employability by equipping graduates with practical, job-ready skills in line with evolving market demands. The certifications may be considered for credit substitution (minimum of one and maximum of six credit hours) upon course mapping and approval by the relevant statutory bodies of the university. In CS program, students may opt to replace certain electives with internationally recognized certifications equivalent to three (03) credit hours. This requirement cannot be substituted with additional course work, internship, or project work.

Entry Provisions at Undergraduate Level

a. Pathway for Graduates with Associate Degree

- Only the students having 14 years qualification in Computing are allowed admission in the fifth semester of the BS in Computer Science. However, as per revised scheme such students are also required to complete deficiency courses from 15-18 credits through bridging semester prior to enrollment in 5th Semester of BS Computer Science Program. The deficiency is for the purpose to make prior learning before 5th semester at par with scheme of BS Computer Science program.
- The minimum eligibility for admission in the fifth semester through above-mentioned pathway is 2.00/4.00 CGPA in the prior qualification i.e., Associate Degree in Computing. The concerned university may, however, set higher eligibility and admission criteria for admission in the fifth semester of BS in Computer Science.

BS COMPUTER SCIENCE

PROGRAM DESCRIPTION

The BS Computer Science is a comprehensive four-year degree aligned with the HEC Undergraduate Education Policy V 1.1. It is designed to equip students with a solid foundation in core computing principles, alongside specialized knowledge in emerging fields such as software engineering, artificial intelligence, data science, cyber security, robotics, gaming design etc. The program emphasizes both theoretical understanding and practical skills, fostering innovative problem-solving, software and hardware development, and critical analysis of technological solutions.

Students will engage in hands-on projects, internships, and research activities that prepare them to address real-world challenges in various sectors including industry, healthcare, entertainment, and national security. The curriculum promotes ethical and professional responsibilities, teamwork, and effective communication, preparing graduates to become adaptable, lifelong learners and leaders in the rapidly evolving field of computing.

STANDARD NOMENCLATURE

For the sake of standardization, it was decided by NCRC that the undergraduate degree program (NQF Level-06) shall have the title of **“BS Computer Science”** with the option of specializations in Software Engineering, Artificial Intelligence, Data Science, Cyber Security, Information Technology, Computer Engineering, Computer Games Development, Multimedia and Animation, Robotics, Human Computer Interaction, Internet of Things (IoT), Network Infrastructure & Cloud Computing, Quantum Computing, Health Informatics. Further specializations as per given framework may be added by HEIs based on availability of faculty, infrastructure and resources. The area of specializations accordingly shall be reflected clearly on the transcript.

PROGRAM LEARNING OUTCOMES / GRADUATE ATTRIBUTES

Keeping in view the transformation from knowledge-based education philosophy to Outcome based education (OBE) system, the OBE model based on Seoul Accord has also been considered. 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).

S. No	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	Understand and assess societal, health, safety, legal, and

	Professionalism and Society	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

ELIGIBILITY CRITERIA

Higher Secondary School Certificate/A-levels (involving minimum 12 years of schooling) or an IBCC equivalent qualification with at least 50% marks and having studied Mathematics is the basic eligibility requirement for admission in all BS Computing Programs.

- The students who have not studied Mathematics at intermediate level have to pass deficiency courses of Mathematics (06 credits) in first year including zero semester.

Additionally, candidates may be required to pass an entry test conducted by the university or admission authority, along with fulfilling any other criteria set by the institution, such as interviews or aptitude assessments.

PROGRAM STRUCTURE OF BS COMPUTER SCIENCE

The degree program follows HEC Undergraduate Education Policy 2023 V 1.1 and comprises a minimum of 08 regular semesters (04 years). Universities may offer courses consisting of a minimum of 130 credit hours provided that the total number of credit hours are reasonably set to achieve the Program Learning Outcomes.

Minimum Credit Hours		130
General Education Courses		34 credit hours (14 courses)
Discipline Related Courses / Major	Total	75 credit hours (26 courses)
	Computer Science Core Courses	48 credit hours (14 courses)
	Specialization Electives	24 credit hours (08 courses)
	Professional Certification	03 credit hours
Interdisciplinary/ Allied Courses		12 credit hours (4 courses)
Field Experience/Internship		3 credit hours
Capstone Project / FYP		6 credit hours
Program Duration		Minimum: 4 Years
		Maximum: 6 Years
		(Further extendable to another year subject to the approval of the university's statutory body)

	following the provisions of HEC Undergraduate Education Policy 2023 V 1.1)
Semester Duration	16-18 weeks for regular semesters (1-2 weeks for examination) 8-9 weeks for summer semesters (1 week for examination)
Course Load (per semester)	15-21 credit hours for regular semesters Credit hours to be offered in Summer/winter semesters, as per HEC guidelines (For remedial/deficiency/failure/repetition courses only)
3 Credit Hours (Theory)	3 classes (1 hour each) OR 2 classes (1.5 hours each) OR 1 class (3 hours) per week throughout the semester.
1 Credit Hours (Practical Work)*	1 Credit hour of practical work requires three contact hours per week throughout the semester.
Policy for Probation in Semester	i) The students acquiring less than 2.00/4.00 GPA in a semester but passing in all papers will be promoted with the condition to achieve more than 2.0 GPA in the next semester and s/he will be put on probation for the next semester. ii) The students acquiring GPA 1.7 and above but failing in any paper(s) will be placed on probation and promoted to the next semester conditionally. They will have to be registered for summer semester to improve the grade. iii) Students acquiring GPA less than 1.7 in two consecutive semesters and failing in any paper(s) even after attending summer semester for one academic year will be dropped from university rolls.

General Education Courses: 34 Credits (14 Courses)

As per HEC UGE Policy V 1.1 and subsequent notifications, following courses are mandatory to be part of every undergraduate degree program including Associate Degree, hence the same are included in the schemes of BS Computer Science;

Serial No.	GE Course Category	No. of Courses	Credit Hours
1	Arts & Humanities *	1	02
2	Natural Sciences *	1	03 (2+1)
3	Social Sciences *	1	02
4	Functional English, Expository Writing	2	06
5	Quantitative Reasoning	2	06
6	Islamic Studies / Ethics	1	02
7	Ideology & Constitution of Pakistan	1	02
8	Application of ICT	1	03 (2+1)
9	Entrepreneurship	1	02
10	Civics and Community Engagement	1	02
11	Pakistan Studies	1	02
12	Fehm e Quran	1	02
	Total	14	34

Note: In the course categories from Serial Number 4-12, HEC has developed the model course outlines for guidance of the Institutions. In the course categories from Serial Number 1-3, the HEIs may choose a course from the pool of subjects under the category approved by statutory bodies of the university.

Major Courses (Compulsory): 48 Credit Hours (14 Courses)

The Core Mandatory courses in a BS Computer Science program help to provide students with a strong and comprehensive foundation in the fundamental areas of BS Computer Science program.

S.No.	Courses	Credit Hours
1	Programming Fundamentals	3+1
2	Object Oriented Programming	3+1
3	Database Systems	3+1
4	Digital Logic Design	3+1
5	Data Structures	3+1
6	Information Security	2+1 / 3+0
7	Artificial Intelligence	2+1 / 3+0
8	Computer Networks	2+1 / 3+0
9	Software Engineering	3+0
10	Computer Organization & Architecture <i>Assembly language included</i>	2+1

11	Operating Systems	3+1
12	Design and Analysis of Algorithms	3+0
13	Theory of Automata	3+0
14	Cloud Computing	2+1 / 3+0
	Total	48 credit hours

Interdisciplinary Courses: Min 12 Credits

As per HEC UGE Policy V 1.1, interdisciplinary courses of Min 12 credit hours are required in four year BS Computer Science program, to complement holistic understanding of the major. The item was thoroughly deliberated in the NCRC of Computing it was emphasized that in order to effectively utilize the compartment of interdisciplinary courses, the list be given so that no important area is left unattended. Universities are at liberty to add more courses to it after approval of HEIs' relevant statutory bodies. However, first two courses in the below mentioned list are mandatory as decided by the NCRC.

S.No	Interdisciplinary / Allied Courses*	Credit Hours
1	Calculus and Analytical Geometry (Mandatory)	3
2	Linear Algebra (Mandatory)	3
3	Discrete Structure	3
4	Probability & Statistics	3
5	Economics	3
6	Organizational Behavior	3
7	Media and Information Literacy	3
8	E-Governance and ICT Policy	3
9	Environmental Science	3
10	Professional Practice	3
11	Digital Marketing & E-Commerce	3
12	Technology Management	3
13	Financial Technology (FinTech) Fundamentals	3
14	Geographic Information Systems (GIS) & Spatial Data	3
15	Bioinformatics	3
16	Smart Cities & Urban Informatics	3
17	Cyber Law & Digital Policy	3
18	Technology and Sustainable Development	3

19	Digital Media & Communication Studies	3
20	Project Management	3
21	Educational Technology (EdTech)	3

SCHEME OF STUDIES

The suggestive scheme of studies for BS Computer Science program is given below;

SEMESTER-I			
S. No	COURSE	CREDIT HOURS	CATEGORY
1	Quantitative Reasoning-I*	3 (3+0)	General Education
2	Functional English*	3 (3+0)	General Education
3	Applications of Information and Communication Technologies*	3 (2+1)	General Education
4	Social Science**	2 (2+0)	General Education
5	Programming Fundamentals	4 (3+1)	Major
6	Calculus & Analytical Geometry – IDS I	3 (3+0)	IDS - Mandatory
Total Credits (18)			

SEMESTER-II			
S. No	COURSE	CREDIT HOURS	CATEGORY
1	Quantitative Reasoning-II*	3 (3+0)	General Education
2	Arts and Humanities**	2 (2+0)	General Education
3	Pakistan Studies*	2 (2+0)	General Education
4	Fehm-e-Quran – I (for Muslim Students)	1 (0+1)	General Education
5	Object Oriented Programming	4 (3+1)	Major
6	Digital Logic Design	4 (3+1)	Major
7	Linear Algebra IDS-II	3 (3+0)	IDS - Mandatory
Total Credits (19)			

SEMESTER-III			
S. No	COURSE	CREDIT HOURS	CATEGORY
1	Expository Writing*	3 (3+0)	General Education
2	Natural Science**	3 (2+1)	General Education
3	Fehm-e-Quran – II (for Muslim Students)	1 (0+1)	General Education
4	Data Structures	4 (3+1)	Major
5	Database Systems	4 (3+1)	Major
6	Operating Systems	4 (3+1)	Major
Total Credits (19)			

SEMESTER-IV			
S. No	COURSE	CREDIT HOURS	CATEGORY
1	Civics and Community Engagement*	2 (2+0)	General Education
2	Ideology and Constitution of Pakistan*	2 (2+0)	General Education
3	Entrepreneurship*	2 (2+0)	General Education
4	Islamic Studies (Religious Education / Ethics for non-Muslim students) *	2 (2+0)	General Education
5	Software Engineering	3 (3+0)	Major
6	Computer Organization & Architecture (Computer Organization & Architecture)	3 (2+1)	Major
7	Design & Analysis of Algorithms	3 (3+0)	Major
Total Credits (17)			

SEMESTER-V			
S. No	COURSE	CREDIT HOURS	CATEGORY
1	Computer Networks	3	Major
2	Information Security	3	Major
3	Artificial Intelligence	3	Major
4	Theory of Automata	3 (3+0)	Major
5	IDS - III	3 (3+0)	IDS
6	IDS - IV	3 (3+0)	IDS
Total Credits (18)			

SEMESTER-VI			
S. No	COURSE	CREDIT HOURS	CATEGORY
1	Cloud Computing	3	Major
2	Elective-I***	3	Major
3	Elective-II***	3	Major
4	Elective-III***	3	Major
5	Elective-IV***	3	Major
Total Credits (15)			

SEMESTER-VII			
S. No	COURSE	CREDIT HOURS	CATEGORY
1	Elective-V***	3	Major
2	Elective-VI***	3	Major
3	Elective-VII***	3	Major
4	Elective-VIII***	3	Major
5	Professional Certification****	3	Certification
Total Credits (15)			

SEMESTER-VIII			
S. No	COURSE	CREDIT HOURS	CATEGORY
1	Final Year Project***** (As per protocols given)	6	Capstone Project
2	Field Experience	3	Field Experience
Total Credits (9)			

Note: Internships of three (03) credit hours must be completed in accordance with HEC Undergraduate Education Policy V 1.1. This requirement cannot be substituted with additional coursework, capstone, research, or project work.

* The university may use HEC-designed model courses.

** The university/offering department may offer any course within the broader subject domain/cluster to meet the given credits.

*** The university/offering department shall offer any advanced course in BS Computer Science from the area of specializations as an elective based on available academic and physical resources.

**** BS Computer Science students are required to complete professional certification(s) (equivalent to 3 credit hours in total) over the period of four-year program as a mandatory condition for degree completion. A certification will be considered equivalent to 1 credit hour if it comprises at least 16 hours of study. The respective department will guide students in selecting relevant certifications, ensuring alignment with current market needs and the program's objectives. **Details given at Annex-C**

***** The Final Year Project may be split in 7th and 8th Semester. Likewise the Field Experience may be offered during summer/winter semester.

SPECIALIZATIONS IN BS COMPUTER SCIENCE PROGRAM:

After detailed consultation in the NCRC, following specializations under BS Computer Science are recommended. HEIs may add more specializations as per availability of faculty, infrastructure and resources keeping in view the framework after approval of their statutory bodies;

1. Software Engineering
2. Data Science
3. Artificial Intelligence
4. Cyber Security
5. Information Technology
6. Computer Engineering
7. Computer Games Development
8. Multimedia and Animation
9. Robotics
10. Human Computer Interaction
11. Internet of Things (IoT)
12. Network Infrastructure & Cloud Computing
13. Quantum Computing
14. Health Informatics

GUIDELINES FOR SPECIALIZATIONS:

A total of 24 credit hours in the BS Computer Science program are allocated to elective courses, which define the specializations offered under the program. Each specialization cluster provides a broad range of elective courses, from which students must select 7–8 courses to complete the required 24 credit hours.

The specialization title may appear either on the degree (e.g., BS Computer Science – Artificial Intelligence) or only on the transcript, as per the approval of the university's statutory bodies.

Universities also have the flexibility to offer a generic BS Computer Science program, wherein students may choose electives from any cluster without committing to a specific specialization. In such cases, no specialization title will appear on the degree or transcript.

1. Software Engineering

SR NO	COURSES	Credit Hour
1.	Software Requirement Engineering	3+0
2.	Software Project Management	3+0
3.	Software Quality and Testing	2+1
4.	DevOps Principles and Practices	2+1/3+0
5.	Software Re-Engineering	2+1/3+0
6.	Software Design and Architecture	3+0
7.	Parallel and Distributed Computing	2+1/3+0
8.	Software Construction and Development	2+1/3+0
9.	Human Computer Interaction	2+1/3+0
10.	Formal Methods and Design	3+0
11.	Cross-platform Application Development	3+0
12.	Advanced Topics in Software Engineering	2+1/3+0
13.	Machine Learning	2+1
14.	Internet of Things and It's Applications	2+1
15.	Green Software Engineering	3+0

2: Data Science

SR NO	COURSES	Credit Hour
1.	Introduction to Data Science	3+0
2.	Computational Statistics	3+0
3.	Data Engineering	3+0
4.	Data Visualization	3+0
5.	Business Intelligence	3+0
6.	Tools and Techniques in Data Science	3+0
7.	Big Data Analytics	3+0
8.	Machine Learning	3+0

9.	Deep Learning	3+0
10.	Natural Language Processing	3+0
11.	Data Ethics & Security	3+0
12.	Computer Vision	3+0
13.	Information Retrieval	3+0
14.	Generative AI	3+0
15.	Data Mining	3+0

3: Artificial Intelligence

SR NO	COURSES	Credit Hour
1.	Programming for AI	3+0
2.	Machine Learning	3+0
3.	Deep Learning	3+0
4.	Knowledge representation and Reasoning	3+0
5.	Generative AI	3+0
6.	Computer Vision	3+0
7.	Natural Language Processing	3+0
8.	Data Mining	3+0
9.	Reinforcement Learning	3+0
10.	Optimization Techniques	3+0
11.	Stochastic Processes	3+0
12.	Machine learning Operations (MLOps)	3+0
13.	Agentic AI	3+0
14.	Speech Processing	3+0
15.	Evolutionary Computing & Swarm Intelligence	3+0

4: Cyber Security

SR NO	COURSES	Credit Hour
1.	Information Assurance	3+0
2.	Network Security	2+1
3.	Secure Software Design and Development	2+1
4.	Digital Forensics and Incident Response	2+1
5.	Applied Cryptography	2+1
6.	Vulnerability Assessment and Penetration Testing	2+1
7.	Ethics in Cyber Security	2+1
8.	Cloud Security	2+1
9.	Embedded Systems Security	2+1
10.	Cyber Threat Intelligence	2+1

11.	Artificial intelligence for Cyber Security	2+1
12.	IoT Security	2+1
13.	Open-Source Intelligence Techniques	2+1
14.	Usable Security and Privacy	2+1
15.	Malware Analysis and Reverse Engineering	2+1
16.	Security Orchestration, Automation, and Response	2+1
17.	Cyber Security Policy and Governance	3+0
18.	Security Analytics and DevSecOps	2+1
19.	Quantum Security	2+1
20.	Blockchain and Web 3 Security	2+1
21.	Wireless and Mobile Security	2+1

5: Information Technology

SR NO	COURSES	Credit Hour
1.	Web Technologies	2+1
2.	Cyber Security	2+1
3.	Database Administration & Management	2+1
4.	System & Network Administration	2+1
5.	Information Technology Infrastructure	2+1
6.	Network Security	2+1
7.	Data Communication	2+1
8.	Wireless and Mobile Networks	2+1
9.	Cloud Infrastructure and Services	2+1
10.	Email Systems and Server Management	2+1
11.	Ethical Hacking and Penetration Testing	2+1
12.	Security Policies and Compliance	3+0
13.	IT Project Management	3+0
14.	Systems Analysis and Design	3+0
15.	Technology Lifecycle Planning	3+0

6: Computer Engineering

SR NO	COURSES	Credit Hour
1.	*Linear Circuit Analysis	3+0
2.	*Electrical Network Analysis	3+0
3.	*Electronic Devices and Circuits	3+0
4.	**Signals and Systems	3+0
5.	**Control Engineering	3+0
6.	**Digital System Design	3+0

7.	**Digital Image Processing	3+0
8.	***Parallel and Distributed Computing	3+0
9.	***Parallel Computer Architecture	3+0
10.	***Computer Interfacing	3+0
11.	***HCI & Computer Graphics	3+0
12.	***Embedded Systems	3+0
13.	***Deep Learning	3+0
14.	*** Microprocessor and Microcontroller based Design	3+0

*,**,*** Choose at least 2 courses from each course group

7: Computer Games Development

SR NO	COURSES	Credit Hour
1.	Game Design Principles	3 (2+1)
2.	Graphic Design and Illustration Art	3 (2+1)
3.	Game Programming	3 (2+1)
4.	Game Engine Architecture	3 (2+1)
5.	3D Design and Modelling	3 (2+1)
6.	3D Animation and Visual Effects Production	3 (2+1)
7.	UI/UX Design and Development	3 (2+1)
8.	Story Boarding and Narrative Development	3 (2+1)
9.	Virtual & Augmented Reality for Games	3 (2+1)
10.	Simulation & Digital Twins in Games	3 (2+1)
11.	Web3 Gaming	3 (2+1)
12.	Sound Design for Games	3 (2+1)
13.	LiveOps And Monetization for Games	3 (2+1)
14.	Game Design Research Methods	3 (2+1)
15.	Game Testing	3 (2+1)

8: Multimedia and Animation

SR NO	COURSES	Credit Hour
1.	Fundamentals of Drawing	3 (2+1)
2.	Graphic Design and Illustration Art	3 (2+1)
3.	Animation Programming	3 (2+1)
4.	2D & 3D Animation Principles	3 (2+1)
5.	3D Design and Modeling	3 (2+1)
6.	3D Animation and Visual Effects Production	3 (2+1)
7.	Character Design & Rigging	3 (2+1)
8.	UI/UX Design and Development	3 (2+1)
9.	Story Boarding and Narrative Development	3 (2+1)

10.	Virtual Production & Immersive Media	3 (2+1)
11.	Digital Film Production & Cinematography	3 (2+1)
12.	Sound Design & Foley Production	3 (2+1)
13.	Motion Graphics & Title Design	3 (2+1)
14.	Digital Media Marketing & Monetization	3 (2+1)
15.	Media Testing, Quality & Post-Production	3 (2+1)

9: Robotics

SR NO	COURSES	Credit Hour
1.	*Introduction to Robotics	3+0
2.	*Electric Circuits	3+0
3.	*Basic of Electronics and Components	3+0
4.	*Instrumentation and Measurement	3+0
5.	**Signals and Systems	3+0
6.	**Digital Signal Processing	3+0
7.	**Linear Control Systems	3+0
8.	***Robot Mechanics-1 (Statics)	3+0
9.	***Robot Mechanics-2 (Dynamics)	3+0
10.	***Actuators	3+0
11.	****Introduction to Embedded Systems	3+0
12.	****Machine Learning	3+0
13.	****3D Machine Vision	3+0
14.	****Localization and Mapping	3+0
15.	****Aerial Drones	3+0
16.	****Navigation and Mobile Platforms	3+0
17.	****Speech Recognition and Speech Synthesis	3+0
18.	****Human Robot Interaction	3+0

*, **, ***, **** Choose at least 2 courses from each course group

10: Human Computer Interaction

SR NO	COURSES	Credit Hour
1.	Human Computer Interaction	3+0
2.	User-Centered Design and Prototyping	3+0
3.	Design Thinking and Innovation	3+0
4.	Usability Engineering	3+0
5.	Rapid Application Design	3+0
6.	Cognitive Psychology for Interface Design	3+0
7.	Accessibility and Inclusive Design	3+0
8.	Social Computing and Online Communities	3+0

9.	Visual Design and Information Architecture	3+0
10.	HCI in Business and Enterprise Systems	3+0
11.	HCI in Healthcare Systems	3+0
12.	HCI in Education and Learning Technologies	3+0
13.	Multimedia Application Design	3+0
14.	Game Design and Interactive Environments	3+0
15.	Conversational Interfaces and Intelligent Agents	3+0
16.	HCI and Cognitive Models in AI Systems	3+0
17.	Capstone Project in Human-Centered Design	1+2

11: Internet of Things (IoT)

SR NO	COURSES	Credit Hour
1.	Microcontroller Architecture and System Design	3+1
2.	Embedded Programming	3+1
3.	Sensors, Transducers, and Interfacing	2+1
4.	Real-Time Operating Systems	2+1
5.	IoT Communication and Networks	2+1
6.	IoT Application Layer Protocol	3+0
7.	Edge AI	3+0
8.	Machine Learning	3+0
9.	IoT Cloud	3+0
10.	IIOT	3+0
11.	Generative AI	3+0
12.	IoT Data Analytics and Visualization	2+1
13.	IoT Security	3+0
14.	Wireless & 5G for IoT	3+0
15.	Emerging Applications of IoT	2+1

12: Network Infrastructure & Cloud Computing

SR NO	COURSES	Credit Hour
1.	Software Development and IT Operations (DevOps)	(2+1)/3
2.	Development, Security, and Operations (DevSecOps)	(2+1)/3
3.	Machine Learning Operations (MLOps)	(2+1)/3
4.	Cloud Design and Implementation	(2+1)/3
5.	Network Management	(2+1)/3
6.	Internetworking	(2+1)/3
7.	Machine Learning	(2+1)/3
8.	Network Security	(2+1)/3
9.	Software Defined Networks (SDN)	(2+1)/3

10.	Wireless and Mobile Networks	(2+1)/3
11.	Enterprise Network Infrastructure Engineering Design	(2+1)/3
12.	Network Programmability and Automation	(2+1)/3
13.	Next-Generation Networks and Edge Computing	3+0
14.	Cloud-Native Application Development	(2+1)/3
15.	Infrastructure as Code (IaC)	3+0
16.	Microservices Architecture and Docker Containers	3+0
17.	Network Automation and Orchestration	(2+1)/3
18.	Cloud Load Balancing	(2+1)/3
19.	Network Performance	(2+1)/3
20.	System Programming	

13: Quantum Computing

SR NO	COURSES	Credit Hour
1	Mathematical Foundations for Quantum Computing	3+0
2	Quantum Physics for Computer Scientists	3+0
3	Introduction to Quantum Computing	3+0
4	Quantum Circuits and Gates	3+0
5	Quantum Programming and Simulation	3+0
6	Quantum Algorithms	3+0
7	Quantum Information Processing	3+0
8	Quantum Error Correction and Fault Tolerance	3+0
9	Quantum Cryptography and Security	3+0
10	Post Quantum Cryptography	3+0
11	Quantum Hardware and Emerging Technologies	3+0
12	Quantum Machine Learning	3+0
13	Quantum Networking	3+0
14	Quantum Internet	3+0
15	Advanced Quantum Applications	3+0

14: Health Informatics

SR NO	COURSES	Credit Hour
1	Introduction to Health Informatics & Public Health	3+0
2	Statistics for Health Informatics	3+0
3	Natural Language Processing in Healthcare	2+1
4	Health Data Modalities and Visualization	2+1
5	Big Data Analytics in Healthcare	3+0/2+1
6	Machine Learning for Healthcare	2+1
7	Health Data Standards and Interoperability	3+0
8	Information Security and Privacy	3+0

9	Health Data Governance and Ethics	3+0
10	Blockchain and Secure Health Data Exchange	3+0
11	Health Information Systems and Electronic Health Records	2+1
12	Project Management for Healthcare	2+1
13	Telehealth Systems and Remote Monitoring	3+0/2+1
14	Predictive Analytics and Decision Support Systems	2+1
15	Genomic Data Science and Precision Medicine	3+0
16	Leadership and Innovation in Health Informatics	3+0
17	Data Management in Healthcare	3+1

Note: *The universities who intend to continue the existing programs of BS Computing Specialization degrees other than BS Computer Science are allowed to continue and get accreditation from NCEAC, however such degrees shall not be considered equivalent to BS Computer Science.*

ASSOCIATE DEGREE – COMPUTING

STANDARD NOMENCLATURE

For the sake of standardization, it was decided by NCRC that the undergraduate degree program (NQF Level-05) shall have the title of **“Associate Degree - Computing”**.

ELIGIBILITY CRITERIA

Higher Secondary School Certificate/A-levels (involving minimum 12 years of schooling) or an IBCC equivalent qualification with at least 45% marks and having studied Mathematics is the basic eligibility requirement for admission in all Associate Degrees in Computing Programs.

- The students who have not studied Mathematics at intermediate level have to pass deficiency courses of Mathematics (06 credits) in first year including zero semester.

Additionally, candidates may be required to pass an entry test conducted by the university or admission authority, along with fulfilling any other criteria set by the institution, such as interviews or aptitude assessments.

PROGRAM STRUCTURE OF ASSOCIATE DEGREE COMPUTING

The degree program follows HEC Undergraduate Education Policy 2023 V 1.1 and comprises a minimum of 04 regular semesters (02 years). Universities may offer courses consisting of a minimum of 72 credit hours provided that the total number of credit hours are reasonably set to achieve the Program Learning Outcomes.

Minimum Credit Hours		72
General Education Courses		34 credit hours (14 courses)
Discipline Related Courses / Major	Total	38 credit hours (11 courses)
	Computing Core	26 credit hours (7 courses)
	Specialization Electives	12 credit hours (3-4 courses)
Program Duration		Minimum: 2 Years Maximum: 3.5 Years
Semester Duration		16-18 weeks for regular semesters (1-2 weeks for examination) 8-9 weeks for summer semesters (1 week for examination)
Course Load (per semester)		15-21 credit hours for regular semesters Credit hours to be offered in Summer/winter semesters, as per HEC guidelines (For remedial/deficiency/failure/repetition courses only)
3 Credit Hours (Theory)		3 classes (1 hour each) OR 2 classes (1.5 hours each) OR 1 class (3 hours) per week throughout the semester.
1 Credit Hours (Practical Work)*		1 Credit hour of practical work requires three contact hours per week throughout the semester.

General Education Courses: 34 Credits (14 Courses)

As per HEC UGE Policy V 1.1 and subsequent notifications, following courses are mandatory to be part of every undergraduate degree program including Associate Degree, hence the same are included in the schemes of Associate degree Computing;

Serial No.	GE Course Category	No. of Courses	Credit Hours
1	Arts & Humanities *	1	02
2	Natural Sciences *	1	03 (2+1)
3	Social Sciences *	1	02
4	Functional English, Expository Writing	2	06
5	Quantitative Reasoning	2	06
6	Islamic Studies / Ethics	1	02
7	Ideology & Constitution of Pakistan	1	02
8	Application of ICT	1	03 (2+1)
9	Entrepreneurship	1	02
10	Civics and Community Engagement	1	02
11	Pakistan Studies	1	02
12	Fehm e Quran	1	02
	Total	14	34

Major Courses (Compulsory): 26 Credit Hours (07 Courses)

The Core Mandatory courses in Associate Degree Computing program help to provide students with a strong and comprehensive foundation in the fundamental areas of computing.

S.No.	Courses	Credit Hours
1	Programming Fundamentals	3+1
2	Object Oriented Programming	3+1
3	Database Systems	3+1
4	Data Structures	3+1
5	Computer Networks	2+1 / 3+0
6	Software Engineering	3+0
7	Operating Systems	3+1
	Total	26 credit hours

Pool of Electives

S. No.	Courses	Credit Hours
1	Introduction to Cloud Fundamentals	3
2	Web Development (Frontend and Backend)	3
3	Mobile App Development	3

4	Data Analytics and SQL for Practitioners	3
5	Introduction to Machine Learning and AI Tools	3
6	Cybersecurity Fundamentals	3
7	DevOps and CI/CD Essentials	3
8	Database Administration and SQL Tuning	3
9	UX/UI and Product Prototyping	3
10	Internet of Things (IoT) Basics	3
11	Practical Python Scripting and Automation	3
12	Applied Networking and Wireless Technologies	3
13	Low-code/No-code Application Development	3
14	Blockchain and Smart Contract Basics	3
15	Digital Logic Design	3
16	Information Security	3
17	Computer Organization & Architecture	3
18	Design and Analysis of Algorithms	3
19	Theory of Automata	3
20	Software Project Management	3

Note: HEP's may add further courses in the above pool of electives upon approval from their statutory bodies.

SCHEME OF STUDIES FOR ASSOCIATE DEGREE COMPUTING

SEMESTER-I			
S. No	COURSE	CREDIT HOURS	CATEGORY
1	Quantitative Reasoning-I*	3 (3+0)	General Education
2	Functional English*	3 (3+0)	General Education
3	Applications of Information and Communication Technologies*	3 (2+1)	General Education
4	Social Science**	2 (2+0)	General Education
5	Programming Fundamentals	3+1	Major
6	Elective - I	3	Major
Total Credits (18)			

SEMESTER-II			
S. No	COURSE	CREDIT HOURS	CATEGORY
1	Quantitative Reasoning-II*	3 (3+0)	General Education
2	Arts and Humanities**	2 (2+0)	General Education
3	Pakistan Studies*	2 (2+0)	General Education
4	Fehm-e-Quran – I (for Muslim Students)	1 (0+1)	General Education
5	Object Oriented Programming	3+1	Major
6	Data Structures	3+1	Major
7	Elective - II	3	Major
Total Credits (19)			

SEMESTER-III			
S. No	COURSE	CREDIT HOURS	CATEGORY
1	Expository Writing*	3 (3+0)	General Education
2	Natural Science**	3 (2+1)	General Education
3	Fehm-e-Quran – II (for Muslim Students)	1 (0+1)	General Education
4	Computer Networks	3	Major
5	Database Systems	3+1	Major
6	Elective-III	3	Major
Total Credits (17)			

SEMESTER-IV			
S. No	COURSE	CREDIT HOURS	CATEGORY
1	Civics and Community Engagement*	2 (2+0)	General Education
2	Ideology and Constitution of Pakistan*	2 (2+0)	General Education
3	Entrepreneurship*	2 (2+0)	General Education
4	Islamic Studies (Religious Education / Ethics for non-Muslim students) *	2 (2+0)	General Education
5	Software Engineering	3+0	Major
6	Operating Systems	3+1	Major
7	Elective-IV	3	Major
Total Credits (18)			

COURSE LEARNING OUTCOMES (CLOS):

MANDATORY MAJOR COURSES

1. Programming Fundamentals

By the end of this course student will be able to:

- Demonstrate proficiency in writing, debugging, and executing basic programs using programming languages such as C or Python.
- Explain core programming concepts including variables, control structures, functions, and data types.
- Develop simple algorithms to solve computational problems.
- Apply best practices in coding to produce efficient and readable programs.
- Analyze program outputs and troubleshoot common errors effectively.

2. Object Oriented Programming

By the end of this course student will be able to:

- Understand and apply the principles of object-oriented programming, including encapsulation, inheritance, and polymorphism.
- Design and implement classes and objects to model real-world entities.
- Write reusable and modular code using OOP concepts.
- Analyze the advantages of OOP over procedural programming paradigms.
- Develop small-scale applications utilizing OOP concepts in languages like Java or C++.

3. Database Systems

By the end of this course student will be able to:

- Design and normalize relational database schemas based on user requirements.
- Write SQL queries for data retrieval, insertion, update, and deletion.
- Explain the concepts of database transactions, concurrency, and recovery.
- Implement basic database management tasks using popular database management systems.
- Analyze the role of databases in information systems and their security considerations.

4. Digital Logic Design

By the end of this course student will be able to:

- Draw and interpret digital logic diagrams, including combinational and sequential circuits.
- Design basic digital components such as multiplexers, flip-flops, and encoders.
- Analyze the behavior of digital systems using truth tables and Boolean algebra.
- Implement simple digital circuits using logic gates.
- Understand the fundamentals of digital system design and their applications.

5. Data Structures

By the end of this course student will be able to:

- Implement and analyze various data structures such as arrays, linked lists, stacks, queues, trees, and graphs.
- Select appropriate data structures to optimize algorithm performance.
- Demonstrate proficiency in traversing and manipulating data structures.
- Evaluate the efficiency of algorithms based on data structure choices.
- Solve complex problems using suitable data structures and algorithms.

6. Information Security

By the end of this course student will be able to:

- Explain fundamental concepts of information security, including confidentiality, integrity, and availability.
- Identify common security threats and vulnerabilities.
- Apply security measures such as encryption, authentication, and access control.
- Conduct basic security audits and risk assessments.
- Understand legal and ethical issues related to information security.

7. Artificial Intelligence

By the end of this course student will be able to:

- Describe core AI concepts including search algorithms, knowledge representation, and reasoning.
- Implement basic AI algorithms for problem-solving and decision-making.
- Analyze the applications of AI in real-world scenarios.
- Discuss ethical considerations and limitations of AI systems.
- Develop simple AI models using appropriate tools and frameworks.

8. Computer Networks

By the end of this course student will be able to:

- Explain the fundamental concepts of computer networking, including protocols, topologies, and models (OSI, TCP/IP).
- Configure and troubleshoot basic network devices and connections.
- Analyze network security threats and mitigation techniques.
- Demonstrate understanding of data transmission and error handling.
- Design simple network architectures to meet organizational needs.

9. Software Engineering

By the end of this course student will be able to:

- Apply software development life cycle models to manage projects effectively.
- Develop software requirements specifications and design documents.
- Implement and test software applications using best practices.
- Analyze and manage software project risks and quality.
- Collaborate effectively in team-based software projects.

10. Computer Organization & Architecture

By the end of this course student will be able to:

- Describe the structure and function of computer components such as CPU, memory, and I/O devices.
- Write and understand basic assembly language programs.
- Analyze how hardware components interact during program execution.
- Explain the concepts of instruction set architecture and microarchitecture.
- Optimize programs considering hardware limitations.

11. Operating Systems

By the end of this course student will be able to:

- Explain the functions and services provided by operating systems.
- Manage processes, threads, and synchronization mechanisms.
- Analyze memory management techniques and file systems.
- Implement basic scheduling algorithms.
- Evaluate operating system performance and security features.

12. Design and Analysis of Algorithms

By the end of this course student will be able to:

- Design efficient algorithms for common computational problems.

- Analyze algorithm complexity using Big O notation.
- Solve problems involving recursion, divide-and-conquer, and dynamic programming.
- Compare different algorithmic approaches to problem-solving.
- Demonstrate correctness and optimality of algorithms.

13. Theory of Automata

By the end of this course student will be able to:

- Explain the concepts of finite automata, regular expressions, and formal languages.
- Design automata to recognize specific languages.
- Demonstrate the equivalence of automata, regular expressions, and grammars.
- Analyze the limitations of finite automata and context-free grammars.
- Apply automata theory to compiler design and language processing.

14. Cloud Computing

By the end of this course student will be able to:

- Describe fundamental cloud computing models and services (IaaS, PaaS, SaaS).
- Deploy and manage applications in cloud environments.
- Analyze the benefits and challenges of cloud computing.
- Implement basic cloud security and compliance measures.
- Evaluate cloud solutions for scalability, cost, and performance.

15. Calculus and Analytical Geometry (IDS – Mandatory Course)

By the end of this course student will be able to:

- Understand and apply the fundamental concepts of limits, derivatives, and integrals to solve mathematical problems.
- Analyze and interpret geometric problems involving equations of lines, circles, and conic sections in the coordinate plane.
- Develop problem-solving skills by applying calculus techniques to real-world scenarios in science and engineering.
- Evaluate the behavior of functions through differentiation and integration, including applications such as optimization and area calculations.
- Communicate mathematical reasoning effectively through written explanations and problem-solving procedures.

16. Linear Algebra (IDS – Mandatory Course)

By the end of this course student will be able to:

- Understand and perform operations on matrices and vectors, including addition, multiplication, and inverse calculations.
- Solve systems of linear equations using matrix methods such as Gaussian elimination and matrix inversion.
- Analyze vector spaces, subspaces, basis, and dimension to understand the structure of linear systems.
- Apply eigenvalues and eigenvectors in solving problems related to transformations and stability analysis.
- Utilize linear algebra concepts to model and solve practical problems in computer science, such as algorithms and data representations.

COURSE LEARNING OUTCOMES (CLOS):

SPECIALIZATIONS:

1. SOFTWARE ENGINEERING ELECTIVES:

1. Software Requirement Engineering

By the end of this course the student will be able to:

- Explain the principles of requirements elicitation, analysis, and specification. **(C2: Understand)**
- Apply structured techniques to document software requirements. **(C3: Apply)**
- Analyze stakeholder requirements to identify conflicts and derive feasible solutions. **(C4: Analyze)**

2. Software Project Management

By the end of this course the student will be able to:

- Explain fundamental project management concepts. **(C2: Understand)**
- Apply project planning techniques to software projects. **(C3: Apply)**
- Analyze risks and constraints in project execution. **(C4: Analyze)**
- Use project management tools to plan and track progress. **(C3: Apply)**
- Demonstrate leadership and teamwork in simulated project environments. **(A4: Characterize)**

3. Software Quality and Testing

By the end of this course the student will be able to:

- Explain software testing and quality assurance principles. **(C2: Understand)**
- Apply the concepts of software verification and validation. **(C3: Apply)**
- Apply various software testing techniques in development project. **(C3: Apply)**
- Design automated test cases. **(C5: Design)**

4. DevOps Principles and Practices

By the end of this course the student will be able to:

- Explain the core principles and cultural shifts necessary for successful DevOps adoption. **(C2: Understand)**
- Implement automated pipelines for continuous integration and continuous delivery of software. **(C3: Apply)**
- Use industry-standard tools and techniques to build and deploy modern, cloud-based applications. **(C3: Apply)**
- Use collaborative tools and scripting languages to foster teamwork in DevOps practices. **(A4: Characterize)**

5. Software Re-Engineering

By the end of this course the student will be able to:

- Understand the concepts and technique of software re-engineering. **(C2: Understand)**
- Apply reengineering techniques to maintain and modify software. **(C3: Apply)**
- Analyze maintenance issues associated with software systems. **(C4: Analyze)**
- Identify solutions to mitigate software maintenance issues. **(C2: Understand)**

6. Software Design and Architecture

By the end of this course the student will be able to:

- Describe key design principles and architectural styles. **(C2: Understand)**
- Apply modeling techniques to represent software subsystems. **(C3: Apply)**
- Collaborate in teams to present design specifications. **(A4: Characterize)**

7. Parallel and Distributed Computing

By the end of this course the student will be able to:

- Explain concepts of concurrency, synchronization, and distributed architectures. **(C2: Understand)**
- Apply parallel programming libraries to solve computational problems. **(C3: Apply)**
- Analyze the performance trade-offs between sequential and parallel solutions. **(C4: Analyze)**

8. Software Construction and Development

By the end of this course the student will be able to:

- Understand abstraction, modularity, concurrency, and software design patterns. **(C2: Understand)**
- Develop optimized, error-free, and reusable code using OOP constructs. **(C5: Design)**
- Design and develop a small-scale software system collaboratively. **(C5: Design)**

9. Human Computer Interaction

By the end of this course the student will be able to:

- Explain fundamental principles of user-centered design. **(C2: Understand)**
- Analyze user feedback to refine interface prototypes. **(C4: Analyze)**
- Design interactive prototypes using modern HCI tools. **(C5: Design)**
- Identify accessibility and inclusivity as essential elements of HCI. **(C2: Understand)**

10. Formal Methods and Design

By the end of this course the student will be able to:

- Explain the role of formal methods in software specification and verification. **(C2: Understand)**
- Apply formal mathematical methods and tools to solve elementary computation problems. **(C3: Apply)**
- Use mathematics based techniques and tools for specifying, designing, and verifying software. **(C3: Apply)**

11. Cross-Platform Application Development

By the end of this course the student will be able to:

- Explain the principles, architectures, and frameworks used in cross-platform application development. **(C2: Understand)**
- Design and develop applications using cross-platform frameworks. **(C5: Design)**
- Utilize client-side and server-side APIs, tools, and libraries. **(C3: Apply)**
- Demonstrate responsible engineering practices by designing and deploying secure, accessible, and user-centric cross-platform solutions. **(A3: Valuing)**

12. Advanced Topics in Software Engineering

By the end of this course the student will be able to:

- Explain the latest trends and emerging technologies in the software development industry. **(C2: Understand)**
- Apply new technologies in software engineering projects. **(C3: Apply)**
- Analyze professional practices in software engineering. **(C4: Analyze)**

13. Machine Learning

By the end of this course the student will be able to:

- Explain core concepts of Machine Learning. **(C2: Understand)**
- Apply machine learning algorithms to real-world datasets. **(C3: Apply)**
- Analyze the performance of ML models using evaluation metrics. **(C4: Analyze)**
- Implement ML solutions using modern toolkits. **(C3: Apply)**
- Discuss ethical considerations while applying ML to societal problems. **(A3: Valuing)**

14. Internet of Things and its Application

By the end of this course the student will be able to:

- Differentiate between the layers of the IoT stack based on the TCP/IP model, demonstrating familiarity with the key technologies and protocols employed at each layer. **(C4: Analyze)**
- Implement IoT applications by integrating sensors and communication modules. **(C3: Apply)**
- Understand performance, scalability, security, and privacy issues in IoT solutions across different application domains. **(C2: Understand)**
- Build a functional IoT system by prototyping, programming, and analyzing data, ensuring the system meets specified project requirements. **(C5: Design)**
- Evaluate the role of big data, machine learning, and cloud/fog/edge computing, in an IoT solution, assessing how each component contributes to the system's functionality and effectiveness. **(C6: Evaluate)**

15. Green Software Engineering

By the end of this course the student will be able to:

- Explain the principles of green software engineering and sustainability in the context of software design, development, and deployment. **(C2: Understand)**
- Analyze energy efficiency, environmental impact, and sustainability challenges in software systems across their lifecycle. **(C4: Analyze)**
- Apply sustainable development practices in software design and implementation. **(C3: Apply)**
- Demonstrate ethical and responsible practices in developing, maintaining, and updating software systems to support long-term social and environmental sustainability. **(A3: Valuing)**

2. DATA SCIENCE ELECTIVES:

1. Introduction to Data Science

By the end of this course the student will be able to:

- Understand the fundamental principles of data science. C2 (Understand)
- Apply EDA and the Data Science processes. C3 (Apply)
- Apply basic machine learning algorithms to solve real-world problems of moderate complexity. C3 (Apply)

2. Computational Statistics

By the end of this course the student will be able to:

- Understand the fundamental statistics for data science and applications of statistics in data science. C1 (Knowledge)
- Apply Statistical techniques in real-life problems. C3 (Apply)
- Apply basic data science statistical techniques by using tools on real-world datasets. C3 (Apply)

3. Data Engineering

By the end of this course the student will be able to:

- Explain the principles of data engineering, including data pipelines, ETL processes, and storage systems. C3 (Understand)
- Apply data integration and transformation techniques to build scalable and efficient data pipelines. C3 (Apply)
- Design and implement data architectures for big data and real-time analytics using modern tools and platforms. C3 (Apply)

4. Data Visualization

By the end of this course the student will be able to:

- Describes the basic concept and techniques for any given raw data visualization. C3 (Understand)
- Select and apply a proper data mining algorithm to discover interesting patterns C3 (Apply)
- Analyze and extract patterns to solve problems and point out how to deploy solution C4 (Analyze)
- Evaluate systematically supervised, semi-supervised, and unsupervised models and algorithms with respect to their accuracy C4 (Analyze)

5. Business Intelligence

By the end of this course the student will be able to:

- Explain the concepts, architecture, and components of Business Intelligence systems.C2 (Understand)
- Apply BI tools and techniques to extract, transform, and visualize data for decision-making.C3 (Apply)
- Design and evaluate BI solutions to support strategic and operational business decisions.. C3 (Evaluate)

6. Tools and Techniques in Data Science

By the end of this course the student will be able to:

- Explain the purpose and functionality of essential tools and techniques used in data science workflows.C2 (Understand)
- Apply data preprocessing, transformation, and visualization techniques using appropriate tools to solve practical problems..C3 (Apply)
- Design and implement integrated data science workflows using multiple tools for real-world datasets.C6(Create)

7. Big Data Analytics

By the end of this course the student will be able to:

- Understand the fundamental concepts of Big Data and its programming paradigm. C3(Understand)
- Hadoop/MapReduce Programming, Framework, and Ecosystem C2 (Understand)
- Apache Spark Programming C3 (Apply)

8. Machine Learning

By the end of this course the student will be able to:

- Understand the fundamentals of Machine Learning used in Data Science C2 (Understand)
- Apply Machine Learning for classification Problems C3 (Apply)
- Apply Machine learning algorithms to real-world problems C3 (Apply)

9. Deep Learning

By the end of this course the student will be able to:

- Understand the fundamentals of neural networks in AI C2 (Understand)
- Explain how simple ANNs can be designed C2 (Understand)
- Apply deep learning algorithms to real-world classification problems C3 (Apply)
- Analyze results from deep learning to select appropriate solutions C4 (Analyze)

10. Natural Language Processing

By the end of this course the student will be able to:

- Explain the fundamental concepts and techniques of Natural Language Processing, including tokenization, POS tagging, and parsing. C2 (Understand)
- Apply NLP libraries and frameworks (e.g., NLTK) to preprocess and analyze text data for real-world applications.C3 (Apply)
- Design and implement NLP-based solutions such as sentiment analysis or text classification using machine learning models. C6 (Create)

11. Data Ethics & Security

By the end of this course the student will be able to:

- Explain the principles of data ethics, privacy laws, and governance frameworks.C2 (Understand)
- Analyze ethical and legal implications of data collection, storage, and sharing in real-world scenarios.C4 (Analyze)
- Evaluate compliance strategies for data protection and governance in organizational contexts.C5 (Evaluate)
- Design policies and guidelines to ensure ethical data usage and regulatory compliance.C6 (Create)

12. Computer Vision

By the end of this course the student will be able to:

- Understand and explain the fundamental concepts, techniques, and algorithms used in Computer Vision, including image processing, feature extraction, and object recognition.C2 (Understand)
- Implement computer vision algorithms and techniques using modern libraries and frameworks, such as OpenCV, TensorFlow, or PyTorch, to solve real-world image processing problems.C3 (Apply)
- Evaluate and assess the performance of computer vision models using metrics like accuracy, precision, recall, Intersection over Union (IoU), and mean average precision (mAP).C4 (Analyze)

13. Information Retrieval

By the end of this course the student will be able to:

- Explain the core principles and algorithms used in Information Retrieval systems, including indexing, ranking, and query processing.C2 (Understand)
- Apply IR techniques to build a simple search engine or retrieval system, using tools and programming languages like Python, Elasticsearch, or Apache Lucene.C3 (Apply)
- Evaluate the performance of Information Retrieval systems through metrics such as precision, recall, F1-score, and Mean Average Precision (MAP) C4 (Analyze)

14. Generative AI

By the end of this course the student will be able to:

- Describe the foundational concepts, models, and architectures of Generative AI, including GANs, VAEs, and large language models.C2 (Understand)

- Implement and fine-tune generative models using modern AI frameworks such as TensorFlow or PyTorch. C3 (Apply)
- Analyze the ethical, societal, and practical implications of deploying Generative AI in real-world applications. C4 (Analyze)

15. Data Mining

By the end of this course the student will be able to:

- Explain fundamentals of data mining concepts, techniques, and algorithms.C2 (Understand)
- Apply data mining algorithms to real-world datasets using appropriate tools and techniques.C3 (Apply)
- Evaluate and interpret the performance of data mining models for informed decision-making.C5 (Evaluate)

3. ARTIFICIAL INTELLIGENCE ELECTIVES:

1. Programming for AI

By the end of this course the student will be able to:

- Write efficient and well-documented code.
- Utilize version control (Git) to manage and collaborate on AI-related code repositories.
- Implement basic Machine Learning models using standard ML frameworks

2. Machine Learning

By the end of this course the student will be able to:

- Differentiate between various ML paradigms and select appropriate models for different tasks.
- Implement, train, and evaluate classical ML models (e.g., SVM, Decision Trees, K-Means Clustering).
- Analyse model performance using appropriate metrics.

3. Deep Learning

By the end of this course the student will be able to:

- Explain the architectures of deep learning models
- Implement appropriate deep learning models for real world problems.
- Analyze and compare different deep learning models

4. Knowledge representation and Reasoning

By the end of this course the student will be able to:

- Explain the fundamental concepts, models and techniques of knowledge representation and reasoning.
- Apply logical formalism, ontologies and inference mechanisms
- Use appropriate tools and methods to implement reasoning systems and evaluate their effectiveness.

5. Generative AI

By the end of this course the student will be able to:

- Explain the fundamental principles, models and ethical considerations of generative AI
- Design and implement generative AI solutions using modern tools, frameworks and datasets for real world applications.
- Evaluate the performance, ethical risks and societal impact of large scale generative models.

6. Computer Vision

By the end of this course the student will be able to:

- Understand the fundamental principles and techniques of computer vision and image processing
- Apply concepts of computer vision for solving real world problems
- Evaluate computer vision methods for accuracy, efficiency and applicability in real world scenarios.

7. Natural Language Processing

By the end of this course the student will be able to:

- Understand fundamental NLP tasks and feature extraction techniques
- Apply appropriate NLP techniques and tools to preprocess, analyze and extract meaningful information from textual data
- Design and evaluate NLP systems with respect to accuracy, usability and ethical implications in real world contexts.

8. Data Mining

By the end of this course the student will be able to:

- Understand the fundamental concepts, techniques and applications of data mining
- Apply appropriate data mining tools and techniques for knowledge and pattern discovery
- Evaluate and interpret patterns, trends and insights derived from the data mining processes for decision making.

9. Reinforcement Learning

By the end of this course the student will be able to:

- Understand the core components of Markov decision processes and Reinforcement Learning
- Implement reinforcement learning algorithms to model and solve dynamic environments
- Evaluate reinforcement learning approaches in terms of performance scalability and applicability to real world domains

10. Optimization Techniques

By the end of this course the student will be able to:

- Explain the theoretical basis of optimization problems.
- Formulate and solve optimization problems using analytical and computational methods.
- Implement and evaluate most effective optimization tools and techniques for real world problems.

11. Stochastic Processes

By the end of this course the student will be able to:

- Explain probability theory and stochastic models to characterize randomness and uncertainty in computational systems
- Apply probabilistic techniques and stochastic methods to analyze and model real world data and processes
- Evaluate the suitability and limitations of stochastic models in engineering and computer science applications

12. Agentic AI

By the end of this course the student will be able to:

- Demonstrate understanding of agentic AI concepts and design autonomous AI agents capable of reasoning, planning, and tool-use for real-world tasks.
- Implement agentic AI systems using modern frameworks with integration of APIs, databases, and external tools.
- Evaluate agentic AI systems with special focus on ethical, societal, and professional implications of deploying autonomous AI agents, and propose responsible design strategies.

13. Speech Processing

By the end of this course the student will be able to:

- Explain fundamental concepts of speech production, perception, and signal processing, and apply them to analyze and model speech signals.
- Implement and evaluate speech processing techniques using modern programming tools and frameworks.
- Evaluate effectiveness of speech processing techniques along with ethical and societal implications of speech technologies such as privacy, surveillance, and accessibility.

14. Machine learning Operations (MLOps)

By the end of this course the student will be able to:

- Explain the principles of MLOps and apply them to design end-to-end pipelines for deploying, monitoring, and maintaining machine learning models.
- Implement MLOps workflows using modern tools in real-world deployment scenarios.
- Evaluate the performance of the implementations and assess ethical, security, sustainability issues in model deployment and lifecycle management.

15. Evolutionary Computing & Swarm Intelligence

By the end of this course the student will be able to:

- Explain the principles of evolutionary algorithms and swarm intelligence, and apply them to formulate solutions for optimization and search problems.
- Implement and analyze evolutionary and swarm-based algorithms using modern programming tools and frameworks.
- Evaluate the performance of evolutionary and swarm-based algorithms and assess their ethical and societal implications.

4. CYBER SECURITY ELECTIVES:

1. Information Assurance

By the end of this course the student will be able to:

- Define principles of confidentiality, integrity, and availability in information systems. (BT1 – Remember) (PLO1)
- Apply assurance frameworks to safeguard organizational data. (BT3 – Apply) (PLO2)
- Evaluate information assurance policies for effectiveness and compliance. (BT5 – Evaluate) (PLO4)

2. Network Security

By the end of this course the student will be able to:

- Explain common network threats and vulnerabilities. (BT2 – Understand) (PLO1)
- Configure basic network security mechanisms such as firewalls and VPNs. (BT3 – Apply) (PLO5)

- Design and develop network defense solutions (BT3-Apply)(PLO3)
- Evaluate existing security architectures to identify strengths and weaknesses. (BT5 – Evaluate) (PLO1)

3. Secure Software Design and Development

By the end of this course the student will be able to:

- Explain secure coding principles and design patterns. (BT2 – Understand) (PLO1)
- Apply secure software development practices across the SDLC. (BT3 – Apply) (PLO2)
- Critically evaluate the threats and vulnerabilities associated with information, computing and management systems and use required security principles. (BT5 – Evaluate) (PLO2)
- Design robust software systems resistant to common exploits. (BT6 – Create) (PLO4)

4. Digital Forensics and Incident Response

By the end of this course the student will be able to:

- Explain principles of evidence collection and chain of custody. (BT2 – Understand) (PLO1)
- Apply forensic tools to analyze compromised systems. (BT3 – Apply) (PLO5)
- Design incident response strategies for organizational resilience. (BT6 – Create) (PLO4)
- Write forensic reports based on findings of investigations performed by forensic tools. (BT3-Apply) (PLO7)

5. Applied Cryptography

By the end of this course the student will be able to:

- Understand the basics of cryptography, cryptanalysis, number theory, algebra, and complexity theory. (BT2 – Understand) (PLO1)
- Understand cryptographic protocols and algorithms in different application areas (BT2 – Understand)(PLO1)
- Evaluate security and performance of cryptographic algorithms and propose improvements (BT5 – Evaluate)(PLO4)
- Implement fundamental public key and symmetric key cryptographic algorithms. (BT3 – Apply) (PLO5)

6. Vulnerability Assessment and Penetration Testing

By the end of this course the student will be able to:

- Explain the software vulnerabilities, network vulnerabilities, malware analysis and mitigation strategies. (BT2 – Understand) (PLO1)
- Describe program analysis techniques for code analysis. (BT2 – Understand) (PLO1)
- Perform penetration testing on network infrastructures to find network vulnerabilities using various tools and techniques (BT3 – Apply) (PLO5)
- Apply reverse engineering techniques for code analysis, vulnerability assessment and mitigation. (BT3 – Apply) (PLO5)

7. Ethics in Cyber Security

By the end of this course the student will be able to:

- Understand professional ethics requirements and relevant laws and values in all activities. (BT1 – Understand) (PLO1)

- Act in compliance with laws, organizational values, and stakeholder interests. (BT3 – Apply) (PLO5)
- Ensure confidentiality, integrity, and availability of sensitive information. (BT3 – Apply)(PLO5)
- Apply safe cybersecurity practices. (BT3 – Apply)(PLO5)

8. Cloud Security

By the end of this course the student will be able to:

- Describe security challenges in cloud environments. (BT2 – Understand) (PLO1)
- Apply identity, access, and encryption controls in cloud systems. (BT3 – Apply)(PLO5)
- Evaluate cloud service providers' compliance with security standards. (BT5 – Evaluate)(PLO4)

9. Embedded Systems Security

By the end of this course the student will be able to:

- Explain embedded system security challenges. (BT2 – Understand) (PLO1)
- Apply secure coding and firmware protection techniques. (BT3 – Apply)(PLO5)
- Analyze embedded vulnerabilities in IoT and CPS. (BT4 – Analyze) (PLO3)

10. Cyber Threat Intelligence

By the end of this course the student will be able to:

- Describe the role of CTI in proactive defense. (BT2 – Understand) (PLO1)
- Apply CTI frameworks to collect and analyze threat data. (BT3 – Apply)(PLO5)
- Evaluate CTI effectiveness in preventing targeted attacks. (BT5 – Evaluate) (PLO4)

11. Artificial Intelligence for Cyber Security

By the end of this course the student will be able to:

- Explain core cybersecurity concepts and the role of ML/DL in addressing modern threats. (BT2-Understand) (PLO1)
- Analyze attack vectors and assess ML/DL techniques for detection and prevention. (BT4 – Analyze) (PLO3)
- Critique traditional vs. ML/DL approaches in terms of accuracy, adaptability, and ethics. (BT5 – Evaluate) (PLO4)

12. IoT Security

By the end of this course the student will be able to:

- Identify unique security challenges in IoT ecosystems. (BT1 – Remember) (PLO1)
- Apply authentication and encryption techniques in IoT devices. (BT3 – Apply)(PLO5)
- Evaluate IoT security frameworks for scalability and resilience. (BT5 – Evaluate) (PLO4)

13. Open-Source Intelligence Techniques

By the end of this course the student will be able to:

- Explain OSINT principles, scope, and related ethical and legal considerations. (BT2-Understand) (PLO1)
- Analyze tools, methods, and frameworks for gathering and validating intelligence from diverse open sources. (BT4 – Analyze) (PLO3)
- Evaluate OSINT applications across domains such as law enforcement, journalism, finance, and cybersecurity, noting risks and best practices. (BT5 – Evaluate) (PLO4)

- Apply OSINT techniques, including SOCMINT and dark web exploration, to conduct safe and effective investigations. (BT3 – Apply) (PLO5)

14. Usable Security and Privacy

By the end of this course the student will be able to:

- Explain foundational principles of usable security and privacy, including laws, PETs, and HCI methods. (BT2- Understand) (PLO1)
- Analyze usability challenges and evaluate research, tools, and design strategies in privacy and security mechanisms. (BT5- Evaluate) (PLO3)
- Apply user-centered design and HCI methods to assess and improve usability of privacy and security solutions. (BT3- Apply) (PLO5)
- Develop and present a prototype that integrates usability principles with privacy and security by design. (BT6 – Create) (PLO4)

15. Malware Analysis and Reverse Engineering

By the end of this course the student will be able to:

- Describe types of malware and infection techniques. (BT2 – Understand) (PLO1)
- Apply reverse engineering methods to analyze malicious code. (BT3 – Apply) (PLO5)
- Design countermeasures based on malware behavior analysis. (BT6 – Create) (PLO4)

16. Security Orchestration, Automation, and Response

By the end of this course the student will be able to:

- Describe SOAR platforms and their functionalities. (BT2 – Understand) (PLO1)
- Implement automated playbooks for incident response. (BT3 – Apply) (PLO5)
- Design SOAR solutions to improve organizational security posture. (BT6 – Create) (PLO4)

17. Cyber Security Policy and Governance

By the end of this course the student will be able to:

- Explain key concepts, frameworks, and regulatory requirements in cybersecurity policy and governance. (BT2 – Understand) (PLO1)
- Analyze national and international cybersecurity policies, standards, and legal frameworks for their implications on organizations. (BT4 – Analyze) (PLO3)
- Evaluate governance models, risk management strategies, and compliance mechanisms to ensure organizational resilience. ((BT5- Evaluate) (PLO4)
- Develop a policy or governance framework addressing security, privacy, and ethical considerations for a given organizational context. (BT6 – Create) (PLO4)

18. Security Analytics and DevSecOps

By the end of this course the student will be able to:

- Explain core concepts, tools, and processes of Security Analytics and DevSecOps in modern software lifecycles. (BT2 – Understand) (PLO1)
- Analyze how DevSecOps practices enhance threat detection, vulnerability management, and compliance in CI/CD pipelines. (BT4 – Analyze) (PLO3)
- Evaluate automated security tools and techniques for monitoring, incident detection, and secure delivery. (BT5- Evaluate) (PLO4)
- Apply SIEM, log analysis, and automated testing to develop a project that integrates DevSecOps into a secure CI/CD pipeline. ((BT3 – Apply) (PLO5)

19. Quantum Security

By the end of this course the student will be able to:

- Explain the fundamental concepts of quantum cryptography and post-quantum cryptography. (BT2 – Understand) (PLO1)
- Use simulation tools (such as Qiskit, Cirq, or similar) to model and analyze basic quantum key distribution protocols (e.g., BB84) and compare these with classical encryption methods. (BT3 – Apply) (PLO5)
- Evaluate quantum and post-quantum cryptographic approaches in protecting information through case studies and simulations. (BT3-Analyze) (PLO3)

20. Blockchain and Web 3 Security

By the end of this course the student will be able to:

- Explain blockchain security principles and consensus mechanisms. (BT2 – Understand) (PLO1)
- Apply smart contract auditing and vulnerability analysis. (BT3 – Apply) (PLO5)
- Design secure decentralized applications ensuring trust and privacy. (BT6 – Create) (PLO4)

21. Wireless and Mobile Security

By the end of this course the student will be able to:

- Describe key concepts, protocols, and vulnerabilities in wireless and mobile networks (BT2 – Understand) (PLO1)
- Simulate and analyze common wireless/mobile attacks and apply corresponding defense mechanisms using security tools. (BT3 – Apply) (PLO5)
- Evaluate the effectiveness of wireless encryption standards and mobile security frameworks in mitigating real-world threats through case studies and simulation experiments. (BT5- Evaluate) (PLO3)

5. INFORMATION TECHNOLOGY ELECTIVES:**1. Web Technologies**

By the end of this course the student will be able to:

- Demonstrate understanding of web architecture and protocols.
- Design and implement responsive web pages using HTML, CSS, and JavaScript.
- Use modern web development tools and frameworks effectively.
- Work collaboratively to develop and test web applications.
- Communicate design decisions through documentation and presentations.

2. Cyber Security

By the end of this course the student will be able to:

- Explain core concepts of cybersecurity and threat models.
- Analyze vulnerabilities and propose mitigation strategies.
- Apply security tools to detect and prevent cyber threats.
- Assess ethical and legal implications of cybersecurity practices.
- Recognize the importance of continuous learning in evolving security landscapes.

3. DB Administration & Management

By the end of this course the student will be able to:

- Describe database architecture and administrative roles.
- Configure and manage database systems for performance and reliability.
- Use database tools for backup, recovery, and monitoring.

- Collaborate in teams to manage enterprise-level databases.
 - Document database policies and procedures effectively.
- 4. System & Network Administration**
By the end of this course the student will be able to:
- Explain principles of system and network administration.
 - Configure and troubleshoot network services and operating systems.
 - Use administrative tools to monitor and secure systems.
 - Demonstrate professionalism in managing shared computing resources.
 - Work effectively in teams to maintain IT infrastructure.
- 5. Information Technology Infrastructure**
By the end of this course the student will be able to:
- Identify components of IT infrastructure and their roles.
 - Design basic infrastructure models for organizational needs.
 - Apply tools for infrastructure monitoring and support.
 - Assess societal and organizational impacts of infrastructure decisions.
 - Engage in self-directed learning to stay updated with infrastructure trends.
- 6. Network Security**
By the end of this course the student will be able to:
- Describe network security principles and protocols.
 - Analyze and mitigate network-based threats.
 - Implement firewalls, VPNs, and intrusion detection systems.
 - Evaluate legal and ethical issues in network security.
 - Communicate security policies and incident reports.
- 7. Data Communication**
By the end of this course the student will be able to:
- Explain data transmission methods and communication protocols.
 - Analyze performance of communication systems.
 - Use simulation tools to model data communication networks.
 - Collaborate in teams to design communication solutions.
 - Present technical findings in written and oral formats.
- 8. Wireless and Mobile Networks**
By the end of this course the student will be able to:
- Describe wireless technologies and mobile communication standards.
 - Analyze challenges in wireless network design and deployment.
 - Apply tools to configure and test wireless networks.
 - Assess health and safety concerns in mobile network environments.
 - Demonstrate effective teamwork in wireless network projects.
- 9. Cloud Infrastructure and Services**
By the end of this course the student will be able to:
- Explain cloud service models and infrastructure components.
 - Deploy and manage virtualized cloud environments.
 - Use cloud management tools for provisioning and monitoring.
 - Evaluate legal and compliance issues in cloud computing.

- Engage in lifelong learning to adapt to cloud innovations.

10. Email Systems and Server Management

By the end of this course the student will be able to:

- Describe email protocols and server configurations.
- Install and manage email servers securely.
- Use administrative tools for email system monitoring.
- Communicate system policies and user guidelines effectively.
- Demonstrate ethical responsibility in managing communication systems.

11. Ethical Hacking and Penetration Testing

By the end of this course the student will be able to:

- Explain ethical hacking methodologies and tools.
- Conduct penetration tests to identify vulnerabilities.
- Use modern tools for exploit development and reporting.
- Demonstrate ethical conduct in simulated attack scenarios.
- Reflect on the need for continuous skill development in cybersecurity.

12. Security Policies and Compliance

By the end of this course the student will be able to:

- Explain the role of security policies in organizational governance.
- Analyze compliance frameworks and legal standards.
- Assess societal and cultural impacts of security regulations.
- Demonstrate ethical reasoning in policy development.
- Communicate policy guidelines and compliance reports.

13. IT Project Management

By the end of this course the student will be able to:

- Describe phases and methodologies of IT project management.
- Apply planning and scheduling tools to manage IT projects.
- Work effectively in multidisciplinary project teams.
- Communicate project plans and progress reports.
- Reflect on professional responsibilities in project execution.

14. Systems Analysis and Design

By the end of this course the student will be able to:

- Explain system development life cycle and modeling techniques.
- Analyze user requirements and propose system solutions.
- Design system components with consideration for societal needs.
- Document system specifications and design artifacts.
- Demonstrate teamwork in system design projects.

15. Technology Lifecycle Planning

By the end of this course the student will be able to:

- Describe stages of technology lifecycle and asset management.
- Evaluate strategies for technology upgrade and replacement.
- Assess organizational and environmental impacts of lifecycle decisions.
- Communicate lifecycle plans and documentation effectively.
- Recognize the need for continuous learning in technology evolution.

6. COMPUTER ENGINEERING ELECTIVES:

1. Linear Circuit Analysis

By the end of this course the student will be able to:

- Apply Kirchhoff's laws, mesh, and nodal analysis to solve DC and AC circuits.
- Analyze electrical circuits using Thevenin, Norton, and Superposition theorems.
- Evaluate transient and steady-state responses of RL, RC, and RLC circuits.
- Employ differential equations to model and solve first- and second-order circuits.

2. Electrical Network Analysis

By the end of this course the student will be able to:

- Analyze transient and steady-state responses of RLC circuits under AC and DC excitation.
- Examine resonance, Q-factor, and analog filter behavior in AC circuits.
- Apply phasor techniques, star-delta transformations, and three-phase power analysis.
- Use Laplace transforms and two-port network models for circuit analysis.

3. Electronic Devices and Circuits

By the end of this course the student will be able to:

- Explain the operation of diodes, BJTs, and FETs, and their basic applications.
- Design and analyze rectifier, clamping, clipping, and biasing circuits.
- Model and analyze small-signal and power amplifiers using hybrid and parameter methods.
- Apply operational amplifiers and transistor circuits in analog signal processing.

4. Signals and Systems

By the end of this course the student will be able to:

- Explain fundamental concepts of signals and linear time-invariant systems.
- Apply Fourier, Laplace, and Z-transforms for signal analysis.
- Analyze system behavior using convolution and frequency response methods.
- Use MATLAB tools to model, simulate, and interpret continuous and discrete-time signals.

5. Parallel and Distributing Computing

By the end of this course the student will be able to:

- Learn about parallel and distributed computers.
- Write portable programs for parallel or distributed architectures using Message Passing Interface (MPI) Library
- Analyze complex problems with shared memory programming with openMP.

6. Parallel Computer Architecture

By the end of this course the student will be able to:

- Analyze and compare architectural features of parallel systems (e.g. interconnection networks, memory consistency, cache coherence).
- Design and evaluate parallel algorithms and synchronization strategies for multicore and many core systems.
- Model performance, speedup, scalability, and overheads in parallel architectures.
- Apply programming techniques and tools (e.g. threads, message passing) to realize parallel solutions on real systems.

7. Digital System Design

By the end of this course the student will be able to:

- Develop digital systems using Verilog HDL and industry-standard FPGA tools.
- Implement combinational and sequential designs such as adders, multipliers, and state machines.
- Apply FPGA architectures and design flow for synthesis, placement, and routing.
- Evaluate performance trade-offs of digital building blocks in system-level design

8. Computer Interfacing

By the end of this course the student will be able to:

- Explain microcontroller architecture, instruction sets, and memory organization.
- Apply programming techniques to interface digital and analog peripherals.
- Analyze and implement embedded applications including PWM, interrupts, and scheduling.
- Design, simulate, and test embedded systems using assembly and high-level languages.

9. Control Engineering

By the end of this course the student will be able to:

- Explain modeling techniques for electrical, mechanical, and biological control systems.
- Analyze stability and performance of systems using time and frequency domain methods.
- Design controllers using root locus, Bode plots, and state-space techniques.
- Apply MATLAB tools to simulate and design modern control systems.

10. HCI & Computer Graphics

By the end of this course the student will be able to:

- Explain key principles, models, and theories of human-computer interaction.
- Analyze and identify usability issues in user interfaces through heuristic evaluation, cognitive walkthroughs, and user studies.
- Design interactive user interfaces applying usability, accessibility, and visual design principles.
- Describe fundamental graphics concepts such as rendering pipelines, transformation, clipping, and projection.
- Implement 2D and 3D rendering algorithms including shading, lighting, rasterization, and texturing.

11. Digital Signal Processing

By the end of this course the student will be able to:

- Explain fundamental principles of discrete-time signals and systems.
- Apply Fourier, Z-transform, and FFT methods for digital signal analysis.
- Design and analyze IIR and FIR filters for engineering applications.
- Use MATLAB to simulate, analyze, and interpret DSP problems.

12. Embedded Systems

By the end of this course the student will be able to:

- Explain microcontroller architecture, instruction sets, and memory organization.
- Apply programming techniques to interface digital and analog peripherals.
- Analyze and implement embedded applications including PWM, interrupts, and scheduling.

- Design, simulate, and test embedded systems using assembly and high-level languages.

13. Artificial Neural Networks and Deep Learning

By the end of this course the student will be able to:

- Explain structure, function, and types of deep neural networks.
- Apply deep learning techniques such as CNNs, RNNs, and transfer learning for AI applications.
- Analyze and optimize neural networks for performance using forward and backward propagation.
- Demonstrate applications of deep reinforcement learning using modern tools.

14. Digital Image Processing

By the end of this course the student will be able to:

- Explain fundamental concepts of digital image representation, sampling, and quantization.
- Apply spatial and frequency-domain techniques for image enhancement and restoration.
- Implement algorithms for image segmentation, feature extraction, and classification.
- Use modern tools to design and evaluate image processing applications.

15. Microprocessor and Microcontroller based Design

By the end of this course the student will be able to:

- Explain the architecture, instruction sets, and interfacing techniques of microprocessors and microcontrollers.
- Develop assembly and high-level language programs for microcontrollers to implement control and embedded applications.
- Design and integrate hardware and software components for interfacing peripherals such as sensors, actuators, and communication modules.
- Evaluate the performance and functionality of microprocessor/microcontroller-based systems through simulation and prototyping tools.

7. COMPUTER GAMES DEVELOPMENT ELECTIVES:

1. Game Design Principles

By the end of this course the student will be able to:

- Explain fundamental concepts of game design, including ideation, gameplay mechanics, documentation (e.g., GDD), iteration, and teamwork in the development process.
- Apply Unreal Engine 5 tools to import assets, build environments, configure lighting and sound, and manage projects using source control.
- Design and prototype playable levels using level design theory, documentation, greyboxing, and environment-building techniques in Unreal Engine.
- Develop interactive gameplay elements using Blueprint Visual Scripting by implementing basic logic, connecting nodes, and creating interactive objects (e.g., doors).

2. Graphic Design & Illustration Arts

By the end of this course the student will be able to:

- Explain foundational concepts of visual communication, color theory, composition, and illustration principles.
- Use digital and traditional tools to create illustrations, concept art, and visual design elements.
- Design and develop visual assets based on themes, briefs, or character/environment requirements.
- Present and critique artwork with attention to clarity, creativity, and visual communication.

3. Game Programming

By the end of this course the student will be able to:

- Explain fundamental programming concepts used in game development, including object-oriented principles, scripting, events, and game loops.
- Implement gameplay mechanics such as player movement, collisions, physics interactions, or UI elements using a game engine and relevant programming language.
- Develop and integrate scripts to control in-game behaviors and interactions using tools such as C++, C#, or Blueprint scripting.
- Test, debug, and refine gameplay features to improve functionality and player experience.

4. Game Engine Architecture

By the end of this course the student will be able to:

- Explain the architectural principles and subsystems of modern game engines, including rendering, physics, input, scripting, and asset management.
- Analyze and compare the structures of existing commercial or open-source game engines to understand how design choices support different game genres.
- Design and implement selected engine components or subsystems or develop a functional game using an existing engine framework.

5. 3D Design and Modelling

By the end of this course the student will be able to:

- Explain principles of 3D modeling, geometry, topology, UV layouts, and texturing for both visualization and game engines.
- Create 3D assets (organic or hard-surface) using professional tools such as Blender, Maya, or 3ds Max.
- Optimize and prepare 3D models for different applications, including real-time engines and rendered outputs.
- Collaborate or participate in critique and review sessions to refine and improve 3D models based on technical and aesthetic feedback.

6. 3D Animation and Visual Effects Production

By the end of this course the student will be able to:

- Explain the principles of 3D animation, rigging, motion, and compositing for visual effects.
- Create animations using tools such as Blender, Maya, or Unreal Engine.
- Produce and composite visual effects or hybrid media using tools such as After Effects or Premiere Pro.
- Evaluate the quality, performance, and realism of animated or hybrid sequences for refinement.

7. UI/UX Design and Development

By the end of this course the student will be able to:

- Explain principles of usability, interaction design, and visual hierarchy in game UI/UX.
- Design wireframes, mockups, or prototypes for in-game menus, HUDs, and interactive flows.
- Implement UI/UX designs using tools like Figma, Adobe XD, or in-engine systems.
- Evaluate and improve UI/UX designs based on usability testing, accessibility, and player feedback.

8. Storyboarding and Narrative Development

By the end of this course the student will be able to:

- Explain principles of visual storytelling, narrative structure, and character development for games.
- Create visual storyboards for gameplay sequences, cinematic scenes, or comic-style narratives.
- Develop narrative content including dialogues, character interactions, and story arcs.

9. Virtual and Augmented Reality for Games

By the end of this course the student will be able to:

- Explain principles of VR/AR design, including immersion, spatial interaction, and hardware constraints.
- Develop VR/AR game prototypes using tools like Unreal Engine, Unity, and relevant SDKs.
- Implement interactive elements, navigation, and UI/UX tailored for immersive VR/AR experiences.
- Evaluate VR/AR prototypes for usability, performance, and player engagement, integrating feedback for improvement.

10. Simulation and Digital Twins in Games

By the end of this course the student will be able to:

- Explain the principles of simulation and digital twin technology and their applications in games.
- Design and implement digital twin models or simulation scenarios using game engines or simulation frameworks.
- Integrate interactive elements, physics, and real-time behavior into simulations or digital twins.
- Analyze simulation and digital twin performance, accuracy, and user interaction for refinement.

11. Web3 Gaming

By the end of this course the student will be able to:

- Explain the fundamentals of Web3 gaming, including blockchain, NFTs, smart contracts, and decentralized assets.
- Design game concepts that incorporate player ownership, token-based economies, or decentralized mechanics.
- Implement basic Web3 features such as wallets, tokens, or NFTs using SDKs or smart contracts in game engines.

- Analyze ethical, security, legal, and sustainability implications related to Web3-based gaming ecosystems.

12. Sound Design for Games

By the end of this course the student will be able to:

- Explain sound design principles, workflows, terminology, and the narrative role of audio in interactive and non-linear media.
- Create and edit game-ready audio assets including Foley, ambience, effects, and dialogue using DAWs and recording tools.
- Implement audio using game engines and middleware (e.g., FMOD, Wwise), incorporating spatial audio, triggers, and real-time parameters.
- Design adaptive or interactive audio systems that respond dynamically to gameplay, environments, or player actions.
- Evaluate and improve audio quality, performance, and user experience through testing, reflection, iteration, and feedback integration.

13. LiveOps and Monetization for Games

By the end of this course the student will be able to:

- Explain key concepts of LiveOps, player retention, monetization models, and in-game economies.
- Design monetization features or LiveOps strategies such as updates, events, or cosmetic systems.
- Use data analytics or performance metrics to optimize retention, engagement, or monetization features.
- Evaluate ethical, legal, and user-experience implications of monetization strategies and LiveOps decisions.

14. Game Design Research Methods

By the end of this course the student will be able to:

- Explain foundational research concepts, ethical considerations, and the role of research in game design and development.
- Design and apply appropriate research methods such as surveys, interviews, observations, or playtesting to investigate game-related questions.
- Collect, analyze, and interpret qualitative or quantitative data to generate insights for design or evaluation.
- Communicate research findings through written reports, presentations, or documentation suitable for academic or industry contexts.

15. Game Testing

By the end of this course the student will be able to:

- Explain the fundamentals of game testing, including QA workflows, testing types, documentation standards, and certification requirements.
- Perform functional, usability, and playtesting using structured test plans, bug reporting, and feedback documentation.
- Use QA tools and testing techniques to identify, track, and analyze bugs or performance issues across platforms.
- Collaborate with developers and communicate findings through clear reports, feedback sessions, and test documentation.

8. MULTIMEDIA AND ANIMATION ELECTIVES:

1. Fundamentals of Drawing

By the end of this course the student will be able to:

- Explain core concepts of line, form, perspective, proportion, shading, and composition in drawing.
- Create sketches and illustrations using traditional or digital drawing techniques.
- Apply observation and visual analysis to develop accurate representations of objects, figures, and environments.
- Critique and refine drawings based on feedback, aesthetics, and technical improvement.

2. Graphic Design & Illustration Arts

By the end of this course the student will be able to:

- Explain foundational concepts of visual communication, color theory, composition, and illustration principles.
- Use digital and traditional tools to create illustrations, concept art, and visual design elements.
- Design and develop visual assets based on themes, briefs, or character/environment requirements.
- Present and critique artwork with attention to clarity, creativity, and visual communication.

3. Animation Programming

By the end of this course the student will be able to:

- Explain how scripting and automation support animation workflows in software such as Blender.
- Use Python (or similar languages) to create, control, or modify animation sequences and rigs.
- Implement programmatic or AI/computer-vision–based control of animated elements, characters, or cameras.
- Test and refine scripted animations to ensure accuracy, performance, and visual quality.

4. 2D & 3D Animation Principles

By the end of this course the student will be able to:

- Explain core animation principles such as timing, squash and stretch, anticipation, staging, and motion arcs in both 2D and 3D contexts.
- Create animated sequences using digital tools by applying keyframing, interpolation, and posing techniques.
- Apply body mechanics, facial animation, and movement physics to characters and objects.
- Evaluate and refine animations based on realism, appeal, and fluidity of motion.

5. 3D Design and Modelling

By the end of this course the student will be able to:

- Explain principles of 3D modeling, geometry, topology, UV layouts, and texturing for both visualization and game engines.
- Create 3D assets (organic or hard-surface) using professional tools such as Blender, Maya, or 3ds Max.

- Optimize and prepare 3D models for different applications, including real-time engines and rendered outputs.
- Collaborate or participate in critique and review sessions to refine and improve 3D models based on technical and aesthetic feedback.

6. 3D Animation and Visual Effects Production

By the end of this course the student will be able to:

- Explain the principles of 3D animation, rigging, motion, and compositing for visual effects.
- Create animations using tools such as Blender, Maya, or Unreal Engine.
- Produce and composite visual effects or hybrid media using tools such as After Effects or Premiere Pro.
- Evaluate the quality, performance, and realism of animated or hybrid sequences for refinement.

7. Character Design & Rigging

By the end of this course the student will be able to:

- Explain the principles of character anatomy, proportions, silhouette, and personality development for animation.
- Design character concepts, turnarounds, or model sheets for 2D or 3D production.
- Create functional character rigs using bones, controllers, and deformation systems in software such as Blender or Maya.
- Test and refine rigs to ensure proper movement, flexibility, and compatibility with animation workflows.

8. UI/UX Design and Development

By the end of this course the student will be able to:

- Explain principles of usability, interaction design, and visual hierarchy in game UI/UX.
- Design wireframes, mockups, or prototypes for in-game menus, HUDs, and interactive flows.
- Implement UI/UX designs using tools like Figma, Adobe XD, or in-engine systems.
- Evaluate and improve UI/UX designs based on usability testing, accessibility, and player feedback.

9. Storyboarding and Narrative Development

By the end of this course the student will be able to:

- Explain principles of visual storytelling, narrative structure, and character development for games.
- Create visual storyboards for gameplay sequences, cinematic scenes, or comic-style narratives.
- Develop narrative content including dialogues, character interactions, and story arcs.

10. Virtual Production & Immersive Media

By the end of this course the student will be able to:

- Explain the principles of AR, VR, MR, and real-time virtual production workflows.
- Design immersive environments or interactive scenes using engines such as Unreal or Unity.

- Integrate virtual cameras, motion tracking, or mixed-reality elements into production pipelines.
- Test and optimize immersive media experiences for usability, performance, and engagement.

11. Digital Film Production & Cinematography

By the end of this course the student will be able to:

- Explain fundamental concepts of cinematography, camera operation, framing, lighting, and shot composition.
- Plan and produce digital film content through scripting, storyboarding, and production scheduling.
- Operate digital cameras, lighting setups, and related equipment to capture high-quality footage.
- Edit and refine filmed content using post-production tools to achieve narrative clarity and visual style.

12. Sound Design & Foley Production

By the end of this course the student will be able to:

- Explain the principles of sound design, Foley, audio recording, and their role in storytelling and visual media.
- Record and create sound effects, ambience, dialogue, and Foley using microphones, studio setups, or field equipment.
- Edit and mix audio assets using digital audio workstations (DAWs) such as Audition, Pro Tools, or Reaper.
- Integrate and synchronize sound elements with visual media to enhance realism, emotion, and audience engagement.

13. Motion Graphics & Title Design

By the end of this course the student will be able to:

- Explain core principles of motion graphics, typography, visual hierarchy, and kinetic design for media.
- Create animated titles, lower thirds, and graphic sequences using tools such as After Effects or Blender.
- Apply compositing, transitions, and timing techniques to design visually engaging motion assets.
- Integrate motion graphics into video or multimedia projects with attention to branding, clarity, and audience impact.

14. Digital Media Marketing & Monetization

By the end of this course the student will be able to:

- Explain key concepts of digital media marketing, audience engagement, branding, and content strategy.
- Design marketing plans or campaigns using platforms such as social media, streaming services, or digital ads.
- Apply monetization models such as subscriptions, sponsorships, ads, merchandise, or platform-based revenue systems.
- Analyze performance metrics and user feedback to optimize marketing and monetization strategies.

15. Media Testing, Quality & Post-Production

By the end of this course the student will be able to:

- Explain the principles of media testing, quality control, post-production workflows, and industry standards.
- Perform testing and review of audio, video, animation, or interactive content to identify technical or aesthetic issues.
- Edit, correct, and enhance media assets using post-production tools for color grading, sound mixing, and visual refinement.
- Document findings and collaborate with production teams to ensure final outputs meet quality and delivery requirements.

9. ROBOTICS ELECTIVES:

1. Introduction to Robotics

By the end of this course the student will be able to:

- Explain fundamental concepts, history, and applications of robotics across industries.
- Understand robot kinematics, sensors, actuators, and control systems for robotic motion and interaction.
- Learn about robot programming solutions using platforms such as ROS, Python, or C++.
- Demonstrate practical skills in robot assembly, programming, and testing through hands-on projects.

2. Electric Circuits

By the end of this course the student will be able to:

- Apply Ohm's and Kirchhoff's laws to analyze DC and AC circuits.
- Solve problems using Thevenin's theorem, nodal and mesh analysis, and superposition principles.
- Analyze transient and steady-state responses of RC, RL, and RLC circuits.
- Perform laboratory experiments to verify circuit theorems and characteristics using simulations and hardware.

3. Basics of Electronics and Components

By the end of this course the student will be able to:

- Explain fundamental principles of semiconductors, diodes, and transistors.
- Analyze and design rectifiers, clippers, clampers, and voltage regulators.
- Apply circuit analysis techniques to study characteristics of BJTs and MOSFETs.
- Perform experiments to study the behavior and applications of electronic components.

4. Instrumentation and Measurement

By the end of this course the student will be able to:

- Explain principles of precision measurement and instrumentation techniques.
- Operate and analyze data from measurement devices including oscilloscopes, meters, and sensors.
- Apply knowledge of signal conditioning and data acquisition in engineering applications.
- Perform experiments to measure electrical and non-electrical quantities using modern instrumentation tools.

5. Signals and Systems

By the end of this course the student will be able to:

- Explain fundamental concepts of signals and linear time-invariant systems.

- Apply Fourier, Laplace, and Z-transforms for signal analysis.
- Analyze system behavior using convolution and frequency response methods.
- Use MATLAB tools to model, simulate, and interpret continuous and discrete-time signals.

6. Digital Signal Processing

By the end of this course the student will be able to:

- Explain fundamental principles of discrete-time signals and systems.
- Apply Fourier, Z-transform, and FFT methods for digital signal analysis.
- Design and analyze IIR and FIR filters for engineering applications.
- Use MATLAB to simulate, analyze, and interpret DSP problems.

7. Linear Control Systems

By the end of this course the student will be able to:

- Explain modeling techniques for electrical, mechanical, and biological control systems.
- Analyze stability and performance of systems using time and frequency domain methods.
- Design controllers using root locus, Bode plots, and state-space techniques.
- Apply MATLAB tools to simulate and design modern control systems.

8. Robot Mechanics – I (Statics)

By the end of this course the student will be able to:

- Apply static and dynamic force analysis for engineering problems in robotics.
- Analyze equilibrium, center of gravity, and friction in mechanical systems.
- Apply principles of work, energy, and momentum to robotic mechanisms.
- Solve problems involving planar kinematics and kinetics of rigid bodies.

9. Robot Mechanics – II (Dynamics)

By the end of this course the student will be able to:

- Explain classification, components, and control concepts of robotic systems.
- Apply mathematical tools including transformation matrices and DH convention for robot kinematics.
- Analyze forward and inverse kinematics for robotic manipulators.
- Evaluate ethical, societal, and collaborative aspects of robotics applications.

10. Actuators

By the end of this course the student will be able to:

- Explain principles and mathematical modeling of DC and synchronous motors.
- Analyze operating behavior and control methods for electrical drives.
- Apply power electronics fundamentals for motor control applications.
- Design control structures for speed, torque, and position regulation of actuators.

11. Introduction to Embedded Systems

By the end of this course the student will be able to:

- Explain microcontroller architecture, instruction sets, and memory organization.
- Apply programming techniques to interface digital and analog peripherals.
- Analyze and implement embedded applications including PWM, interrupts, and scheduling.

- Design, simulate, and test embedded systems using assembly and high-level languages.

12. Machine Learning

By the end of this course the student will be able to:

- Explain fundamental concepts of supervised, unsupervised, and reinforcement learning.
- Apply algorithms including regression, SVM, PCA, and clustering for real-world data analysis.
- Evaluate learning algorithms with respect to performance, bias, and variance tradeoffs.
- Develop and test machine learning solutions using programming tools.

13. 3D Machine Vision

By the end of this course the student will be able to:

- Explain principles of 3D vision including stereo, structured light, and motion-based reconstruction.
- Apply computer vision algorithms for feature extraction, matching, and object tracking.
- Analyze pose estimation and 3D reconstruction using mathematical and computational tools.
- Develop practical vision-based solutions using OpenCV or similar platforms.

14. Localization and Mapping

By the end of this course the student will be able to:

- Explain localization techniques using GPS, IMU, and odometry in robotics.
- Apply Kalman and particle filters for state estimation in mobile robots.
- Analyze SLAM algorithms for mapping and navigation in real-world environments.
- Evaluate perception techniques for indoor and outdoor robot navigation.

15. Aerial Drones

By the end of this course the student will be able to:

- Explain aerodynamic principles, flight control, and power requirements of drones.
- Analyze components of drone platforms including sensors, drives, and communication systems.
- Apply visual odometry and control technologies in UAV applications.
- Evaluate legal and ethical aspects of drone deployment in different contexts.

16. Navigation and Mobile Platforms

By the end of this course the student will be able to:

- Explain locomotion types, kinematics, and steering mechanisms of mobile robots.
- Apply path planning and obstacle avoidance algorithms for navigation tasks.
- Analyze architectures for mobile robot navigation across land, water, and underwater platforms.
- Evaluate real-world case studies of navigation in autonomous systems.

17. Speech Recognition and Synthesis

By the end of this course the student will be able to:

- Explain principles of speech signal processing and synthesis models.
- Apply machine learning techniques for classification and keyword spotting.

- Develop real-time speech recognition and synthesis applications using Python.
- Evaluate performance of speech-based systems on hardware platforms.

18. Human Robot Interaction

By the end of this course the student will be able to:

- Explain principles of multimodal human-robot interfaces and social HRI design.
- Analyze factors influencing acceptance and societal impact of service robots.
- Apply communication and interaction techniques for gesture, speech, and emotion recognition.
- Evaluate case studies of human-robot collaboration in care, education, and service domains.

10. HUMAN COMPUTER INTERACTION ELECTIVES:

1. Human Computer Interaction

By the end of this course the student will be able to:

- Explain fundamental principles and theories of human-computer interaction and their impact on interface design.
- Apply user-centered design methodologies to develop and evaluate interactive systems.
- Conduct usability testing and analyze feedback to improve system effectiveness and user experience.
- Incorporate accessibility and inclusive design practices to create accessible interfaces for diverse users.
- Demonstrate the ability to design, prototype, and present human-centered interactive systems.

2. User-Centered Design and Prototyping

By the end of this course the student will be able to:

- Explain the principles and methodologies of user-centered design and their application in developing interactive systems.
- Create prototypes and mockups for user interfaces using appropriate tools and techniques.
- Evaluate prototypes based on user feedback and usability principles for iterative improvement.

3. Design Thinking and Innovation

By the end of this course the student will be able to:

- Apply design thinking frameworks to identify user needs and generate innovative HCI solutions.
- Develop creative and practical prototypes that address real-world user problems.
- Critically assess the role of empathy and ideation in the design process.

4. Usability Engineering

By the end of this course the student will be able to:

- Analyze usability requirements and incorporate them into the design of interactive systems.
- Conduct usability testing and interpret results to improve interface effectiveness.
- Apply standards and heuristics to evaluate interface usability.

5. Rapid Application Design

By the end of this course the student will be able to:

- Utilize rapid application development tools and techniques to quickly prototype HCI solutions.
- Implement iterative design cycles to refine interface features based on user feedback.
- Balance speed and quality in developing user interfaces

6. Cognitive Psychology for Interface Design

By the end of this course the student will be able to:

- Understand cognitive processes relevant to human-computer interaction, such as perception, memory, and attention.
- Apply cognitive models to design interfaces that align with human mental models.
- Assess the cognitive load of interface designs and optimize for user efficiency.

7. Accessibility and Inclusive Design

By the end of this course the student will be able to:

- Identify barriers faced by users with disabilities and design accessible interfaces.
- Apply guidelines and standards (e.g., WCAG) to evaluate and improve accessibility.
- Promote inclusive design practices to accommodate diverse user populations.

8. Social Computing and Online Communities

By the end of this course the student will be able to:

- Analyze the role of social computing platforms and their impact on user interaction.
- Design systems that facilitate collaboration, communication, and community building.
- Address privacy, security, and ethical considerations in online social environments.

9. Visual Design and Information Architecture

By the end of this course the student will be able to:

- Apply principles of visual design to create aesthetically appealing and effective interfaces.
- Organize information logically through information architecture techniques.
- Use visual hierarchy and layout strategies to enhance user comprehension and engagement.

10. HCI in Business and Enterprise Systems

By the end of this course the student will be able to:

- Understand the unique HCI challenges in enterprise applications.
- Design scalable, efficient, and user-friendly business systems.
- Evaluate enterprise interfaces for usability and productivity enhancements.

11. HCI in Healthcare Systems

By the end of this course the student will be able to:

- Identify specific requirements and constraints of healthcare interfaces.
- Develop HCI solutions that improve patient safety, data accuracy, and workflow efficiency.
- Ensure compliance with healthcare standards and regulations.

12. HCI in Education and Learning Technologies

By the end of this course the student will be able to:

- Design interactive educational tools that promote effective learning.
- Incorporate multimedia and adaptive interfaces in learning environments.
- Assess the usability and pedagogical effectiveness of educational systems.

13. Multimedia Application Design

By the end of this course the student will be able to:

- Develop multimedia interfaces integrating text, graphics, video, and audio.
- Ensure consistency, accessibility, and usability in multimedia applications.
- Evaluate multimedia content for user engagement and effectiveness.

14. Game Design and Interactive Environments

By the end of this course the student will be able to:

- Apply HCI principles to design engaging and intuitive game interfaces.
- Develop interactive environments that enhance user immersion and experience.
- Analyze user feedback to refine game mechanics and interface elements.

15. Conversational Interfaces and Intelligent Agents

By the end of this course the student will be able to:

- Design natural language interfaces and conversational agents.
- Implement dialogue management and user intent recognition.
- Evaluate the usability and effectiveness of conversational systems.

16. HCI and Cognitive Models in AI Systems

By the end of this course the student will be able to:

- Integrate cognitive models into the design of AI-driven interfaces.
- Analyze how AI systems can adapt to user cognitive states.
- Evaluate HCI design for AI systems based on cognitive principles.

17. Capstone Project in Human-Centered Design

By the end of this course the student will be able to:

- Apply comprehensive HCI principles and methods to develop a complete user-centered system.
- Conduct user research, prototyping, testing, and iterative refinement.
- Present and defend design solutions based on usability and user experience evaluations.

11. INTERNET OF THINGS (IoT) ELECTIVES:

1. Microcontroller Architecture and System Design

By the end of this course the student will be able to:

- Explain the architecture and instruction sets of modern microcontrollers. (Understand)
- Apply system design principles to integrate peripherals and interfaces. (Apply)
- Evaluate microcontroller-based designs for performance and scalability. (Evaluate)

2. Embedded Programming

By the end of this course the student will be able to:

- Write structured programs for embedded devices using Python. (Apply)
- Implement interrupt handling and memory management in embedded applications. (Analyze)
- Develop optimized embedded applications for real-time environments. (Create)

3. Sensors, Transducers, and Interfacing

By the end of this course the student will be able to:

- Explain the principles and characteristics of sensors and transducers. (Understand)
- Apply interfacing techniques for analog and digital sensors. (Apply)
- Evaluate sensor selection for accuracy, precision, and reliability. (Evaluate)

4. Real-Time Operating Systems

By the end of this course the student will be able to:

- Explain the role and features of real-time operating systems (RTOS). (Understand)
- Apply multitasking and scheduling concepts using RTOS primitives. (Apply)
- Design and evaluate real-time applications with deterministic performance. (Create)

5. IoT Communication and Networks

By the end of this course the student will be able to:

- Explain the role and features of real-time operating systems (RTOS). (Understand)
- Apply multitasking and scheduling concepts using RTOS primitives. (Apply)
- Design and evaluate real-time applications with deterministic performance. (Create)

6. IoT Application Layer Protocol

By the end of this course the student will be able to:

- Explain the role of messaging protocols in IoT. (Understand)
- Implement device-to-cloud and device-to-device messaging using IoT protocols. (Apply)
- Analyze protocol suitability for various IoT application scenarios. (Analyze)

7. Edge AI

By the end of this course the student will be able to:

- Describe the role of AI inference at the edge. (Understand)
- Implement AI-driven decision-making in edge computing devices. (Apply)
- Evaluate trade-offs between cloud AI and edge AI solutions. (Evaluate)

8. Machine Learning

By the end of this course the student will be able to:

- Explain deep learning architectures, including CNNs, RNNs, and transformers. (Understand)
- Implement deep learning models for IoT datasets. (Apply)
- Evaluate accuracy, latency, and resource utilization of deployed models. (Evaluate)

9. IoT Cloud

By the end of this course the student will be able to:

- Explain cloud platforms supporting IoT applications. (Understand)
- Implement IoT device integration with cloud services. (Apply)
- Design scalable IoT-cloud architectures ensuring security and reliability. (Create)

10. IIOT

By the end of this course the student will be able to:

- Explain the key concepts and technologies driving Industry 4.0. (Understand)
- Apply IoT and cyber-physical system concepts in smart factory contexts. (Apply)
- Evaluate the impact of Industry 4.0 technologies on productivity and sustainability. (Evaluate)

11. Generative AI

By the end of this course the student will be able to:

- Explain the principles of generative AI models. (Understand)
- Implement generative models for IoT-related data synthesis. (Apply)
- Design use-cases leveraging generative AI for digital twin and IoT ecosystems. (Create)

12. IoT Data Analytics and Visualization

By the end of this course the student will be able to:

- Explain data analytics pipelines for IoT applications. (Understand)
- Apply visualization techniques to represent IoT data insights. (Apply)
- Evaluate the effectiveness of analytics-driven decision-making in IoT systems. (Evaluate)

13. IoT Security

By the end of this course the student will be able to:

- Identify and describe the variety of IoT systems architectures, essential components and challenges specific to IoT systems. (Understand)
- Interpret information privacy and data protection requirements in regard to IoT products design. (Analyze)
- Apply appropriate security mechanisms for IoT. (Evaluate)

14. Wireless & 5G for IoT

By the end of this course the student will be able to:

- Explain the fundamental concepts of wireless communication technologies and 5G architecture in the context of IoT. (Understand)
- Analyze the suitability of different wireless and 5G technologies for IoT applications based on performance metrics such as latency, throughput, reliability, and energy efficiency. (Analyze)
- Design and evaluate IoT solutions leveraging wireless and 5G technologies to address real-world challenges in domains like smart cities, healthcare, and industrial IoT. (Evaluate)

15. Emerging Applications of IoT

By the end of this course the student will be able to:

- Explain the fundamental concepts, architectures, and protocols of IoT, and critically evaluate their applicability in emerging application domains. (Understand)
- Analyze real-world IoT use cases to identify challenges and propose innovative solutions leveraging emerging technologies. (Analyze)
- Design and present IoT-based solutions for selected emerging applications, considering aspects of scalability, interoperability, and security. (Design and evaluate)

12. NETWORK INFRASTRUCTURE AND CLOUD COMPUTING ELECTIVES:**1. Software Development and IT Operations (DevOps)**

By the end of this course the student will be able to:

- Explain DevOps principles, culture, and lifecycle.
- Apply CI/CD pipelines for automated software delivery.
- Evaluate performance and reliability of DevOps workflows.

2. Development, Security, and Operations (DevSecOps)

By the end of this course the student will be able to:

- Identify the role of security integration within DevOps workflows.
- Implement automated security scans in CI/CD pipelines.
- Design secure DevOps pipelines balancing agility and compliance.

3. Machine Learning Operations (MLOps)

By the end of this course the student will be able to:

- Explain concepts of MLOps and model lifecycle management.
- Apply automation for data pipelines, training, and deployment.
- Evaluate reproducibility, scalability, and governance in ML systems.

4. Cloud Design and Implementation

By the end of this course the student will be able to:

- Describe architectural models for cloud deployment.
- Implement cloud infrastructure services for computing, storage, and networking.
- Design optimized cloud architectures addressing performance and cost efficiency.

5. Network Management

By the end of this course the student will be able to:

- Explain standard tools and frameworks for managing network infrastructure.
- Apply monitoring techniques to track network performance and faults.
- Analyze collected metrics to diagnose and resolve management issues.

6. Internetworking

By the end of this course the student will be able to:

- Identify the role of routers, switches, and gateways in internetwork communication.
- Configure internetworking devices for data transmission across heterogeneous networks.
- Evaluate performance trade-offs in different internetworking topologies.

7. Machine Learning

By the end of this course the student will be able to:

- Explain the fundamental concepts, techniques, and applications of machine learning.
- Implement machine learning models using appropriate tools, libraries, and frameworks.
- Analyze the performance of machine learning models using standard evaluation metrics.
- Design and develop end-to-end machine learning solutions for real-world datasets.

8. Network Security

By the end of this course the student will be able to:

- Explain key concepts of cryptography, authentication, and network threats.
- Implement access control lists, firewalls, and intrusion prevention measures.
- Evaluate the effectiveness of layered defense mechanisms against attacks.

9. Software Defined Networks (SDN)

By the end of this course the student will be able to:

- Describe the architecture and principles of SDN.

- Configure SDN controllers to manage dynamic traffic flows.
- Design SDN solutions to optimize scalability and security in enterprise environments.

10. Wireless and Mobile Networks

By the end of this course the student will be able to:

- Explain wireless standards, spectrum usage, and mobility protocols.
- Implement configurations for WLANs and mobile networks.
- Analyze challenges of mobility, handoff, and interference in wireless environments.

11. Enterprise Network Infrastructure Engineering Design

By the end of this course the student will be able to:

- Apply systematic design methodologies for enterprise network infrastructures.
- Evaluate scalability, availability, and resilience requirements in enterprise networks.
- Create a comprehensive enterprise network design with redundancy and security features.

12. Network Programmability and Automation

By the end of this course the student will be able to:

- Demonstrate use of APIs for programmable network configurations.
- Develop scripts and playbooks for automated deployment of network services.
- Evaluate trade-offs between manual and automated approaches in network management.

13. Next-Generation Networks and Edge Computing

By the end of this course the student will be able to:

- Explain the concepts of edge computing and 5G architectures.
- Analyze performance benefits and latency reductions enabled by edge computing.
- Design use-case scenarios integrating 5G with edge infrastructure.

14. Cloud-Native Application Development

By the end of this course the student will be able to:

- Explain cloud-native principles, including containerization and microservices.
- Develop scalable applications leveraging cloud-native frameworks.
- Evaluate resilience and scalability features of cloud-native applications.

15. Infrastructure as Code (IaC)

By the end of this course the student will be able to:

- Explain IaC concepts and tools (Terraform, Ansible, etc.).
- Apply declarative scripts to provision and manage infrastructure.
- Evaluate benefits and risks of infrastructure automation approaches.

16. Microservices Architecture and Docker Containers

By the end of this course the student will be able to:

- Explain the principles of microservices and containerization.
- Deploy containerized applications using Docker.
- Design microservice architectures for scalability and maintainability.

17. Network Automation and Orchestration

By the end of this course the student will be able to:

- Demonstrate automation workflows for network provisioning.

- Implement orchestration tools to coordinate multi-domain services
- Design end-to-end automated orchestration solutions.

18. Cloud Load Balancing

By the end of this course the student will be able to:

- Describe principles and algorithms of load balancing in the cloud.
- Configure load balancing services for distributed applications.
- Evaluate performance of different load balancing strategies.

19. Network Performance

By the end of this course the student will be able to:

- Explain key concepts, metrics, and models related to network performance (e.g., throughput, latency, jitter, QoS).
- Apply analytical and simulation techniques to evaluate the performance of computer networks.
- Analyze the impact of protocols, architectures, and traffic patterns on network efficiency.
- Design and propose solutions to enhance performance in real-world networking scenarios.

20. System Programming

By the end of this course the student will be able to:

- Explain core principles of systems programming, including memory management, concurrency, synchronization, and inter-process communication.
- Analyze the performance of systems programs, identify bottlenecks, and optimize efficiency using NUMA-aware and parallel programming strategies.
- Develop multi-threaded and distributed applications that incorporate synchronization, state machine replication, and fault tolerance.
- Integrate systems programming concepts with cloud-based machine learning services, evaluating their performance and limitations.

13. QUANTUM COMPUTING ELECTIVES:

1. Mathematical Foundations for Quantum Computing

By the end of this course the student will be able to:

- Explain key mathematical structures (vector spaces, probability theory, tensor products, group theory) relevant to quantum computing. (C2 – Understand)
- Solve mathematical problems involving matrix operations, eigenvalues/eigenvectors, and tensor algebra to model quantum systems. (C3 – Apply)
- Analyze how mathematical frameworks (e.g., Hilbert spaces, unitary operations) support the representation and evolution of quantum states. (C4 – Analyze)

2. Quantum Physics for Computer Scientists

By the end of this course the student will be able to:

- Describe fundamental quantum principles such as wave-particle duality, superposition, entanglement, and uncertainty principle. (C2 – Understand)
- Relate basic quantum physics concepts to computational models (e.g., how superposition enables parallelism in quantum computing). (C3 – Apply)
- Evaluate the limitations of classical physics in explaining phenomena addressed by quantum mechanics and their implications for computing. (C5 – Evaluate)

3. Introduction to Quantum Computing

By the end of this course the student will be able to:

- Describe concepts of qubits, superposition, entanglement, and measurement. (C2 – Understand)
- Differentiate between classical and quantum paradigms. (C4 – Analyze)
- Analyze basic use cases of quantum computing. (C4 – Analyze)

4. Quantum Circuits and Gates

By the end of this course the student will be able to:

- Explain the quantum circuit model and gate sets. (C2 – Understand)
- Design and represent simple quantum circuits. (C3 – Apply)
- Implement and test circuits on simulators. (C3 – Apply)
- Evaluate circuit complexity and performance. (C5 – Evaluate)

5. Quantum Programming and Simulation

By the end of this course the student will be able to:

- Develop quantum programs using Qiskit/Cirq/Q#. (C3 – Apply)
- Simulate quantum operations on classical/cloud platforms. (C3 – Apply)
- Debug and validate results. (C4 – Analyze)
- Compare simulation outputs with theoretical predictions. (C5 – Evaluate)

6. Quantum Algorithms

By the end of this course the student will be able to:

- Explain key quantum algorithms (Shor, Grover, Deutsch–Jozsa). (C2 – Understand)
- Implement algorithms using simulators/cloud devices. (C3 – Apply)
- Compare computational complexity of quantum vs. classical algorithms. (C4 – Analyze)
- Evaluate the feasibility of quantum algorithms for real-world problems. (C5 – Evaluate)

7. Quantum Information Processing

By the end of this course the student will be able to:

- Describe states, measurements, and entropy concepts. (C2 – Understand)
- Analyze quantum teleportation and info transfer. (C4 – Analyze)
- Relate information-theoretic principles to error handling. (C4 – Analyze)

8. Quantum Error Correction and Fault Tolerance

By the end of this course the student will be able to:

- Explain fault tolerance and error sources. (C2 – Understand)
- Implement basic error-correcting codes (Shor code, stabilizers). (C3 – Apply)
- Analyze effectiveness of error correction. (C4 – Analyze)
- Demonstrate role of error correction in scalable QC. (C5 – Evaluate)

9. Quantum Cryptography and Security

By the end of this course the student will be able to:

- Explain QKD (BB84, E91) and PQC principles. (C2 – Understand)
- Implement QKD protocols in simulations. (C3 – Apply)
- Analyze security benefits/limitations of QKD. (C4 – Analyze)
- Evaluate impact of QC on classical cryptosystems. (C5 – Evaluate)

10. Post Quantum Cryptography

By the end of this course the student will be able to:

- Describe NIST/FIPS PQC standards (FIPS 203–205). (C2 – Understand)
- Interpret compliance requirements for organizations. (C3 – Apply)
- Assess migration strategies toward PQC adoption. (C5 – Evaluate)

11. Quantum Hardware and Emerging Technologies

By the end of this course the student will be able to:

- Describe superconducting qubits, trapped ions, and photonic systems. (C2 – Understand)
- Compare hardware approaches in scalability and noise resilience. (C4 – Analyze)
- Evaluate limitations of current hardware technologies. (C5 – Evaluate)

12. Quantum Machine Learning

By the end of this course the student will be able to:

- Explain hybrid quantum-classical ML models. (C2 – Understand)
- Implement basic QML algorithms. (C3 – Apply)
- Compare QML and classical ML performance. (C4 – Analyze)
- Evaluate QML applications in optimization/classification. (C5 – Evaluate)

13. Quantum Networking

By the end of this course the student will be able to:

- Explain the core concepts of quantum networking, including entanglement distribution and teleportation. (Understand – C2)
- Apply simulation tools to model simple quantum communication channels and repeaters. (Apply – C3)
- Analyze the performance and limitations of different quantum network architectures. (Analyze – C4)

14. Quantum Internet

By the end of this course the student will be able to:

- Describe the architecture, protocols, and applications of the quantum internet. (Understand – C2)
- Simulate quantum internet protocols (e.g., QKD over multiple nodes) using software frameworks. (Apply – C3)
- Evaluate the role of the quantum internet in enabling secure communication and distributed quantum computing. (Evaluate – C5)

15. Advanced Quantum Applications

By the end of this course the student will be able to:

- Identify real-world problems solvable with QC. (C4 – Analyze)
- Design and implement a project applying QC frameworks. (C6 – Create)
- Demonstrate feasibility of QC solutions with simulations/prototypes. (C5 – Evaluate)
- Present project findings with critical evaluation. (C6 – Create)

14. HEALTH INFORMATICS ELECTIVES:**1. Introduction to Health Informatics & Public Health (3 CH — Theory)**

By the end of this course the student will be able to:

- Design components of a health information system and EHR. (C2 – Understand)

- Implement basic workflows for data collection and retrieval from EHR. (C3 – Apply)
 - Analyze system requirements and interoperability needs. (C4 – Analyze)
- 2. Statistics for Health Informatics (3 CH — Theory)**
By the end of this course the student will be able to:
- Apply basic statistical techniques to health data analysis. (C2 – Understand)
 - Interpret statistical results in healthcare contexts. (C3 – Apply)
 - Use statistical tools for evidence-based decision making. (C3 – Apply)
- 3. Natural Language Processing in Healthcare (3 CH — Theory)**
By the end of this course the student will be able to:
- Explain NLP techniques for processing clinical text. (C2 – Understand)
 - Apply NLP methods to extract information from unstructured healthcare data. (C5 – Design)
 - Design NLP pipelines for clinical decision support. (C4 – Analyze)
- 4. Health Data Modalities and Visualization (3+1 CH — Theory + Lab)**
By the end of this course the student will be able to:
- Understand basic healthcare data modalities and design effective visualizations for healthcare data. (C2 – Understand)
 - Interpret data trends and patterns for decision-making. (C3 – Apply)
 - Build dashboards for clinical and administrative reporting. (C3 – Apply)
- 5. Big Data Analytics in Healthcare (3+1 CH — Theory + Lab)**
By the end of this course the student will be able to:
- Explain principles of big data technologies in healthcare. (C3 – Apply)
 - Apply analytics techniques to large health datasets. (C3 – Apply)
 - Design predictive models for clinical decision support (C4 – Analyze)
- 6. Machine Learning for Healthcare (3+1 CH — Theory + Lab)**
By the end of this course the student will be able to:
- Explain fundamental concepts of machine and deep learning. (C2 – Understand)
 - Apply ML/DL algorithms to healthcare datasets. (C3 – Apply)
 - Analyze model performance using evaluation metrics. (C4 – Analyze)
- 7. Health Data Standards and Interoperability (3 CH — Theory)**
By the end of this course the student will be able to:
- Describe key data standards and terminologies (e.g., HL7, FHIR). (C2 – Understand)
 - Explain interoperability challenges and solutions. (C4 – Analyze)
 - Apply data exchange protocols in healthcare scenarios. (C4 – Analyze)
- 8. Information Security and Privacy (3 CH)**
By the end of this course the student will be able to:
- Explain principles of data security and patient privacy. (C2 – Understand)
 - Identify vulnerabilities in healthcare data systems. (C3 – Apply)
 - Analyze security protocols and encryption techniques. (C4 – Analyze)
- 9. Health Data Governance and Ethics (3+1 CH — Theory + Lab)**
By the end of this course the student will be able to:
- Describe principles of data governance in healthcare. (C2 – Understand)

- Analyze ethical and legal issues in handling health data. (C3 – Apply)
 - Apply governance frameworks for data stewardship (C3 – Apply)
- 10. Blockchain and Secure Health Data Exchange (3 CH — Theory)**
By the end of this course the student will be able to:
- Describe the fundamentals of blockchain technology. (C2 – Understand)
 - Analyze the applications of blockchain in secure health data exchange. (C3 – Apply)
 - Design and evaluate blockchain-based solutions for healthcare interoperability. (C5 – Evaluate)
- 11. Health Information Systems and Electronic Health Records (3 CH — Theory)**
By the end of this course the student will be able to:
- Design components of a health information system and EHR. (C2 – Understand)
 - Implement basic workflows for data collection and retrieval from EHR. (C3 – Apply)
 - Analyze system requirements and interoperability needs. (C4 – Analyze)
- 12. Project Management for Health IT (3+0 CH — Theory)**
By the end of this course the student will be able to:
- Explain project management principles in healthcare IT. (C2 – Understand)
 - Develop project plans including scope, timelines, and resources. (C3 – Apply)
 - Analyze risk management techniques in health IT projects (C4 – Analyze)
- 13. Telehealth Systems and Remote Monitoring (3 CH — Theory)**
By the end of this course the student will be able to:
- Describe components and technologies of telehealth systems. (C2 – Understand)
 - Design remote monitoring solutions for patient care. (C5 – Design)
 - Analyze implementation challenges and solutions. (C4 – Analyze)
- 14. Predictive Analytics and Decision Support Systems (3 CH — Theory)**
By the end of this course the student will be able to:
- Explain predictive analytics methods used in healthcare. (Understand – C2)
 - Apply predictive models to support clinical decision-making. (Apply – C3)
 - Design decision support tools based on predictive outcomes. (Evaluate – C5)
- 15. Genomic Data Science and Precision Medicine (Capstone, 3+1 CH — Project)**
By the end of this course the student will be able to:
- Explain the role of genomics in personalized medicine. (C4 – Analyze)
 - Analyze genomic datasets for clinical decision-making. (C6 – Create)
 - Evaluate bioinformatics tools for data interpretation. (C5 – Evaluate)
- 16. Leadership and Innovation in Health Informatics (3 CH — Theory)**
By the end of this course the student will be able to:
- Describe leadership roles in health informatics projects. (C2 – Understand)
 - Apply innovation strategies to solve healthcare IT challenges. (C3 – Apply)
 - Demonstrate teamwork and collaboration in interdisciplinary contexts. (C3 – Apply)
- 17. Data Management in Healthcare (3 CH — Theory)**
By the end of this course the student will be able to:
- Design data collection and storage solutions. (C2 – Understand)
 - Implement data quality control processes. (C5 – Design)
 - Manage healthcare databases for clinical use. (C4 – Analyze)

LIST OF RECOMMENDED CERTIFICATIONS

➤ Software Engineering Specialization Certifications:

1	Microsoft Certified: Azure Developer Associate	https://learn.microsoft.com/en-us/credentials/certifications/azure-developer/?practice-assessment-type=certification
2	AWS Certified Developer – Associate	https://aws.amazon.com/certification/certified-developer-associate/
3	Professional Scrum Master (PSM) Certification	https://www.scrum.org/assessments/professional-scrum-master-i-certification
4	Oracle Certified Professional, Java Application Developer Certificate	https://education.oracle.com/oracle-certified-professional-java-ee-7-application-developer/trackp_900
5	PCEP™ – Certified Entry-Level and Professional Python Programmer	https://pythoninstitute.org/pcep
6	Google Advanced Data Analytics Certificate	https://grow.google/certificates/advanced-data-analytics/
7	IBM Full Stack Software Developer Professional Certificate	https://www.coursera.org/professional-certificates/ibm-full-stack-cloud-developer

➤ Human Computer Interaction: Course Mapping (International Benchmarks)

- Foundations of Human-Computer Interaction**
 Stanford University – HCI Graduate Certificate
<https://online.stanford.edu/programs/human-computer-interaction-graduate-certificate>
- Cognitive Psychology for Interface Design**
 Iowa State University – Online Master’s in HCI
<https://iowastateonline.iastate.edu/programs-and-courses/analytics/human-computer-interaction-master-of-computer-human-interaction/>
- User-Centered Design and Prototyping**
 Interaction Design Foundation – HCI Foundations of UX Design
<https://www.interaction-design.org/courses/hci-foundations-of-ux-design>
- Visual Design and Information Architecture**
 Coursera – UI/UX Design Specialization (CalArts)
<https://www.coursera.org/specializations/ui-ux-design>
 edX – HCI Programs
<https://www.edx.org/learn/human-computer-interaction>
- Usability Engineering and Evaluation**
 MIT CSAIL – HCI for UX Design
<https://prolearn.mit.edu/human-computer-interaction-user-experience-design>
 Coursera – Interaction Design Specialization
<https://www.coursera.org/specializations/interaction-design>

- **HCI in Business and Enterprise Systems**
Coursera – Digital Product Management
<https://www.coursera.org/specializations/uva-darden-digital-product-management>
Stanford – HCI Graduate Certificate
<https://online.stanford.edu/programs/human-computer-interaction-graduate-certificate>
- **HCI in Education and Learning Technologies**
Coursera – Learning How to Learn
<https://www.coursera.org/learn/learning-how-to-learn>
- **Social Computing and Online Communities**
Coursera – Social Computing
<https://www.coursera.org/learn/social-computing>
Stanford – Social Computing (HCI Certificate)
<https://online.stanford.edu/programs/human-computer-interaction-graduate-certificate>
- **Game Design and Interactive Environments**
Coursera – Game Design and Development
<https://www.coursera.org/specializations/game-design>
edX – Introduction to Game Design
<https://www.edx.org/course/introduction-to-game-design>
- **HCI and Cognitive Models in AI Systems**
Georgia Tech – Cognitive Science and AI
<https://omscs.gatech.edu/specialization-human-computer-interaction>
- **Design Thinking and Innovation Studio**
Coursera – Design Thinking for Innovation
<https://www.coursera.org/learn/uva-darden-design-thinking-innovation>
edX – Design Thinking Fundamentals
<https://www.edx.org/course/design-thinking-fundamentals>

➤ **Certified Professional in Health Informatics**

- 1) **Certified Health Data Analyst**
 - **Issuing Body:** American Health Information Management Association (AHIMA) – USA
- 2) **Certified Health Informatics Systems Professional (CHISP®)**
 - **Issuing Body:** American Society of Health Informatics Managers (ASHIM) – USA
- 3) **Health Informatics Certification (HIC)**
 - **Issuing Body:** Healthcare Information and Management Systems Society (HIMSS)
- 4) **Digital Health Leader Certificate**
 - **Issuing Body:** Global Digital Health Partnership (GDHP) or Digital Health Canada
- 5) **PGD Digital Health Informatics**
 - **Issuing Body:** Health Services Academy (HSA) Islamabad, Pak.

➤ AI Specialization MOOC courses, Certifications and Recommendations

1. Foundational Core (4 Courses)

Course Name	MOOC/Online Platform Alignment (Example)
Stochastic Processes	edX: <i>Probability - The Science of Uncertainty and Data</i> (MIT) or Coursera: <i>Introduction to Probability and Data</i> (Duke University)
Discrete Mathematics for Computing	edX: <i>Discrete Mathematics and Probability</i> (Arizona State University) or Coursera: <i>Mathematical Thinking in Computer Science</i> (UCSD)
Data Structures and Algorithms	Coursera: <i>Algorithms for Dummies</i> Specialization (Princeton University) or edX: <i>Data Structures and Algorithms</i> (Various)

2. Core AI and Machine Learning (6 Courses)

Course Name	Rationale	Course Learning Outcomes (CLOs)	MOOC/Online Platform Alignment (Example)	Credit
Programming for AI	The entry point focusing on advance programming, data manipulation, and clean coding practices.	CLO 1: Write efficient and well-documented code using standard libraries. CLO 2: Utilize version control (such as Git) to manage and collaborate on AI-related code repositories. CLO 3: Implement basic AI algorithms	Coursera: <i>Python for Everybody</i> Specialization (University of Michigan) or edX: <i>Introduction to Computer Science and Programming Using Python</i> (MIT)	3
Machine Learning	Covers the main supervised and unsupervised learning paradigms.	CLO 1: Differentiate between various ML paradigms and select appropriate models for different tasks. CLO 2: Implement, train, and evaluate classical ML models (e.g., SVM, Decision Trees, K-Means Clustering). CLO 3: Analyze model performance using appropriate metrics.	Coursera: <i>Machine Learning Specialization</i> (DeepLearning.AI / Stanford) or edX: <i>Machine Learning</i> (Columbia University)	3
Deep Learning	The core of modern AI, covering core	CLO 1: Explain the architecture of foundational artificial	Coursera: <i>Deep Learning Specialization</i> (DeepLearning.AI) or	3

Course Name	Rationale	Course Learning Outcomes (CLOs)	MOOC/Online Platform Alignment (Example)	Credit
	architectures and training techniques.	neural networks. CLO 2: Implement appropriate deep learning models for real-world problems. CLO 3: Analyse and compare different deep learning models	edX: <i>Introduction to Deep Learning</i> (Various)	
Knowledge Representation & Reasoning	Essential for Explainable AI (XAI) and systems that handle logic and semantics.	CLO 1: Represent domain knowledge using formalisms like Semantic Networks, Frames, and First-Order Logic . CLO 2: Implement inference mechanisms (e.g., Resolution, Backward Chaining) to derive new knowledge. CLO 3: Design and evaluate the constraints of an intelligent agent system based on logical principles.	Udemy: <i>Knowledge Graphs and Reasoning Systems</i>	3
Optimization Techniques	Fundamental to training all Machine Learning models.	CLO 1: Explain the theoretical basis of optimization problems. CLO 2: Formulate and solve optimization problem using analytical and computational methods CLO 3: Implement and evaluate the most effective optimization tools and technique for real-world problems.	edX: <i>Convex Optimization</i> (Various) or Coursera: <i>Optimization Methods in Applied Mathematics</i> (Various)	3
Generative AI	A future-facing course on the most recent industry developments.	CLO 1: Explain the fundamental principals, models and ethical considerations of generative AI. CLO 2: Design and implement	DeepLearning.AI/Coursera: <i>Generative AI with Large Language Models</i> Specialization or edX:	3

Course Name	Rationale	Course Learning Outcomes (CLOs)	MOOC/Online Platform Alignment (Example)	Credit
		generative AI solutions using modern tools, frameworks and datasets for real world applications. CLO 3: Evaluate the performance, ethical risks, and societal impact of large-scale generative models.	<i>Practical Guide to Generative AI</i>	

3. Data, Systems, and Scalability (4 Courses)

Course Name	Rationale	Course Learning Outcomes (CLOs)	MOOC/Online Platform Alignment (Example)
Data Mining	This course teaches key data mining concepts and practical skills to analyze data and support decision-making.	CLO 1: Understand the fundamental concepts, techniques and applications of data mining. CLO 2: Apply appropriate data mining tools and techniques for knowledge and pattern discovery. CLO 3: Evaluate and interpret patterns, trends and insights from the data mining process for decision making.	Coursera: <i>Data Mining Specialization</i> (University of Illinois) or edX: <i>Data Mining</i> (Various)
Machine Learning Operations (MLOps)	This course equips learners with essential MLOps skills to develop, deploy, and maintain scalable machine learning models, ensuring reliable and efficient	CLO 1: Design an end-to-end MLOps workflow, including model versioning, CI/CD, and testing. CLO 2: Deploy a trained ML model as a scalable API endpoint (e.g., using Docker or Kubernetes). CLO 3: Monitor deployed models for performance degradation (model drift, data drift) and implement automated retraining loops.	Coursera: <i>Machine Learning Engineering for Production (MLOps) Specialization</i> (DeepLearning.AI) or edX: <i>DevOps for Machine Learning</i>

Course Name	Rationale	Course Learning Outcomes (CLOs)	MOOC/Online Platform Alignment (Example)
	production workflows.		

4. Applied and Advanced Electives (4 Courses Recommended)

Course Name	Rationale	Course Learning Outcomes (CLOs)	MOOC/Online Platform Alignment (Example)	
Natural Language Processing (NLP)	This course teaches core NLP techniques, advanced models, and application development, preparing learners for real-world text processing tasks.	CLO 1: Implement fundamental NLP tasks and feature extraction techniques (e.g., TF-IDF). CLO 2: Apply sequential models (RNNs/LSTMs) and Attention mechanisms for tasks like Named Entity Recognition. CLO 3: Design and evaluate a solution for a text-based application (e.g., sentiment analysis or text summarization).	Coursera: <i>Natural Language Processing Specialization</i> (DeepLearning.AI) or edX: <i>Natural Language Processing</i> (Columbia University)	
Computer Vision	This course equips learners with essential image processing concepts, deep learning techniques, and practical skills to solve real-	CLO 1: Explain the fundamental concepts of image processing and geometric transformations. CLO 2: Implement deep learning architectures (e.g., ResNet, YOLO) for object detection, segmentation, and image generation. CLO 3: Apply image recognition techniques to a real-world problem (e.g., surveillance, autonomous navigation).	Coursera: <i>Computer Vision Basics Specialization</i> (SUNY Buffalo) or edX: <i>Computer Vision</i> (Various)	

Course Name	Rationale	Course Learning Outcomes (CLOs)	MOOC/Online Platform Alignment (Example)	
	world vision problems.			
Reinforcement Learning (RL)	This course teaches core RL concepts, algorithms, and how to develop agents for practical decision-making in simulated environments.	CLO 1: Define the core components of an RL problem and the Markov Decision Process (MDP) framework. CLO 2: Implement and compare value-based (e.g., Q-learning, DQN) and policy-based RL algorithms. CLO 3: Design an RL agent to solve a planning or control problem in a simulated environment.	Coursera: <i>Reinforcement Learning Specialization</i> (University of Alberta) or edX: <i>Deep Reinforcement Learning</i> (Various)	

Recognized AI/Data Certifications by Category

Certification Name	Focus Area	Platform / Issuer	Mode
--- 1. Free Certifications (Proof-of-Knowledge) ---			
Microsoft Azure AI Fundamentals (AI-900) Course Material	Introductory concepts of ML and AI on Azure cloud. (Exam fee is separate, but learning path is free).	Microsoft Learn	Free
IBM Data Science Fundamentals Certificate	Core concepts of Data Science methodology and tools (Python, SQL).	Coursera / IBM	Free (If auditing the course)
AWS Cloud Quest: Cloud Practitioner (Game-Based)	Foundational understanding of cloud computing services and security.	Amazon Web Services (AWS)	Free

Certification Name	Focus Area	Platform / Issuer	Mode
Hugging Face: NLP Fundamentals	Practical skills in using the Hugging Face ecosystem for state-of-the-art NLP models.	Hugging Face	Free (Course content)
--- 2. Premium Certifications			
Professional Certificate in Data Science	Data Science with Python, Pandas, and visualization.	edX / Harvard or IBM	Fee (Cost for verified certificate only)
Machine Learning Engineer Career Path (Certificate)	Practical, project-based ML engineering skills (deployment, modeling).	Codecademy Pro	Fee (Subscription fee)
Tableau Desktop Specialist Certification	Demonstrates foundational skills in data visualization and business intelligence.	Tableau	Fee (Relatively low exam fee)
--- 3. International (Industry Standard) Certifications ---			
AWS Certified Machine Learning – Specialty	Advanced skills in designing, implementing, and deploying ML solutions on AWS.	Amazon Web Services (AWS)	
Google Cloud Professional Machine Learning Engineer	Expert-level proficiency in building and deploying ML models on Google Cloud Platform.	Google Cloud	
Microsoft Certified: Azure AI Engineer Associate (AI-102)	Focuses on using cognitive services, bot services, and knowledge mining on Azure.	Microsoft	
TensorFlow Developer Certificate	Verifies the ability to build and train basic ML models using TensorFlow 2.x and Python.	Google / TensorFlow	
Certified Analytics Professional (CAP)	Vendor-neutral certification focusing on the full analytics lifecycle.	INFORMS	

➤ **Certifications Recommended by PSEB and Pakistan Software Houses Association (P@SHA)**

S. #	Category	Technologies	Official Certifications if any	
1	Programming - Web Engineering	Javascript Fullstack (MEAN/MERN)		
		.Net Fullstack (C#/JavaScript/ Angular /React/SQL)	IBM Full Stack Software Developer Professional Certificate	
		C# .NET		
		Python (Native, Django)	IT Specialist Python Certification by Certiport (Certification Code INF-303/ITS-OD-303)	
		<i>Java + Angular / React</i>	Oracle Certified Professional, Java SE 8 Programmer	
		PHP Fullstack (JavaScript/Angular/React)		
2	Cloud Infrastructure - Tool / Certification	AWS Developer Certifications	AWS Certified Cloud Practitioner,	Solutions Architect/SysOps Administrator /Developer
		Microsoft Azure Certifications	Azure Fundamentals	Administrator/Developer Associate
3	Artificial Intelligence	Machine Learning/Deep Learning	Microsoft Certified: Azure AI Fundamentals (AI-900)	
4	Digital Forensic & Cyber Security	CEH, CHFI, CompTIA Security+, Autopsy, Kali Linux	CHFI by EC Council	
5	Automation & Software Testing	ISTQB and Selenium ((Java/C#/Python/JavaScript))	ISTQB Foundation tester and Advanced Level	
6	Mobile Development	React Native (Hybrid), Flutter (Hybrid)		
7	DevOps	Git, Jenkins, Docker, Kubernetes, Ansible	Microsoft Certified DevOps Engineer Expert (AZ-400)	
8	Database	Relational Database (MS SQL, MySQL, Oracle, DB2, Postgres)	Oracle Database PL/SQL Certified Associate	MySQL 8.0 Database Developer Oracle Certified Professional
9	UI / UX	UI / UX (Indesign, Captivate, Adobe Photoshop)	Google UX Design Professional Certificate	
10	Games & Art	Game Development (Unity / Unreal)	Unity Certified Game Development by EC Council - UCA 156	
11	Soft Skills	Business communication, Presentation, Public Speaking, Interview skills, Resume writing etc.		

12	Language	English, Spanish, French , Arabic, German, Japanese		
Advanced				
1	CRM - Tool	Salesforce Microsoft Dynamics 365	Salesforce Certified Administrator (ADM-201) , Microsoft Dynamics 365 (MB-910) ,	
2	Data Analyst Tools	Data Analyst Tools (Power BI, Tableau, Splunk, Qlik)	Microsoft Power BI Data Analyst Professional	Tableau Certified Data Analyst Training
3	Cloud Certifications	Microsoft Azure Certifications	Advanced courses of those mentioned in cloud recommendations above	
		AWS Developer Certifications	Advanced courses of those mentioned in cloud recommendations above	
4	Project Management - Certifications	Project Management Software (Asana, Trello, Jira)	Google Project Management: Professional Certificate	

FIELD EXPERIENCE / INTERNSHIP GUIDELINES – COMPUTING PROGRAMS

1. Field Experience / Internship Requirement

- Internship is **mandatory** for all students.
- It carries **3 credit hours**, which equals a **minimum of 150 working hours**.
- Students may complete their internship **at any time during the 4-year degree**.
- However, it is **recommended to do it after the 2nd year** (i.e., after completing 4 semesters).

2. Accepted Modes of Internship

Internships can be completed through any of the following modes:

- **On-site (Physical)** – Working at an organization's office or workplace.
- **Remote / Online** – Working virtually for an organization.
- **University-assigned Projects/Tasks** – Completing approved work given by the university.
- **Industry-focused Courses / Workshops / Bootcamps** offered or approved by the university.

3. Evaluation and Assessment

- The university will establish a **proper evaluation mechanism** to assess all internships.
- Assessment may include:
 - Internship reports/logbooks
 - Employer/supervisor feedback
 - Presentations or viva
 - Project deliverables

FINAL YEAR PROJECT (FYP) GUIDELINES – COMPUTING PROGRAMS

1. Credit Hours and Duration

- The Final Year Project is **mandatory**.
- It carries a **minimum of 6 credit hours**.
- It must be completed over **at least two consecutive semesters or during Last semester** as given in the scheme.

2. Alignment with SDGs

- All projects should be **aligned with one or more UN Sustainable Development Goals (SDGs)** or relevant thematic areas (e.g., health, education, environment, smart cities, innovation, etc.).

3. Scope and Team Structure

- Students may work **individually or in teams** (as per department policy), but workload should match the credit hours.
- Each student must have a **clearly defined role**.

4. Supervision

- Each project must be supervised by a **faculty member** approved by the department.
- Co-supervisors from industry are encouraged where applicable.

5. Evaluation and Assessment

Evaluation will be conducted in **multiple stages**, such as:

- Proposal submission and approval
- Progress reviews (mid evaluations)
- Final presentation/demonstration
- Report/documentation submission
- External evaluation where required

6. Deliverables

Students must submit the following upon completion:

- Final report/thesis
- Presentation slides
- Working prototype/system/model
- Poster or summary document

7. Ethical and Professional Requirements

- Projects must follow **ethical guidelines** and avoid plagiarism.
- Any use of licensed data, copyrighted material, or AI tools must be properly acknowledged.

Framework for Integration of International Certifications into BS Computer Science Program

1. Title

Integration of International Professional Certifications into BS Computer Science Program.

2. Objectives

- To recognize selected international certifications as equivalent to **one elective (3 credit hours)** in BS Computing programs.
- To ensure that recognition of certifications does not compromise academic integrity or NCEAC accreditation standards.
- To enhance employability of graduates by combining formal degree education with verifiable industry credentials.
- List of certifications recommended by PSEB – Ministry of IT & Telecom will be approved by BoS, BoF, and Academic Council and the university may add more certifications.
- Government and other funding avenues will cover the cost of certifications for the university students and affiliated colleges.

3. Scope & Applicability

- Applicable to **all HEC-recognized universities/DAIs** in Pakistan offering BS Computing programs accredited by NCEAC.
- Effective from **Fall 2026 intake**.

4. Guiding Principles

1. **Credit Recognition: 3 credit hours** per student with at least 48 hours of engagement with certification.
2. **Category Restriction:** Certification credit may only replace **one elective course**.
3. **Verification Requirement:** Student must pass a **university-administered test and/or** to validate skills before credit is awarded.
4. **Validity & Authenticity:** Certification must be valid (not expired) and verifiable with the issuing body.
5. **Standardization Across Universities:** All universities follow the same framework approved by NCRC.
6. **Approved Certifications (Initial List – 2025)**
Initial list provided by PSEB, MoITT is attached with this document. Further updates shall be provided on the website: <https://techdestination.com/>

5. Review & Continuous Improvement

- HEC will review policy every **3 years**.