University of Mumbai

Website - mu.ac.in
Email id - dr.aams@fort.mu.ac.in
aams3@mu.ac.in

Academic Authorities,
Meetings & Services (AAMS)
Room No. 128, M. G. Road, Fort,
Mumbai – 400 032.
Tel. 022-68320033

Re- accredited with A ++ Grade (CGPA 3.65) by NAAC
Category- I University Status awarded by UGC

No.2026/May/AAMS-III/C-83/42997 Date: 08th May, 2026

CIRCULAR:-

Attention of all the Principals of the Affiliated Colleges & Autonomous Colleges, all Academic Heads of University Departments, the Directors of the Recognized Institutions/Centers, Director, Centre for Distance & Online Education, Director, Dharmaveer Anand Dighe Thane Sub-Campus, Co-ordinator, School of Engineering and Applied Sciences, Kalyan Sub-Campus, Director, Chitrakar Padmabhushan Dr.Dhananjay Keer Ratnagiri Sub-Campus, Director, Sindhudurg Sub-Campus, Principal, Vishwabhushan Bharatratna Dr. B. A. Ambedkar College, Ambadave, Ratnagiri and Principal, V.V. Dalvie College, Talere, Sindhudurg is invited to this office Circular No. AAMS/ICD/2025-26/ 37 of dated 27 May, 2025 relating to the NEP UG & PG Syllabus.

They are hereby informed that the recommendations made by the **Ad-hoc Board of Studies in Computer Science** at its meeting held on 16th February, 2026 vide item No.1 and subsequently passed by the Board of Deans at its meeting held on 17th March, 2026 vide item No. 6.14 (N) have been accepted by the Academic Council at its meeting held on 25th March, 2026 **vide** item No. 6.33 (N). In accordance therewith syllabus of **B.Sc. (Computer Science) (Scheme - I) (Sem V & VI)** (NEP 2020) is introduced as per appendix with effect from the academic year 2026-27.

(The said circular is available on the University's website www.mu.ac.in).

MUMBAI – 400 032
08th May, 2026

(Dr. Prasad Karande)
REGISTRAR

To,

All the Principals of the Affiliated Colleges & Autonomous Colleges, all Academic Heads of University Departments, the Directors of the Recognized Institutions/Centers, Director, Centre for Distance & Online Education, Director, Dharmaveer Anand Dighe Thane Sub-Campus, Co-ordinator, School of Engineering and Applied Sciences, Kalyan Sub-Campus, Director, Chitrakar Padmabhushan Dr.Dhananjay Keer Ratnagiri Sub-Campus, Director, Sindhudurg Sub-Campus, Principal, Vishwabhushan Bharatratna Dr. B. A. Ambedkar College, Ambadave, Ratnagiri and Principal, V.V. Dalvie College, Talere, Sindhudurg.

AC./6.33 (N)/25/3/2026

Copy forwarded with Compliments for information to:-

1) The Chairman, Board of Deans,
2) The Dean, Faculty of Science,
3) The Chairman, **Ad-hoc Board of Studies in Board of Studies in Computer Science**
4) The Director, Board of Examinations and Evaluation,
5) The Director, Department of Students Development,
6) The Director, Department of Information & Communication Technology,
7) The Director, Centre for Distance and Online Education (CDOE)Vidyanagari,
8) The Deputy Registrar, Admission, Enrolment, Eligibility & Migration Department (AEM),

Copy forwarded for information and necessary action to :-	
1	The Deputy Registrar, (Admissions, Enrolment, Eligibility and Migration Dept)(AEM), dr@eligi.mu.ac.in
2	The Deputy Registrar, Result unit, Vidyanagari drresults@exam.mu.ac.in
3	The Deputy Registrar, Marks and Certificate Unit,. Vidyanagari dr.verification@mu.ac.in
4	The Deputy Registrar, Appointment Unit, Vidyanagari dr.appointment@exam.mu.ac.in
5	The Deputy Registrar, CAP Unit, Vidyanagari cap.exam@mu.ac.in
6	The Deputy Registrar, College Affiliations & Development Department (CAD), deputyregistrar.uni@gmail.com
7	The Deputy Registrar, PRO, Fort, (Publication Section), Pro@mu.ac.in
8	The Deputy Registrar, Executive Authorities Section (EA) eau120@fort.mu.ac.in He is requested to treat this as action taken report on the concerned resolution adopted by the Academic Council referred to the above circular.
9	The Deputy Registrar, Research Administration & Promotion Cell (RAPC), rapc@mu.ac.in
10	The Deputy Registrar, Academic Appointments & Quality Assurance (AAQA) dy.registrar.tau.fort.mu.ac.in ar.tau@fort.mu.ac.in
11	The Deputy Registrar, College Teachers Approval Unit (CTA), concolsection@gmail.com
12	The Deputy Registrars, Finance & Accounts Section, fort draccounts@fort.mu.ac.in
13	The Deputy Registrar, Election Section, Fort drelection@election.mu.ac.in
14	The Assistant Registrar, Administrative Sub-Campus Thane, thanesubcampus@mu.ac.in
15	The Assistant Registrar, School of Engg. & Applied Sciences, Kalyan, ar.seask@mu.ac.in
16	The Assistant Registrar, Ratnagiri Sub-centre, Ratnagiri, ratnagirisubcentar@gmail.com
17	The Director, Centre for Distance and Online Education (CDOE), Vidyanagari, director@idol.mu.ac.in
18	Director, Innovation, Incubation and Linkages, Dr. Sachin Laddha pinkumanno@gmail.com
19	Director, Department of Lifelong Learning and Extension (DLLE), dlleuniversityofmumbai@gmail.com

Copy for information :-	
1	P.A to Hon'ble Vice-Chancellor, vice-chancellor@mu.ac.in
2	P.A to Pro-Vice-Chancellor pvc@fort.mu.ac.in
3	P.A to Registrar, registrar@fort.mu.ac.in
4	P.A to all Deans of all Faculties
5	P.A to Finance & Account Officers, (F & A.O), camu@accounts.mu.ac.in

To,

1	The Chairman, Board of Deans pvc@fort.mu.ac.in
2	**Faculty of Humanities,** **Offg. Dean** 1. Prof.Anil Singh Dranilsingh129@gmail.com **Offg. Associate Dean** 2. Prof.Manisha Karne mkarne@economics.mu.ac.in 3. Dr.Suchitra Naik Naiksuchitra27@gmail.com
	Faculty of Commerce & Management, **Offg. Dean,** 1 Prin.Ravindra Bambardekar principal@model-college.edu.in **Offg. Associate Dean** 2. Dr.Kavita Laghate kavitalaghate@jbims.mu.ac.in 3. Prin.Kishori Bhagat kishoribhagat@rediffmail.com

	Faculty of Science & Technology **Offg. Dean** 1. Prof. Shivram Garje ssgarje@chem.mu.ac.in **Offg. Associate Dean** 2. Dr. Madhav R. Rajwade Madhavr64@gmail.com 3. Prin. Deven Shah sir.deven@gmail.com
	Faculty of Inter-Disciplinary Studies, **Offg. Dean** 1.Dr. Anil K. Singh aksingh@trcl.org.in **Offg. Associate Dean** 2.Prin.Chadrashekhar Ashok Chakradeo cachakradeo@gmail.com 3. Dr. Kunal Ingle drkunalingle@gmail.com
3	Chairman, Board of Studies,
4	The Director, Board of Examinations and Evaluation, dboee@exam.mu.ac.in
5	The Director, Board of Students Development, dsd@mu.ac.in DSW direcotr@dsw.mu.ac.in
6	The Director, Department of Information & Communication Technology, director.dict@mu.ac.in

AC – 25/03/2026
Item No. – 6.33 (N)

As Per NEP 2020

University of Mumbai

Syllabus for Major Vertical – 1, 4 & 6 (Scheme – I)		
Faulty of Science & Technology		
Board of Studies in Computer Science		
Name of the Programme – B.Sc. (Computer Science)		
U.G. Third Year Programme	**Exit Degree**	**B.Sc. (Computer Science)**
Semester		**V & VI**
From the Academic Year		**2026-27**

University of Mumbai

(As per NEP 2020)

Sr. No.	Heading	Particulars
1	**Title of program** **O: _____**	**B.Sc. (Computer Science)**
2	**Exit Degree**	**B.Sc. (Computer Science)**
3	**Scheme of Examination** **R: _____**	NEP **40% Internal** **60% External**, Semester End Examination **Individual Passing** in Internal and External Examination
4	**Standards of Passing** **R: _____**	**40%** in **each component**
5	**Credit Structure** **R: SU – 555 E (I)** **R: SU – 555 F (I)**	**Attached herewith**
6	**Semesters**	Sem V & VI
7	**Program Academic Level**	5.5
8	**Pattern**	Semester
9	**Status**	New
10	**To be implemented from Academic Year**	From Academic Year: 2026-27

Sd/-	**Sd/-**	**Sd/-**
Sign of the BOS Chairman	**Sign of the Offg. Associate Dean**	**Sign of Offg. Dean**
Dr. Jyotshna Dongardive	**Dr. Madhav R. Rajwade**	**Prof. Shivram S. Garje**
Ad-hoc BOS (Computer Science)	Faculty of Science & Technology	Faculty of Science & Technology

Preamble

1) Introduction

The Third Year of the B.Sc. (Computer Science) programme represents the culmination of undergraduate study at Academic Level 5.5 under the NEP 2020 framework. At this stage, the curriculum transitions from foundational and intermediate concepts to advanced, applied, and research-oriented domains of Computer Science. The focus is on developing depth in emerging areas such as Artificial Intelligence, Cyber & Information Security, Data Science, Cloud Computing, and Software Project Management, along with elective specializations in Information Retrieval and Linux Server Administration.

The programme is structured to integrate conceptual understanding with hands-on implementation through advanced practical courses and two structured mini-projects. Mini Project – I emphasize solution development for real-world problems using appropriate design methodologies and documentation standards, while Mini Project – II focuses on research orientation, experimentation, analysis, and scholarly reporting in established academic format. This dual approach ensures that learners are prepared for both industry deployment and academic progression.

In alignment with the vision of the University of Mumbai and the National Education Policy 2020, the Third Year curriculum promotes analytical rigor, innovation, ethical responsibility, and professional competence. It prepares graduates not only to meet contemporary industry demands but also to pursue higher studies, research, entrepreneurship, and lifelong learning in rapidly evolving technological ecosystems.

2) Aims and Objectives

Advanced Conceptual Understanding: To provide in-depth knowledge of advanced computing domains including Artificial Intelligence, Information Security, Data Science, and Cloud Computing, thereby strengthening the theoretical and applied foundations of Computer Science.

Research and Analytical Competence:
To cultivate research aptitude through literature review, methodology design, experimentation, data analysis, and structured research reporting.

Industry-Oriented Skill Development:
To enhance practical and professional skills through hands-on laboratory work, real-world project development, and exposure to current industry tools and technologies.

Project Planning and Management Skills:
To develop competencies in software project management, including planning, scheduling, risk management, quality assurance, and agile methodologies.

Ethical and Responsible Computing:
To instill ethical awareness in areas such as cybersecurity, responsible AI, data privacy, and professional conduct in computing practices.

Preparation for Higher Studies and Career Progression:

To equip learners with the intellectual maturity, technical expertise, and research orientation required for postgraduate studies, research careers, and professional roles in the IT industry.

3) Learning Outcomes

By the end of the third year (T.Y.B.Sc.), students will be able to:

- Demonstrate advanced knowledge in Artificial Intelligence, Data Science, Cloud Computing, Cyber Security, and related emerging domains.
- Design, develop, and deploy complete software solutions addressing real-world problems using appropriate architectures, tools, and technologies.
- Apply research methodology principles including literature review, problem formulation, hypothesis validation, experimentation, and result interpretation.
- Analyze datasets using statistical and machine learning techniques and interpret results using appropriate performance metrics.
- Implement secure computing practices by applying cryptographic techniques, authentication mechanisms, network security models, and system hardening principles.
- Plan, manage, and monitor software projects using structured methodologies including traditional and agile frameworks.
- Evaluate alternative technological solutions and select appropriate tools and platforms based on performance, scalability, security, and cost considerations.
- Work effectively as individuals and as team members in multidisciplinary environments, demonstrating leadership and collaborative skills.
- Prepare and present technical documentation, research papers, and project reports in standardized professional formats.
- Demonstrate ethical responsibility, professional integrity, and awareness of legal and societal implications of computing technologies.
- Pursue higher education, research, entrepreneurship, or professional employment with confidence and competence in advanced computing domains.

4) Credit Structure of the Program

B.Sc. (Computer Science)

Level	Semester	Major		Minor	OE	VSC, SEC (VSEC)	AEC, VEC, IKS	OJT, FP, CEP, CC, RP	Cum. Cr. / Sem.	Degree/ Cum. Cr.
		Mandatory	Electives							
	R: SU – 555 E (I)									
5.5	**V**	**MJ11:** Artificial Intelligence (TH) – 2 **MJ12:** Cyber & Information Security (TH) – 2 **MJP5:** Computer Science Practical 5 (PR) – 2 **MJP6:** Mini Project – I (PR) – 2 **MJ13:** IKS in Computational Systems (TH) – 2 **10**	**MJEL1:** Software Testing & Quality Assurance (TH) – 2 **OR** **MJEL2:** Wireless & Sensor Networks (TH) – 2 **MJELP1:** Software Testing & Quality Assurance Practical (PR) – 2 **OR** **MJELP2:**Wireless & Sensor Networks Practical (PR) – 2 **4**	4	-	**VSC: 2** Ethical Hacking – 2	-	FP/CEP:2	22	**UG Degree 132**
	R: SU – 555 F (I)									
	VI	**MJ14:** Data Science (TH) – 2 **MJ15:** Cloud Computing (TH) – 2 **MJ16:** Software Project Management (TH) – 2 **MJP7:** Computer Science Practical 6 (PR) – 2 **MJP8:** Mini Project – II (PR) – 2 **10**	**MJEL3:** Information Retrieval (TH) – 2 **OR** **MJEL4:** Linux Server Administration (TH) – 2 **MJELP3:** Information Retrieval Practical (PR) – 2 **OR** **MJELP4:** Linux Server Administration Practical (PR) – 2 **4**	4	-	-	-	OJT:4	22	
	Cum Cr.	**48**	**8**	**18**	**12**	**8+6**	**8+4+2**	**8+6+4**	**132**	
Exit option: Award of UG Degree in Major with 132 credits OR Continue with Major and Minor										

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project]

Under Graduate Diploma in Computer Science

Program Structure for Sem. V & VI
(NEP 2020)

Vertical No		Paper Title	Credits
Sem. V			
1.	**Mandatory**	Artificial Intelligence	**2**
		Cyber & Information Security	**2**
		Computer Science Practical 5	**2**
		Mini Project – I	**2**
		IKS in Computational Systems	**2**
	Electives *(any one)*	Software Testing & Quality Assurance	**2**
		Wireless & Sensor Networks	**2**
		Software Testing & Quality Assurance Practical	**2**
		Wireless & Sensor Networks Practical	**2**
2.	**Minor (To be selected from other discipline)**		Ethical Hacking
4.	**VSC (Practical)**	Ethical Hacking	**2**
6.	**CEP (To be selected from CEP topic list)**		**2**
		Credits	**22**
Sem. VI			
1.	**Mandatory**	Data Science	**2**
		Cloud Computing	**2**
		Software Project Management	**2**
		Computer Science Practical 6	**2**
		Mini Project – II	**2**
	Electives *(any one)*	Information Retrieval	**2**
		Linux Server Administration	**2**
		Information Retrieval Practical	**2**
		Linux Server Administration Practical	**2**

2.	**Minor** **(To be selected from other discipline)**		
6.	**VI**	**OJT**	**4**
		Credits	**22**
		Total Credits	**44**

Sem. – V

Sem. - V

Vertical – 1 Major Mandatory (2+2+2+2+2)

Syllabus
B.Sc. (Computer Science)
(Sem.- V)

Name of the Course: Artificial Intelligence

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** The Artificial Intelligence course introduces students to the foundational principles that enable machines to simulate intelligent behavior. It covers intelligent agents, search techniques, logical reasoning, probabilistic modeling, machine learning paradigms, reinforcement learning, and responsible AI. The course builds a structured understanding of how intelligent systems are designed, analyzed, and deployed. **Relevance:** This course is highly relevant in today's data-driven world where automation and intelligent decision-making are integral to almost every sector. Understanding AI bridges the gap between theoretical computer science concepts and real-world intelligent applications such as recommendation systems, predictive analytics, and autonomous systems. **Usefulness:** The course equips students with practical and analytical skills required to design AI-based solutions. It enhances problem-solving ability through search algorithms, logical inference, probabilistic reasoning, and machine learning techniques. These competencies are essential for developers, analysts, researchers, and technology professionals working with intelligent systems. **Application:** Artificial Intelligence principles are applied in domains such as healthcare diagnostics, financial forecasting, robotics, natural language processing, cybersecurity threat detection, recommendation engines, and generative AI systems. From chatbots and fraud detection systems to autonomous vehicles and smart assistants, AI forms the backbone of modern intelligent applications. **Interest:** The subject stimulates curiosity by demonstrating how

		machines can learn, reason, and make decisions. Concepts like A* search, game-playing algorithms (Minimax), Bayesian reasoning, neural networks, and reinforcement learning make the course intellectually challenging and closely connected to emerging technologies such as generative AI and deep learning. **Connection with Other Courses:** AI concepts are closely linked with Data Structures and Algorithms (search trees and graphs), Discrete Mathematics (logic and probability), Data Science (model building and evaluation), Software Engineering (AI system lifecycle), Cybersecurity (AI-driven threat detection), and Cloud Computing (AI deployment at scale). The course strengthens interdisciplinary understanding within the computer science curriculum. **Demand in the Industry:** The demand for AI and ML skills is rapidly increasing across IT services, fintech, healthcare, retail, manufacturing, and media industries. Organizations require professionals who understand intelligent systems, predictive modeling, decision automation, and ethical AI deployment, making this course strategically important for industry readiness. **Job Prospects:** Students with AI knowledge can pursue roles such as AI/ML intern, junior data analyst, AI application developer, or software engineer integrating AI solutions. With advanced study and project experience, they can progress to positions such as machine learning engineer, data scientist, AI research associate, or AI product specialist.
2	**Vertical:**	Major (Mandatory)
3	**Type:**	Theory
4	**Credits:**	2 credits
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** **CO 1.** To introduce the foundational concepts of Artificial Intelligence and intelligent agent design. **CO 2.** To develop problem-solving skills using search strategies and logical reasoning techniques.	

	CO 3. To provide understanding of probabilistic reasoning and machine learning methodologies. **CO 4.** To familiarize students with reinforcement learning and latent variable models. **CO 5.** To cultivate awareness of ethical considerations and responsible AI practices.
8	**Course Outcomes (OC):** After successful completion of this course, students would be able to - **OC 1.** Explain the principles of intelligent agents, search strategies, and logical reasoning frameworks. **OC 2.** Apply informed and uninformed search algorithms and adversarial reasoning techniques to solve AI problems. **OC 3.** Implement and evaluate basic machine learning models for classification, regression, and clustering tasks. **OC 4.** Analyze probabilistic models including Bayesian networks and hidden variable models for uncertain scenarios. **OC 5.** Assess AI systems from a responsible AI perspective, identifying bias, fairness, transparency, and generative AI risks.
9	**Modules:** **Module 1 (15 hours):**
	Foundations of AI & Intelligent Agents: What is AI? Rational agents vs human thinking, Computational agents, Agent–Environment interaction, Types of environments, Agent architectures (simple reflex, model-based, goal-based, utility-based, learning agents) **Problem Solving & Search:** Problem formulation, Uninformed search: BFS, DFS, Uniform Cost, IDS, Informed search: Greedy, A*, Heuristics: admissibility & consistency, Adversarial search: Minimax & Alpha-Beta pruning **Knowledge Representation & Logical Reasoning:** Knowledge-based agents, Propositional logic & inference, First-Order Logic, Rule-based systems, Planning basics (STRIPS concept), Fuzzy Logic & Fuzzification **Reasoning Under Uncertainty:** Probabilistic reasoning, Bayes theorem, Conditional independence, Bayesian Networks
	Module 2 (15 hours):
	Introduction of Machine Learning: Forms of learning (supervised, unsupervised, reinforcement), Parametric vs Nonparametric models, Bias–variance tradeoff, Overfitting & regularization, Gradient descent (intuitive) **Supervised Learning Models:** Classification vs Regression, k-NN, Decision Trees, Naive Bayes, SVM, Artificial Neural Networks:(single-layer & concept of deep learning, Ensemble methods & Boosting **Probabilistic & Latent Variable Models:** Statistical learning framework, Maximum Likelihood Estimation, Learning with complete data, Hidden variables, EM Algorithm, Hidden Markov Models

	Unsupervised & Reinforcement Learning: Concept of clustering, Association rule mining (Apriori concept), Reinforcement learning framework, Markov Decision Processes, Q-Learning (update rule intuition) **Responsible AI:** Ethical issues in AI systems, Bias and fairness in AI models, Transparency and explainability, Accountability and human oversight, Risks in Generative AI (hallucination, deepfakes, misuse)	
10	**Text Books** 1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, 3rd Edition, Pearson, 2010.	
11	**Reference Books** 1. Artificial Intelligence: Foundations of Computational Agents, David L Poole,Alan K. Mackworth, 2nd Edition, Cambridge University Press ,2017. 2. Artificial Intelligence, Kevin Knight and Elaine Rich, 3rd Edition, 2017 3) The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2013	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

Name of the Course: Cyber and Information Security

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** The Cyber & Information Security course introduces students to the principles and mechanisms used to protect digital information, communication systems, and computing infrastructure. It covers foundational security concepts, classical and modern cryptography, authentication mechanisms, network security protocols, intrusion detection, malware threats, and firewall architectures. **Relevance:** In an era of digital transformation, cybersecurity is critical for safeguarding data, privacy, and national infrastructure. This course is highly relevant as organizations increasingly depend on secure communication, encrypted transactions, identity management systems, and network defense mechanisms to prevent cyber threats. **Usefulness:** The course equips students with conceptual and practical understanding of encryption algorithms, authentication protocols, digital signatures, secure communication standards, and network protection mechanisms. This knowledge is essential for identifying vulnerabilities, implementing security controls, and designing secure systems in real-world environments. **Application:** Cyber and information security principles are applied in banking systems, e-commerce platforms, cloud infrastructure, government services, defense networks, and enterprise IT systems. Technologies such as AES, RSA, SSL/TLS, IPsec, Kerberos, digital signatures, intrusion detection systems, and firewalls are integral to securing modern digital ecosystems. **Interest:** The subject generates strong interest as it deals with real-world cyber-attacks, encryption techniques, malware analysis, and network defense strategies. Topics such as ethical hacking concepts, digital forensics basics, cryptographic algorithms, and DDoS countermeasures make the course both technically engaging and socially

		significant. **Connection with Other Courses:** This course is closely connected with Computer Networks (protocols and OSI model), Operating Systems (system security mechanisms), Discrete Mathematics (number theory and cryptographic foundations), Cloud Computing (secure infrastructure), and Software Engineering (secure coding practices). It strengthens interdisciplinary knowledge required for building secure computing systems. **Demand in the Industry:** There is a rapidly growing demand for cybersecurity professionals across IT services, fintech, healthcare, telecom, defense, and cloud service providers. Organizations require experts in encryption, authentication, network security, intrusion detection, and firewall management to mitigate evolving cyber threats and comply with regulatory standards. **Job Prospects:** Students can pursue roles such as cybersecurity analyst, network security engineer, information security officer, SOC analyst, penetration testing trainee, and security consultant. Advanced expertise can lead to careers in cryptography research, cloud security architecture, digital forensics, and security auditing.
2	**Vertical:**	Major (Mandatory)
3	**Type:**	Theory
4	**Credits:**	2 credits
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** **CO 1.** To introduce foundational concepts of cybersecurity, including security services, attacks, and mechanisms. **CO 2.** To develop understanding of symmetric and asymmetric cryptographic techniques and key management. **CO 3.** To provide knowledge of authentication methods, digital signatures, and secure communication protocols. **CO 4.** To familiarize students with network security technologies, intrusion detection, and malware countermeasures. **CO 5.** To enable understanding of firewall design principles and security architecture implementation.	

8	**Course Outcomes (OC):** After successful completion of this course, students would be able to - **OC 1.** Explain security architecture, attack models, and fundamental cryptographic principles. **OC 2.** Apply symmetric and public-key cryptographic techniques (AES, RSA, Diffie-Hellman) for secure communication. **OC 3.** Analyse and implement authentication mechanisms including digital signatures, hash functions, MACs, and PKI-based systems. **OC 4.** Evaluate network and web security protocols such as SSL/TLS, IPsec, Kerberos, and email security mechanisms. **OC 5.** Identify and assess intrusion techniques, malware threats, DDoS attacks, and firewall configurations for effective defence.
9	**Modules:** **Module 1 (15 hours):**
	Introduction: Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms **Classical Encryption Techniques:** Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Steganography, Block Cipher Principles, The Data Encryption Standard, The Strength of DES, AES (round details not expected), Multiple Encryption and Triple DES, Block Cipher Modes of Operation, Stream Ciphers **Public-Key Cryptography and RSA:** Principles of Public-Key Cryptosystems, The RSA Algorithm **Key Management:** Public-Key Cryptosystems, Key Management, Diffie-Hellman Key Exchange **Message Authentication and Hash Functions:** Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs, Secure Hash Algorithm, HMAC
	Module 2 (15 hours):
	Digital Signatures and Authentication: Digital Signatures, Authentication Protocols, Digital Signature Standard **Authentication Applications:** Kerberos, X.509 Authentication, Public-Key Infrastructure **Electronic Mail Security:** Pretty Good Privacy, S/MIME **IP Security:** Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management **Web Security:** Web Security Considerations, Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction **Intrusion:** Intruders, Intrusion Techniques, Intrusion Detection **Malicious Software:** Viruses and Related Threats, Virus Countermeasures, DDOS

	Firewalls: Firewall Design Principles, Types of Firewalls	
10	**Text Books** 1. Cryptography and Network Security: Principles and Practice 7th edition, William Stallings, Pearson	
11	**Reference Books** 1. Cryptography and Network Security, 2nd edition, Behrouz A Fourouzan, Debdeep Mukhopadhyay, TMH. 2. Atul Kahate, "Cryptography and Network Security", Tata McGraw-Hill.	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

IKS
(2)

Name of the Course: IKS in Computational Systems

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** The IKS in Computational Systems course explores classical Indian Knowledge Systems (IKS) through the lens of modern computational thinking. It examines how śāstra-based formalization, Pāṇini's generative grammar, Piṅgala's combinatorics, Nyāya logic, Ayurvedic classification, and Arthaśāstra cryptography embody structured, rule-based, and algorithmic principles analogous to contemporary Computer Science concepts. **Relevance:** This course is highly relevant in the context of interdisciplinary education under NEP 2020, linking traditional knowledge frameworks with modern computational theory. It demonstrates that formal systems, symbolic encoding, logical inference, and structured reasoning have deep historical roots aligned with current developments in AI, formal grammars, combinatorics, and cybersecurity. **Usefulness:** The course develops conceptual clarity in knowledge representation, rule-based reasoning, generative systems, symbolic abstraction, and inference validation. It strengthens analytical thinking by mapping classical structures such as pramāṇa theory, prastāra enumeration, and five-member syllogism to modern logic systems, parsing algorithms, and explainable AI. **Application:** The computational parallels of śāstra methodology, Pāṇinian grammar, and Nyāya logic apply to formal language theory, natural language processing, expert systems, decision trees, inference engines, and secure communication models. Students gain perspective on how structured reasoning models can inform AI system design, feature engineering, classification systems, and encryption foundations. **Interest:** The course is intellectually stimulating as it reveals algorithmic and logical sophistication embedded in ancient Indian texts. Discovering binary encoding in

		Piṅgala, generative grammar in Pāṇini, inference structures in Nyāya, and coded communication in Arthaśāstra creates a unique synthesis of philosophy and computer science. **Connection with Other Courses:** This course connects strongly with Artificial Intelligence (knowledge representation and inference), Formal Languages and Automata Theory (CFG and rewrite systems), Data Science (classification and feature mapping), Cyber Security (encryption principles), and Information Retrieval/NLP (grammar and parsing foundations). It enriches computational subjects through historical and conceptual depth. **Demand in the Industry:** Modern fields such as Artificial Intelligence, Natural Language Processing, and cybersecurity rely heavily on rule-based systems, logical inference, and structured knowledge representation. The course highlights the foundational principles behind these systems through classical knowledge frameworks. **Job Prospects:** The concepts covered support careers in AI, data science, computational linguistics, knowledge engineering, and cybersecurity. The analytical and interdisciplinary perspective gained also benefits students pursuing research and advanced studies in computer science.
2	**Vertical:**	Major (Mandatory)
3	**Type:**	Theory
4	**Credits:**	2 credits
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** **CO 1.** To introduce the concept of śāstra as a structured and formal knowledge system aligned with computational thinking. **CO 2.** To analyze algorithmic and combinatorial principles in Piṅgala's Chandaḥśāstra and their relation to binary systems and recursion. **CO 3.** To examine Pāṇini's grammatical system as a generative formal grammar and relate it to modern language theory. **CO 4.** To interpret Nyāya logic and Ayurvedic classification systems as structured inference and expert systems.	

	CO 5. To explore cryptographic principles in Arthaśāstra and connect them with modern secure communication models.
8	**Course Outcomes (OC):** After successful completion of this course, students would be able to - **OC 1.** Explain and interpret classical Indian knowledge frameworks as structured computational models. **OC 2.** Analyze Piṅgala's combinatorial methods and relate them to binary encoding, recursion, and tree structures. **OC 3.** Model and compare Pāṇini's generative grammar with modern formal grammars and parsing systems. **OC 4.** Apply Nyāya logical structures and Ayurvedic classification principles to modern inference engines and expert systems. **OC 5.** Relate ancient cryptographic techniques to contemporary symmetric encryption and cybersecurity foundations.
9	**Modules:** **Module 1 (15 hours):**
	Śāstra Methodology and Knowledge Formalization Indian knowledge traditions systematized disciplines as *śāstra* — structured bodies of rule-governed knowledge. Topics: • Concept of śāstra as a formal knowledge system • Sutra method as compressed symbolic encoding • Pramāṇa theory (Pratyakṣa, Anumāna, Āgama) • Knowledge validation and structured reasoning As Modern CS Concept: Knowledge representation, formal specification, symbolic abstraction. **Piṅgala's Chandaḥśāstra as Binary Encoding and Combinatorial Generation** Study of prosodic structures in Piṅgala's text and its algorithmic features. Topics: • Laghu–Guru representation as binary encoding • Prastāra (systematic enumeration of patterns) • Naṣṭa and Uddiṣṭa (index retrieval mechanisms) • Meru-Prastāra and combinatorial expansion As Modern CS Concept: Binary number systems, Recursive enumeration, Tree structures, Pascal triangle and combinatorics, Algorithmic generation **Pāṇini's Aṣṭādhyāyī as a Generative Formal Grammar** Examination of the structural architecture of Pāṇini's grammatical system. Topics: • Rule-based generative structure • Meta-rules and rule precedence • Context-sensitive operations

	• Conflict resolution mechanisms As Modern CS Concept: Context-Free Grammar (CFG), Rewrite systems, Automata theory, foundations, Parsing algorithms, Foundations of NLP
	Module 2 (15 hours):
	Nyāya Logic as Structured Inference and Knowledge Systems Study of classical Indian logical structures. Topics: • Five-member syllogism • Anumāna (inference) structure • Debate methodology and validation • Padārtha (categorical ontology overview) As Modern CS Concept: Propositional logic, Predicate logic, Inference engines, Explainable AI **Ayurvedic Classification as Rule-Based Expert System (5 Hours)** Examination of structured diagnostic reasoning in Ayurvedic texts. Topics: • Tridoṣa framework • Symptom–feature mapping • Multi-attribute classification • Decision principles in diagnosis As Modern CS Concept: Decision trees, Rule-based systems, Expert systems, Feature engineering, Multi-class classification **Arthaśāstra Cryptography as Secure Communication Model** Study of intelligence and secure communication practices. Topics: • Secret communication techniques • Substitution and transposition systems • Concealment and coded messaging • Information protection mechanisms As Modern CS Concept: Symmetric encryption, Cipher algorithms, Basic steganography, Secure protocol abstraction, Foundations of cybersecurity
10	**Text Books** 1. Computing Science in Ancient India, T.R.N. Rao/ Subhash Kak 2. The Mathematics of India: Concepts, Methods, Connections P. P. Divakaran, Hindustan Book Agency, 2018
11	**Reference Books** 1. Pāṇini: Background and introduction, George Cardona, Motilal Banarsidass, 1988 2. Nyaya-Vaisesika (A History of Indian Literature) B. K. Matilal, Harrassowitz, 1977 3. The Computation Meme: Explorations in Indic Computational Thinking K.

	Gopinath & Shailaja D. Sharma, Indian Institute of Science (IISc), Bengaluru, 2022 4. Indian Mathematics Engaging the World from Ancient to Modern Times G. G. Joseph, 2016	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

Name of the Course: Computer Science Practical 5

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** This practical course provides hands-on implementation experience in Artificial Intelligence and Cyber & Information Security. It enables students to translate theoretical concepts into working models, algorithms, and configurations through structured laboratory exercises. The course emphasizes experimentation, performance evaluation, comparative analysis, and real-world system implementation. **Relevance:** The course is highly relevant as it integrates two critical domains—AI and Cyber Security—through applied learning. It strengthens students' ability to implement algorithms such as search techniques, machine learning models, encryption methods, and network security mechanisms, thereby bridging the gap between conceptual understanding and practical competence. **Usefulness:** Students gain direct exposure to algorithm implementation, dataset handling, model evaluation, cryptographic programming, and security configuration. The practical approach enhances debugging skills, analytical comparison of algorithms, performance assessment, and secure system configuration, which are essential for industry-ready graduates. **Application:** The AI practicals support applications such as intelligent search systems, predictive modeling, classification tasks, neural networks, and data-driven decision systems. The cybersecurity practicals apply to secure communication, encryption systems, digital signatures, firewall configuration, intrusion detection, and secure web deployment—skills vital in enterprise and cloud environments. **Interest:** The course maintains high engagement through implementation-based learning, algorithm comparisons, model visualization, encryption experimentation, and network security setup. Working with tools like

		TensorFlow/OpenAI frameworks and configuring SSL/TLS or IDS systems makes the experience technically immersive and practically rewarding. **Connection with Other Courses:** This course complements Artificial Intelligence, Cyber & Information Security, Data Structures, Computer Networks, Operating Systems, and Cloud Computing. It reinforces algorithmic thinking, system-level understanding, and applied cryptography, thereby strengthening interdisciplinary technical competence. **Demand in the Industry:** Industry increasingly demands professionals who can not only understand AI and security concepts but also implement, evaluate, and deploy them. Skills such as machine learning model building, encryption implementation, SSL/TLS configuration, IDS setup, and firewall management are highly valued across IT services, fintech, cybersecurity firms, and cloud platforms. **Job Prospects:** Students completing this practical course are better prepared for roles such as AI/ML intern, data analyst trainee, cybersecurity analyst, SOC analyst, security operations trainee, and junior DevOps engineer. The hands-on exposure also builds a strong foundation for advanced roles in machine learning engineering, security consulting, and system administration.
2	**Vertical:**	Major (Mandatory)
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted:**	60 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** **CO 1.** To develop practical proficiency in implementing AI search algorithms and machine learning models. **CO 2.** To enable students to train, evaluate, and compare supervised learning and ensemble models. **CO 3.** To provide hands-on experience in implementing classical and modern cryptographic techniques. **CO 4.** To familiarize students with secure communication protocols, intrusion detection, and firewall configuration.	

	CO 5. To strengthen analytical and comparative skills through performance evaluation and system experimentation.
8	**Course Outcomes (OC):** After successful completion of this course, students would be able to - **OC 1.** Implement and compare AI search algorithms and supervised learning models, evaluating their performance using suitable metrics. **OC 2.** Design and train machine learning models including decision trees, neural networks, SVM, ensemble methods, and association rule mining techniques. **OC 3.** Develop and test cryptographic algorithms such as RSA, Diffie-Hellman, MACs, and digital signatures for secure communication. **OC 4.** Configure and analyze network security mechanisms including SSL/TLS, IPsec, intrusion detection systems, and firewall rules. **OC 5.** Evaluate the effectiveness, security implications, and performance characteristics of implemented AI and cybersecurity solutions.
9	**Modules:** **Module 1 (30 hours):**
	Practical based on Artificial Intelligence **Breadth First Search & Iterative Depth First Search** • Implement the Breadth First Search algorithm to solve a given problem. • Implement the Iterative Depth First Search algorithm to solve the same problem. • Compare the performance and efficiency of both algorithms. **A* Search and Recursive Best-First Search** • Implement the A* Search algorithm for solving a pathfinding problem. • Implement the Recursive Best-First Search algorithm for the same problem. • Compare the performance and effectiveness of both algorithms. **Decision Tree Learning** • Implement the Decision Tree Learning algorithm to build a decision tree for a given dataset. • Evaluate the accuracy and effectiveness of the decision tree on test data. • Visualize and interpret the generated decision tree. **Feed Forward Backpropagation Neural Network** • Implement the Feed Forward Backpropagation algorithm to train a neural network. • Use a given dataset to train the neural network for a specific task. • Evaluate the performance of the trained network on test data. **Support Vector Machines (SVM)** • Implement the SVM algorithm for binary classification. • Train an SVM model using a given dataset and optimize its parameters. • Evaluate the performance of the SVM model on test data and analyze the results. **Adaboost Ensemble Learning** • Implement the Adaboost algorithm to create an ensemble of weak classifiers.

	• Train the ensemble model on a given dataset and evaluate its performance. • Compare the results with individual weak classifiers. **Naive Bayes' Classifier** • Implement the Naive Bayes' algorithm for classification. • Train a Naive Bayes' model using a given dataset and calculate class probabilities. • Evaluate the accuracy of the model on test data and analyze the results. **K-Nearest Neighbors (K-NN)** • Implement the K-NN algorithm for classification or regression. • Apply the K-NN algorithm to a given dataset and predict the class or value for test data. • Evaluate the accuracy or error of the predictions and analyze the results. **Association Rule Mining** • Implement the Association Rule Mining algorithm (e.g., Apriori) to find frequent itemsets. • Generate association rules from the frequent itemsets and calculate their support and confidence. • Interpret and analyze the discovered association rules. **Demo of OpenAI/TensorFlow Tools** • Explore and experiment with OpenAI or TensorFlow tools and libraries. • Perform a demonstration or mini-project showcasing the capabilities of the tools. • Discuss and present the findings and potential applications.
	Module 2 (30 hours):
	Practical based on Cyber & Information Security **Implementing Substitution and Transposition Ciphers:** Design and implement algorithms to encrypt and decrypt messages using classical substitution and transposition techniques. **RSA Encryption and Decryption:** Implement the RSA algorithm for public-key encryption and decryption, and explore its properties and security considerations. **Message Authentication Codes:** Implement algorithms to generate and verify message authentication codes (MACs) for ensuring data integrity and authenticity. **Digital Signatures:** Implement digital signature algorithms such as RSA-based signatures, and verify the integrity and authenticity of digitally signed messages. **Key Exchange using Diffie-Hellman:** Implement the Diffie-Hellman key exchange algorithm to securely exchange keys between two entities over an insecure network. **IP Security (IPsec) Configuration:**

	Configure IPsec on network devices to provide secure communication and protect against unauthorized access and attacks. **Web Security with SSL/TLS:** Configure and implement secure web communication using SSL/TLS protocols, including certificate management and secure session establishment. **Intrusion Detection System:** Set up and configure an intrusion detection system (IDS) to monitor network traffic and detect potential security breaches or malicious activities. **Malware Analysis and Detection:** Analyze and identify malware samples using antivirus tools, analyze their behavior, and develop countermeasures to mitigate their impact. **Firewall Configuration and Rule-based Filtering:** Configure and test firewall rules to control network traffic, filter packets based on specified criteria, and protect network resources from unauthorized access.	
10	**Text Books** 1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, 3rd Edition, Pearson, 2010. 2. Cryptography and Network Security: Principles and Practice 7th edition, William Stallings, Pearson	
11	**Reference Books** 1. Artificial Intelligence: Foundations of Computational Agents, David L Poole,Alan K. Mackworth, 2nd Edition, Cambridge University Press ,2017. 2. Artificial Intelligence, Kevin Knight and Elaine Rich, 3rd Edition, 2017 3) The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2013 3. Cryptography and Network Security, 2nd edition, Behrouz A Fourouzan, Debdeep Mukhopadhyay, TMH. 4. Atul Kahate, "Cryptography and Network Security", Tata McGraw-Hill.	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

Name of the Course: Mini Project – I *(Real-World Application Development)*

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** Mini Project – I is a structured, development-oriented course that enables students to design and implement a complete real-world software solution. The course guides students through problem identification, requirement engineering, system modeling, architecture design, implementation, testing, and deployment. It emphasizes professional documentation, structured development practices, and industry-standard tools. **Relevance:** This course is highly relevant as it simulates an industry-style software development lifecycle from concept to deployment. Students experience real-world project constraints, stakeholder analysis, SDLC planning, and cloud deployment, preparing them for professional software development environments. **Usefulness:** The course develops practical competencies in requirement analysis, UML modeling, backend–frontend integration, database design, testing strategies, and version control using GitHub. It strengthens documentation skills, structured thinking, and collaborative development—essential capabilities for modern software engineers. **Application:** Students design solutions for real-world industry, social, or institutional problems such as service portals, management systems, automation tools, or mobile/web applications. The structured implementation and deployment process mirrors real software product development workflows used in startups, enterprises, and IT service organizations. **Interest:** The course generates strong engagement because students build a tangible, working application that solves a real problem. The opportunity to design architecture, implement features, deploy to cloud or mobile platforms, and demonstrate the final product makes the experience practically rewarding and professionally motivating. **Connection with Other Courses:** Mini Project – I integrates knowledge from Software

		Engineering, Database Management Systems, Web Development, Mobile Application Development, Cloud Computing, Cyber Security, and Data Structures. It acts as a capstone-style integrative experience that consolidates prior learning into a unified application. **Demand in the Industry:** Industry demands graduates who can independently conceptualize, design, develop, test, and deploy scalable applications. Skills such as requirement engineering, SDLC planning, UML modeling, GitHub version control, cloud deployment, and security validation are highly valued in software development, DevOps, and startup ecosystems. **Job Prospects:** Completion of this course strengthens readiness for roles such as software developer trainee, full-stack developer intern, backend/frontend developer, DevOps trainee, or application support engineer. A well-executed mini-project also enhances employability by serving as a portfolio project during placements.
2	**Vertical:**	Major (Mandatory)
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted:**	60 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** **CO 1.** To enable students to identify and analyze real-world problems and conduct feasibility studies. **CO 2.** To develop skills in requirement engineering, UML-based system modeling, and architecture design. **CO 3.** To provide hands-on experience in application development, testing, and deployment. **CO 4.** To inculcate best practices in version control, documentation, and project management. **CO 5.** To prepare students for professional software development environments through structured SDLC implementation.	
8	**Course Outcomes (OC):** After successful completion of this course, students would be able to - **OC 1.** Identify and analyse a real-world problem and prepare structured project documentation including SRS and feasibility reports.	

	OC 2. Design complete system models using UML diagrams and develop suitable frontend, backend, and database architectures. **OC 3.** Implement and integrate application components with authentication, validation, and error handling mechanisms. **OC 4.** Test and deploy applications using systematic testing approaches and version control tools such as GitHub. **OC 5.** Evaluate and present the developed solution through technical documentation, demonstration, and performance/security validation.
9	**Modules:** **Module 1 (30 hours):**
	Problem Identification, Requirement Engineering & System Design Phase **Problem Identification & Feasibility Study:** Identification of a real-world problem (industry / social / institutional), Problem justification and scope definition, Stakeholder identification, Feasibility analysis (technical, economic, operational) **Requirement Engineering:** Functional requirements specification, Non-functional requirements, Use-case analysis, Requirement prioritization, Constraints and assumptions **Software Development Life Cycle (SDLC) Planning:** Selection of SDLC model, Work Breakdown Structure (WBS), Project timeline (Gantt Chart), Resource planning **System Modeling using UML:** Use Case Diagram, Class Diagram, Sequence Diagram, Activity Diagram, ER Diagram, Deployment Diagram **System Architecture Design:** Frontend architecture, Backend architecture, Database schema design, API structure, Security considerations
	Documentation Deliverables at the End of Module 1 • Project Proposal • SRS Document • Complete UML Set • Architecture Design Document
	Module 2 (30 hours):
	Implementation, Testing, Deployment & Evaluation Phase **Application Development:** Frontend implementation, Backend implementation, Database integration, Authentication & validation, Error handling **Integration & System Testing:** Unit testing, Black-box testing, Integration testing, Test case preparation, Bug tracking **Deployment:** Cloud deployment / Local hosting, APK build, Server configuration (if applicable), Version control using GitHub (Mandatory) **Performance & Security Testing:** Basic load testing, Input validation checks, Security validation **Final Documentation:** Technical Report, User Manual, Screenshots, Source Code Documentation

	<u>Final Deliverables at the End of Module 2</u> • Working Application • GitHub Repository • Final Report • Presentation & Demonstration	
10	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

SEM V
Vertical – 1
Electives
(2+2)

Name of the Course: Software Testing & Quality Assurance

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** The Software Testing & Quality Assurance course introduces students to systematic techniques for evaluating software quality and ensuring reliability. It covers testing fundamentals, SDLC alignment, verification and validation, white-box and black-box testing strategies, defect management, software metrics, statistical quality assurance, and quality improvement tools. The course builds a structured understanding of how quality is engineered into software systems. **Relevance:** In modern software-driven environments, delivering reliable, secure, and high-quality applications is critical. This course is highly relevant as it equips students with structured testing strategies and quality assurance principles that are essential in agile, DevOps, and large-scale enterprise development environments. **Usefulness:** The course provides practical knowledge of test case design, defect tracking, validation techniques, complexity metrics, and quality standards such as ISO 9000. It enables students to systematically evaluate software, measure reliability, improve processes, and contribute effectively to quality-driven development teams. **Application:** Software testing and QA principles are applied in web applications, enterprise systems, mobile apps, cloud platforms, embedded systems, and safety-critical systems. Techniques such as equivalence partitioning, boundary value analysis, unit and integration testing, defect lifecycle management, and statistical quality control are widely used in industry. **Interest:** The course is engaging as it reveals how hidden defects are identified, analyzed, and prevented through structured methodologies. Practical exposure to white-box and black-box techniques, defect metrics, reliability measurement, and quality tools like Pareto and cause-

		effect diagrams makes the subject analytically stimulating and professionally relevant. **Connection with Other Courses:** This course complements Software Engineering, Database Management Systems, Web Development, Operating Systems, and Cyber Security. It strengthens understanding of system validation, performance evaluation, reliability analysis, and structured development practices across the computer science curriculum. **Demand in the Industry:** There is consistent demand for professionals skilled in testing strategies, automation support, quality metrics, and defect management. IT companies, product-based firms, fintech organizations, and startups require trained QA engineers and test analysts to ensure software reliability and compliance with quality standards. **Job Prospects:** Students can pursue roles such as software test engineer, QA analyst, automation testing trainee, defect analyst, or quality assurance associate. With experience, career progression may include test lead, quality manager, process improvement specialist, or reliability engineer.
2	**Vertical:**	Major (Elective)
3	**Type:**	Theory
4	**Credits:**	2 credits
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** **CO 1.** To develop understanding of software testing fundamentals and their integration within SDLC. **CO 2.** To introduce verification, validation, and systematic testing strategies. **CO 3.** To provide knowledge of white-box and black-box testing techniques and defect management processes. **CO 4.** To familiarize students with software metrics, statistical quality assurance, and reliability concepts. **CO 5.** To inculcate quality improvement practices and standards for professional software development.	
8	**Course Outcomes (OC):**	

	After successful completion of this course, students would be able to - **OC 1.** Explain and apply fundamental testing principles, SDLC-based testing approaches, and V&V mechanisms. **OC 2.** Design effective test cases using black-box, white-box, and experience-based testing techniques. **OC 3.** Analyse and manage defects using structured defect lifecycle and quality metrics. **OC 4.** Evaluate software reliability and quality using statistical methods and software metrics. **OC 5.** Apply quality assurance practices, review techniques, and quality improvement tools to enhance software processes.
9	**Modules:** **Module 1 (15 hours):**
	Software Testing Fundamentals: Nature of errors and the need for testing, Basics of software testing process, Test case design principles and techniques, Test execution, reporting, and documentation **Software Development Life Cycle (SDLC):** Overview of SDLC, phases and their relationship to testing, Role of testing in each phase, Software quality factors and their impact on testing **Definition of Quality and Quality Assurance:** Understanding quality, in software development, Distinction between Quality Assurance (QA), Quality Control (QC), Quality Management (QM), and Software Quality Assurance (SQA) **Verification and Validation (V&V):** Definition of V&V and its significance in software development, Different types of V&V mechanisms, Concepts of Software Reviews, Inspection, and Walkthrough **Software Testing Strategies:** Strategic approach to software testing, Unit Testing: purpose, techniques, and best practices; Integration Testing: approaches and challenges; Validation Testing: ensuring adherence to user requirements, System Testing: comprehensive end to end testing
	Module 2 (15 hours):
	White Box Testing and Black Box Testing: Functional/Specification, based Testing as Black Box, Black box: Equivalence Partitioning, Boundary Value Analysis, Decision Table Testing, State Transition, Testing. Structural Testing as White Box, White Box: Statement testing, Branch testing. Experience-based: Error guessing, Exploratory testing, Checklist-based testing. **Software Metrics:** Concept of software metrics and their importance, Developing and utilizing different types of metrics, Complexity metrics, and their significance in testing **Defect Management:** Definition of defects and their lifecycle, Defect management process, including defect reporting and tracking, Metrics related to defects and their utilization for process improvement

	Software Quality Assurance: Understanding quality concepts and the Quality Movement: Background issues and challenges in SQA, Activities and approaches in Software Quality Assurance, Statistical Quality Assurance and Software Reliability, Statistical process control techniques: Software reliability measurement and improvement, ISO 9000 Quality Standards **Software Reviews & Quality Improvement Techniques:** Formal Technical Reviews and their benefits; Introduction to quality, improvement methodologies, Utilizing quality costs for decision making, Introduction to quality improvement tools: Pareto Diagrams, Cause-effect Diagrams, Scatter Diagrams, Run charts	
10	**Text Books** 1. Software Engineering for Students, A Programming Approach, Douglas Bell, 4th Edition,, Pearson Education, 2005 2. Software Engineering – A Practitioners Approach, Roger S. Pressman, 7th Edition, Tata McGraw Hill	
11	**Reference Books** 1. Quality Management, Donna C. S. Summers, 5th Edition, Prentice-Hall. 2. Software Testing and Quality Assurance Theory and Practice, Kshirsagar Naik, Priyadarshi Tripathy , John Wiley & Sons, Inc. , Publication.	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

Name of the Course: Wireless & Sensor Networks

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** The Wireless & Sensor Networks course introduces students to the principles, architecture, and communication mechanisms of wireless sensor systems and mobile communication technologies. It covers sensor node design, network architecture, MAC and routing protocols, WSN operating systems, ad-hoc networking, and advanced wireless transmission fundamentals including GSM and satellite systems. **Relevance:** Wireless sensor networks form the backbone of modern IoT, smart cities, environmental monitoring, industrial automation, and mobile communication systems. This course is highly relevant as it provides foundational knowledge required to design, deploy, and optimize wireless and sensor-based communication systems in real-world environments. **Usefulness:** The course equips students with technical understanding of energy-efficient networking, routing challenges, MAC protocols, radio technologies, and wireless transmission principles. It develops analytical skills necessary for designing scalable, reliable, and secure wireless communication infrastructures. **Application:** WSN concepts are applied in smart agriculture, healthcare monitoring, industrial IoT, disaster management, military surveillance, environmental sensing, and smart grids. Knowledge of GSM, UMTS, satellite communication, and IEEE 802.15.4 standards supports applications in telecom networks, mobile communication systems, and satellite-based services. **Interest:** The subject is engaging as it explores how distributed sensor nodes communicate wirelessly under constraints such as limited power and bandwidth. Topics like ad-hoc networking, energy optimization, mobile handover, satellite routing, and spread spectrum communication make the course technically stimulating and industry-relevant.

		Connection with Other Courses: This course complements Computer Networks, Operating Systems, Embedded Systems, IoT, Cyber Security, and Cloud Computing. It integrates concepts of protocol design, system architecture, signal processing fundamentals, and secure communication in distributed environments. **Demand in the Industry:** There is growing demand for professionals skilled in IoT systems, wireless communication, telecom infrastructure, and sensor-based automation. Industries such as telecommunications, smart manufacturing, defense, automotive systems, and satellite communication require expertise in wireless network design and optimization. **Job Prospects:** Students can pursue roles such as wireless network engineer, IoT developer, telecom systems analyst, network planning engineer, embedded systems engineer, or communication systems trainee. Advanced specialization can lead to careers in telecom research, satellite communication systems, and next-generation wireless technologies.
2	**Vertical:**	Major (Elective)
3	**Type:**	Theory
4	**Credits:**	2 credits
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** **CO 1.** To introduce architectural principles and design challenges of Wireless Sensor Networks. **CO 2.** To develop understanding of WSN operating systems, MAC protocols, routing strategies, and middleware concepts. **CO 3.** To provide knowledge of wireless transmission fundamentals and radio communication techniques. **CO 4.** To familiarize students with GSM, advanced mobile systems, and satellite communication technologies. **CO 5.** To enable analysis of energy efficiency, security, and optimization in wireless and sensor networks.	
8	**Course Outcomes (OC):** After successful completion of this course, students would be able to -	

	OC 1. Explain WSN architecture, sensor node technologies, and network optimization principles. **OC 2.** Analyze and apply MAC protocols, routing strategies, and middleware concepts in WSN environments. **OC 3.** Evaluate energy efficiency, security, and design challenges in ad-hoc and sensor networks. **OC 4.** Interpret wireless transmission fundamentals including modulation, spread spectrum, and cellular communication principles. **OC 5.** Describe and assess mobile communication systems (GSM, UMTS) and satellite communication architectures.
9	**Modules:** **Module 1 (15 hours):**
	Introduction and Overview of WSNs: Basic sensor network architectural elements, Advantages and challenges of WSNs, Applications of WSNs, Sensor node technology, Sensor taxonomy, WSN operating environment, Radio technology in WSNs, Network architecture and optimization, Optimization goals, figures of merit, Design principles for WSNs, Service interfaces of WSNs, Gateway concepts **WSN Operating Systems and Ad-hoc Networks:** Overview of wireless sensor network operating systems, Examples of WSN operating systems (case/examples), Ad-hoc networks in WSNs, Characteristics and challenges of ad-hoc networks in WSNs, Energy efficiency considerations in ad-hoc networks, Security and privacy issues in ad-hoc networks **Medium Access Control (MAC) in WSN:** Fundamentals of MAC protocols for sensor networks, Sensor-MAC (S-MAC) / Sensor-MAC case study **Routing in WSN:** Routing challenges and design issues in WSNs, Routing strategies in WSNs, IEEE 802.15.4 LR-WPAN standard (case study), **Transport Layer and Middleware in WSN:** Traditional transport control protocols, Transport protocol design issues in WSNs, WSN middleware architecture
	Module 2 (15 hours):
	Wireless Transmission Fundamentals: Frequency for radio transmission, Signals, Antennas, Signal propagation, Multiplexing, Modulation, Spread spectrum, Cellular systems **GSM and Mobile Communication Systems:** GSM mobile services, System architecture, Radio interface, Protocols, Localization and calling, Handover, Security, New data services **Advanced Mobile Communication Systems:** DECT: system architecture and protocol architecture; TETRA, UMTS, IMT-2000 **Satellite and Broadcast Systems:** History of satellite systems, Applications, Satellite basics: GEO, LEO, MEO; Routing in satellite systems, Localization, Handover

10	**Text Books** 1. Wireless Sensor Networks Technology, Protocols, and Applications ,Kazem Sohraby, Daniel Minoli and TaiebZnati, John Wiley & Sons, 2017 2. Protocols and Architectures for Wireless Sensor Network, Holger Kerl, Andreas Willig, John Wiley and Sons, 2015	
11	**Reference Books** 1. Fundamentals of Wireless Sensor Networks, Theory and Practice, Waltenegus Dargie, Christian Poellabauer , Wiley Series on wireless Communication and Mobile Computing, 2011 2. Networking Wireless Sensors, Bhaskar Krishnamachari , Cambridge University Press, 2005	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

Name of the Course: Software Testing & Quality Assurance Practical

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** The Software Testing & Quality Assurance Practical course provides hands-on exposure to automated and manual testing methodologies using industry-standard tools. It covers Selenium IDE, WebDriver, Selenium Grid, TestNG, Jenkins, JMeter, and Bugzilla, enabling students to design, execute, and manage test cases while implementing automation frameworks and continuous integration workflows. **Relevance:** With software quality being a critical success factor in modern IT systems, practical expertise in automation and testing tools is highly relevant. This course bridges theoretical QA concepts with real-time implementation, preparing students for agile and DevOps-driven development environments. **Usefulness:** The course develops practical competencies in test suite creation, cross-browser testing, data-driven testing, framework development (POM), CI/CD integration, load testing, and defect lifecycle management. These skills enable students to validate application reliability, performance, and usability effectively. **Application:** The practical skills acquired are applicable in web application testing, enterprise software validation, regression automation, performance benchmarking, cross-browser compatibility testing, and CI-based quality monitoring. Tools like Selenium, JMeter, Jenkins, and Bugzilla are widely used in IT service companies and product-based organizations. **Interest:** The course is engaging because students automate real-world web functionalities such as login validation, file uploads, table handling, and API responses. Exposure to automation frameworks, continuous integration, load testing, and bug tracking provides an industry-simulated testing experience. **Connection with Other Courses:**

		This course complements Software Testing & QA (theory), Software Engineering, Web Development, Operating Systems, and DevOps-related subjects. It reinforces understanding of SDLC, defect management, automation strategies, and performance optimization. **Demand in the Industry:** There is strong industry demand for automation testers, QA engineers, and DevOps-integrated testing professionals. Knowledge of Selenium, TestNG, Jenkins, JMeter, and defect tracking tools is highly valued across IT services, fintech, e-commerce, and product development firms. **Job Prospects:** Students can pursue roles such as automation test engineer, QA analyst, Selenium automation tester, performance testing trainee, DevOps testing associate, or defect management analyst. Advanced expertise can lead to roles such as test lead, automation architect, or quality assurance manager.
2	**Vertical:**	Major (Elective)
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted:**	60 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** **CO 1.** To develop proficiency in designing and executing manual and automated test cases using Selenium tools. **CO 2.** To enable implementation of automation frameworks such as Page Object Model and TestNG-based testing. **CO 3.** To provide hands-on experience in cross-browser testing, CI integration, and load testing. **CO 4.** To strengthen understanding of defect tracking and test reporting using industry tools. **CO 5.** To prepare students for professional QA and automation roles through tool-based implementation.	
8	**Course Outcomes (OC):** After successful completion of this course, students would be able to - **OC 1.** Design and execute automated test scripts using Selenium IDE and WebDriver for functional validation. **OC 2.** Implement automation frameworks (POM, TestNG) and perform cross-browser testing using Selenium Grid.	

	OC 3. Integrate automation scripts with CI tools like Jenkins and generate structured test reports. **OC 4.** Perform load testing using Apache JMeter and analyze performance metrics. **OC 5.** Manage and track software defects using Bugzilla and prepare structured defect lifecycle reports.
9	**Modules:** **Module 1 (30 hours):**
	Creation of Test Suite Using Selenium IDE Tool: Selenium IDE (Chrome/Firefox Extension) Install Selenium IDE and create a test suite containing a minimum of four test cases to validate different web page formats (HTML page, XML response, JSON API output, and a dynamic form page). Execute the suite and export the automation script in WebDriver format. **Cross-Website Functional Testing Using Selenium IDE** Tool: Selenium IDE Develop and execute a test suite for two different websites. Perform functional operations such as clicking links, filling forms, selecting dropdown options, and verifying displayed content. Analyze test results and generate execution logs. **Manual Test Case Design and Execution** Tool: MS Excel / Google Sheets Prepare a Test Plan, Test Scenarios, and Test Cases for a given web application. Execute test cases manually and prepare a Test Execution Report and Defect Report as per STLC process. **Selenium Server (Grid) Configuration and Remote Execution** Tool: Selenium Server (Grid) Install and configure Selenium Server in Hub-Node architecture. Execute a test script on multiple browsers using Remote WebDriver and compare cross-browser results. **Login Automation and Validation** Tool: Selenium WebDriver (Java/PHP) Write an automation script to perform login on a specified web page. Verify successful login using assertions such as URL validation, welcome message validation, and logout visibility check. **Web Element Interaction and Synchronization Handling** Tool: Selenium WebDriver (Java) Develop a script to automate interaction with various web elements (textbox, radio button, checkbox, dropdown, alert, and frames). Implement implicit and explicit waits for synchronization handling. **Data-Driven Testing Using Excel Integration** Tool: Selenium WebDriver + Apache POI

	Write a program to update 10 student records in an Excel file and validate changes on the web application. Perform data reading, writing, and verification using Apache POI library. **Data Extraction and Score Analysis from Web Table** Tool: Selenium WebDriver Automate extraction of student marks from a dynamic web table and determine the number of students scoring above 60 in at least one subject. Compute total count and percentage. **Object Identification and Counting on Web Page** Tool: Selenium WebDriver Write a program to identify and count the total number of objects present on a web page including links, images, buttons, and input elements. Display categorized counts. **List and Combo Box Item Verification** Tool: Selenium WebDriver Automate the process of identifying a list or dropdown box on a web page and determine the total number of items. Validate default selected value and perform selection testing.
	Module 2 (30 hours):
	Checkbox Identification and Validation Tool: Selenium WebDriver Write a program to count the total number of checkboxes on a web page, including checked and unchecked counts. Perform dynamic selection and validation. **Dynamic Web Table Handling and Sorting Validation** Tool: Selenium WebDriver Automate extraction of data from a dynamic web table. Validate sorting functionality and verify correctness of filtered results. **File Upload and Download Automation** Tool: Selenium WebDriver + Robot Class Develop a script to automate file upload and download functionality. Validate successful upload and verify file existence after download. **Screenshot Capture and Logging Mechanism** Tool: Selenium WebDriver + Log4j Implement screenshot capture on test failure and integrate logging mechanism using Log4j. Generate execution logs and analyze error messages. **Page Object Model (POM) Framework Implementation** Tool: Selenium WebDriver + TestNG Design and implement automation framework using Page Object Model (POM).

	Create reusable page classes and execute modular test cases. **TestNG Framework Implementation and Reporting** Tool: Selenium WebDriver + TestNG Create and execute automated test cases using TestNG annotations. Group test cases, define priorities, and generate HTML test reports. **Continuous Integration using Jenkins** Tool: Jenkins + Selenium WebDriver Configure Jenkins to execute Selenium automation scripts automatically. Generate build reports and analyze test execution history. **Cross-Browser Testing Using Selenium Grid** Tool: Selenium Grid Execute automation scripts across multiple browsers (Chrome, Firefox, Edge) using Selenium Grid. Compare execution time and compatibility issues. **Load Testing Using Apache JMeter** Tool: Apache JMeter Design and execute load testing scenarios for a web application. Configure thread groups, ramp-up time, and loop count. Analyze response time, throughput, and error percentage using graphical reports. **Bug Tracking and Defect Lifecycle Management** Tool: Bugzilla Log software defects using Bugzilla. Assign severity and priority levels. Track defect lifecycle stages and generate defect summary reports.	
10	**Text Books** 1. Software Engineering for Students, A Programming Approach, Douglas Bell, 4th Edition,, Pearson Education, 2005 2. Software Engineering – A Practitioners Approach, Roger S. Pressman, 7th Edition, Tata McGraw Hill	
11	**Reference Books** 1. Quality Management, Donna C. S. Summers, 5th Edition, Prentice-Hall. 2. Software Testing and Quality Assurance Theory and Practice, Kshirsagar Naik, Priyadarshi Tripathy, John Wiley & Sons, Inc., Publication.	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

Name of the Course: Wireless & Sensor Networks Practical

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** The Wireless & Sensor Networks Practical course provides hands-on experience in simulating, configuring, and analyzing wireless sensor nodes, ad-hoc networks, and mobile communication systems. Using tools such as TinyOS, nesC, TOSSIM, and network simulators, students explore sensor node architecture, routing protocols, MAC mechanisms, and network performance evaluation in dynamic wireless environments. **Relevance:** With the rapid growth of IoT, smart cities, industrial automation, and mobile communication systems, practical understanding of WSN and MANET technologies is highly relevant. This course bridges theoretical concepts with simulation-based implementation, enabling students to analyze real-world wireless network behavior and performance constraints. **Usefulness:** The course equips students with skills in sensor node programming, event-driven system modeling, routing protocol simulation, MAC protocol comparison, and energy-aware communication design. It strengthens analytical abilities in evaluating throughput, delay, packet delivery ratio, and energy efficiency in wireless networks. **Application:** Practical exposure gained in this course applies to IoT deployments, environmental monitoring systems, mobile ad-hoc networks, telecom infrastructure, and satellite communication systems. Skills in routing protocol analysis, network coverage evaluation, and MAC optimization are essential in wireless system design and network planning. **Interest:** The course is technically engaging as students simulate dynamic topologies, observe packet animations, analyze routing behaviors, and evaluate energy conservation strategies. Working with TinyOS, nesC, TOSSIM, and protocol simulations makes learning interactive and closely aligned with real-world wireless communication scenarios.

		Connection with Other Courses: This course complements Wireless & Sensor Networks (theory), Computer Networks, Operating Systems, Embedded Systems, IoT, and Cloud Computing. It reinforces protocol analysis, system-level programming, performance evaluation, and distributed system design concepts. **Demand in the Industry:** Industries working in telecommunications, IoT solutions, defense communication systems, automotive networks, and smart infrastructure require professionals skilled in wireless protocol design and performance evaluation. Expertise in routing protocols, MAC strategies, and energy-efficient network design is increasingly valued. **Job Prospects:** Students can pursue roles such as wireless network engineer, IoT system developer, telecom protocol analyst, embedded systems engineer, network simulation engineer, or communication systems trainee. Advanced specialization can lead to research and development roles in next-generation wireless technologies.
2	**Vertical:**	Major (Elective)
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted:**	60 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** **CO 1.** To provide practical understanding of sensor node architecture and TinyOS-based programming. **CO 2.** To develop skills in simulating wireless sensor and ad-hoc networks using TOSSIM and network simulation tools. **CO 3.** To enable implementation and analysis of routing and MAC protocols in WSN and MANET environments. **CO 4.** To evaluate network performance metrics including throughput, delay, coverage, and energy efficiency. **CO 5.** To strengthen practical knowledge of wireless communication and mobile network simulation.	
8	**Course Outcomes (OC):** After successful completion of this course, students would be able to -	

	OC 1. Design and simulate sensor node architectures and TinyOS-based applications using nesC and TOSSIM. **OC 2.** Implement and analyze routing protocols (AODV, DSR) and generate routing tables under dynamic topologies. **OC 3.** Simulate and evaluate MAC protocols (CSMA/CA, TDMA, energy-aware MAC) for performance and energy efficiency. **OC 4.** Measure and compare network performance metrics such as throughput, packet delivery ratio, and delay. **OC 5.** Model and interpret wireless, MANET, and cellular communication scenarios through simulation and coverage analysis.
9	**Modules:** **Module 1 (30 hours):**
	Study of Sensor Node Hardware Architecture Design a conceptual block diagram of a wireless sensor node and analyse the role of sensing unit, microcontroller, transceiver, memory, and power supply in energy-constrained environments. **Sensor Mote and Base Station Communication Study** Develop a simple communication scenario between multiple sensor motes and a base station to study data aggregation and sink node operation. **Understanding TinyOS Architecture and Execution Model** Examine the event-driven architecture of TinyOS and implement a simple application to demonstrate its non-preemptive scheduler. **Implementation of nesC Programming Model** Create a basic nesC application using modules and configurations to understand component wiring and interface binding. **Implementation of Events, Commands, and Tasks in TinyOS** Develop a TinyOS program that toggles LEDs using events and tasks to demonstrate split-phase execution. **Simulation of Single Mote using TOSSIM** Simulate a single sensor node in TOSSIM and observe execution logs to understand the TinyOS runtime environment. **Mote-to-Mote Radio Communication Simulation** Simulate packet transmission between two motes and analyze signal strength and packet loss under varying radio gain values. **Mote-to-PC Serial Communication Simulation** Implement serial communication between a sensor mote and PC to monitor real-time data transmission. **Creation of Simple Adhoc Network using TOSSIM** Create a multi-node ad-hoc network in TOSSIM and evaluate broadcast

	communication among nodes. **Routing Table Generation and Analysis in WSN** Implement a simple routing mechanism and analyze routing table updates during topology changes.
	Module 2 (30 hours):
	Basic MANET Implementation with Packet Animation Simulate a mobile ad-hoc network and visualize packet movement to study dynamic topology formation. **Implementation of AODV Routing Protocol** Simulate the AODV routing protocol and analyze route discovery and route maintenance mechanisms. **Implementation of DSR Routing Protocol** Simulate the DSR protocol and compare routing overhead with AODV. **Performance Evaluation of MANET Protocols** Measure throughput, packet delivery ratio, and end-to-end delay for different MANET routing protocols. **Simulation of CSMA/CA MAC Protocol** Implement CSMA/CA protocol simulation and evaluate collision avoidance mechanisms. **Simulation of TDMA-based MAC Protocol** Simulate TDMA slot allocation and compare energy efficiency with CSMA. **Energy-Aware MAC Protocol for WSN** Implement duty-cycling MAC protocol and evaluate energy conservation in sensor networks. **MANET Simulation with Directional Antenna** Simulate MANET using directional antennas and analyze improvements in spatial reuse and interference reduction. **Wireless Sensor Network Deployment and Coverage Analysis** Simulate random and grid deployment of sensor nodes and evaluate network coverage and connectivity. **Cellular Network Simulation with Client-Server Communication** Simulate a mobile network consisting of a cell tower, central office server, web server, and web browser, and analyze packet flow during a data request-response cycle.
10	**Text Books** 1. Wireless Sensor Networks Technology, Protocols, and Applications ,Kazem Sohraby, Daniel Minoli and TaiebZnati, John Wiley & Sons, 2017

	2. Protocols and Architectures for Wireless Sensor Network, Holger Kerl, Andreas Willig, John Wiley and Sons, 2015	
11	**Reference Books** 1. Fundamentals of Wireless Sensor Networks, Theory and Practice, Waltenegus Dargie, Christian Poellabauer , Wiley Series on wireless Communication and Mobile Computing, 2011 2. Networking Wireless Sensors, Bhaskar Krishnamachari , Cambridge University Press, 2005	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

Vertical – 4

VSC
(2)
(Practical)

Name of the Course: Ethical Hacking

Sr. No.	Heading	Particulars
1	**Description the course:**	**Introduction:** The Ethical Hacking course provides a structured and defensive understanding of cyber-attack methodologies, system vulnerabilities, and penetration testing frameworks. It covers reconnaissance, scanning, enumeration, privilege escalation, cryptography weaknesses, web application threats, malware analysis, wireless security, exploit frameworks, and professional penetration testing methodologies. The course emphasizes legal boundaries and ethical responsibility in cybersecurity practices. **Relevance:** With increasing cyber threats targeting governments, enterprises, and individuals, ethical hacking has become a critical cybersecurity discipline. This course is highly relevant as it equips students with knowledge of attacker techniques from a defensive perspective, enabling them to identify, analyze, and mitigate vulnerabilities proactively. **Usefulness:** The course develops practical and analytical skills in vulnerability research, network scanning, web application security assessment, password security evaluation, and penetration testing lifecycle management. It strengthens defensive thinking by understanding exploit logic, attack surfaces, and security hardening strategies. **Application:** Ethical hacking principles are applied in security audits, penetration testing engagements, vulnerability assessments, SOC operations, web security testing, wireless security evaluation, and compliance verification. Knowledge of OWASP vulnerabilities, SQL injection logic, buffer overflow concepts, DoS mitigation, and exploit frameworks supports secure system design and organizational defense. **Interest:** The subject is engaging because it explores real-world cyber-attack techniques such as social engineering, session hijacking, ARP poisoning, SQL injection, and

		malware analysis. Understanding how attackers think and how vulnerabilities are exploited makes the course technically stimulating and socially impactful. **Connection with Other Courses:** This course complements Cyber & Information Security, Computer Networks, Operating Systems, Cryptography, Software Testing, and Cloud Security. It integrates networking fundamentals, encryption principles, system-level understanding, and secure coding practices into a unified security-oriented framework. **Demand in the Industry:** There is strong global demand for ethical hackers, penetration testers, SOC analysts, and cybersecurity consultants. Organizations require professionals skilled in vulnerability assessment, exploit analysis, web security testing, and secure configuration management to protect digital infrastructure. **Job Prospects:** Students can pursue roles such as penetration testing trainee, cybersecurity analyst, SOC analyst, vulnerability assessment engineer, security consultant, or incident response associate. Advanced expertise can lead to positions such as red team specialist, security architect, or cybersecurity researcher.
2	**Vertical:**	VSC
3	**Type:**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted:**	60 hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** **CO 1.** To introduce foundational concepts, terminology, and legal frameworks in ethical hacking. **CO 2.** To develop understanding of reconnaissance, scanning, enumeration, and system exploitation techniques from a defensive perspective. **CO 3.** To provide knowledge of web application, network, and wireless security vulnerabilities and mitigation strategies. **CO 4.** To familiarize students with exploit frameworks and structured penetration testing methodologies.	

	CO 5. To inculcate ethical responsibility and professional reporting standards in cybersecurity practice.
8	**Course Outcomes (OC):** After successful completion of this course, students would be able to - **OC 1.** Explain and analyze various cyber-attack methodologies including reconnaissance, scanning, and privilege escalation techniques. **OC 2.** Identify and assess vulnerabilities in web applications, networks, wireless systems, and authentication mechanisms. **OC 3.** Evaluate cryptographic weaknesses, session management flaws, and malware behaviors from a defensive standpoint. **OC 4.** Apply structured penetration testing methodologies and prepare professional vulnerability assessment reports. **OC 5.** Recommend appropriate mitigation strategies and security hardening measures while adhering to legal and ethical guidelines.
9	**Modules:** **Module 1 (30 hours):**
	Foundations of Ethical Hacking and Cyber Terminology: Study core terminology including hacking types, hacker classes, hacktivism, ethical hacking phases, and legal boundaries. Prepare a structured lifecycle model of ethical hacking. **Vulnerability Research and Disclosure Mechanisms:** Understand vulnerability lifecycle, CVE identification, CVSS scoring, responsible disclosure, and bug bounty frameworks. Analyze a real vulnerability case study. **Footprinting and Information Gathering Methodology:** Explore passive and active reconnaissance techniques including OSINT, competitive intelligence, and search engine reconnaissance strategies. **DNS Enumeration and Domain Intelligence:** Study DNS structure, record types (A, MX, NS, TXT, CNAME), WHOIS lookup, and ARIN databases to understand infrastructure mapping. **Social Engineering and Human Exploitation Techniques:** Examine psychological manipulation attacks including phishing, pretexting, baiting, and impersonation. Analyze real-world social engineering cases. **Network Scanning and Port Scanning Techniques:** Understand TCP/IP behavior and scanning methodologies including SYN, FIN, NULL, XMAS, and ACK scans, along with firewall detection logic. **Enumeration and Service Identification:** Study enumeration techniques such as SNMP enumeration, NetBIOS scanning, and service fingerprinting to identify exposed system services. **System Hacking and Privilege Escalation Concepts:** Analyze password cracking methods, privilege escalation techniques, rootkits, and spyware technologies from a defensive perspective. **Cryptography and Password Security Analysis:** Differentiate between encryption

	and hashing mechanisms, password storage models, salting techniques, and cryptographic weaknesses. **Network Sniffing and Man-in-the-Middle Attacks:** Study packet sniffing techniques, ARP poisoning, MAC flooding, DNS spoofing, and countermeasures to secure network communication.
	Module 2 (30 hours):
	Denial of Service and Botnet Architecture: Examine types of DoS/DDoS attacks including SYN flooding and Smurf attacks. Understand botnet structures and mitigation strategies. **Session Hijacking and Token Security:** Analyze session management vulnerabilities, cookie manipulation, sequence prediction, and prevention mechanisms like secure flags and HTTPS. **Web Server Vulnerabilities and Hardening Techniques:** Study common web server misconfigurations, directory traversal attacks, patch management practices, and hardening strategies. **Web Application Threats and OWASP Vulnerabilities:** Examine major web application vulnerabilities (OWASP Top 10), input validation flaws, and search engine exploitation techniques. **SQL Injection – Attack Logic and Prevention:** Understand SQL injection mechanisms, query manipulation techniques, database vulnerabilities, and secure coding practices. **Buffer Overflow and Memory Exploitation Concepts:** Study stack-based buffer overflow, memory layout, return address overwriting, and secure programming techniques. **Wireless Network Security and Authentication Mechanisms:** Compare WEP, WPA, WPA2, WPA3 protocols, wireless sniffing risks, rogue access points, and wireless security best practices. **Malware, Keyloggers, and Spyware Analysis:** Examine the architecture of keyloggers and spyware, detection methods, behavioral analysis, and endpoint protection mechanisms. **Exploit Frameworks and Ethical Use of Metasploit:** Study exploit lifecycle, payload concepts, post-exploitation activities, and ethical/legal considerations of penetration testing tools. **Penetration Testing Methodologies and Reporting:** Understand PTES and OWASP methodologies, penetration testing phases, risk assessment models, and professional report preparation standards.
10	**Text Books** 1. CEH official Certfied Ethical Hacking Review Guide, Wiley India Edition

11	**Reference Books** 1. Certified Ethical Hacker: Michael Gregg, Pearson Education 2. Certified Ethical Hacker: Matt Walker, TMH.	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**