University of Mumbai

CIRCULAR:-

      Attention of the Principals of the Affiliated Colleges and Directors of the Recognized Institutions in Faculty of Science & Technology is invited to this office Circular No. UG/63 of 2018-19 dated 06th July, 2018 relating to the revised syllabus as per the (CBCS) for the T.Y.B. Sc. In Computer Science (Sem -V & VI).

      They are hereby informed that the recommendations made by the Board of Deans at its meeting held on 27th June, 2023 **vide** item No. 6.2 (R) have been accepted by the Academic Council at its meeting held on 27th June, 2023 **vide** item No. 6.2 (R) and that in accordance therewith, the **revised syllabus of T.Y.B. Sc. (Computer Science) (CBCS) (Sem – V & VI)** has been brought into force with effect from the academic year 2023-24.

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

MUMBAI – 400 032
13th July, 2023

(Prof. Sunil Bhirud)
I/c. REGISTRAR

To

      The Principals of the Affiliated Colleges and Directors of the Recognized Institutions in Faculty of Science & Technology.

A.C/6.2 (R) /27/06/2023

Copy forwarded with Compliments for information to:-

1) The Dean, Faculty of Science & Technology,
2) The Chairman, Board of Studies **Computer Science**,
3) The Director, Board of Examinations and Evaluation,
4) The Director, Board of Students Development,
5) The Director, Department of Information & Communication Technology,
6) The Co-ordinator, MKCL.

Copy for information and necessary action :-

1. The Deputy Registrar, College Affiliations & Development Department (CAD),
2. College Teachers Approval Unit (CTA),
3. The Deputy Registrar, (Admissions, Enrolment, Eligibility and Migration Department (AEM),
4. The Deputy Registrar, Academic Appointments & Quality Assurance (AAQA)
5. The Deputy Registrar, Research Administration & Promotion Cell (RAPC),
6. The Deputy Registrar, Executive Authorities Section (EA)
   He is requested to treat this as action taken report on the concerned resolution adopted by the Academic Council referred to the above circular.
7. The Deputy Registrar, PRO, Fort, (Publication Section),
8. The Deputy Registrar, Special Cell,
9. The Deputy Registrar, Fort Administration Department (FAD) Record Section,
10. The Deputy Registrar, Vidyanagari Administration Department (VAD),

Copy for information :-

1. The Director, Dept. of Information and Communication Technology (DICT), Vidyanagari,
   He is requested to upload the Circular University Website
2. The Director of Department of Student Development (DSD),
3. The Director, Institute of Distance and Open Learning (IDOL Admin), Vidyanagari,
4. All Deputy Registrar, Examination House,
5. The Deputy Registrars, Finance & Accounts Section,
6. The Assistant Registrar, Administrative sub-Campus Thane,
7. The Assistant Registrar, School of Engg. & Applied Sciences, Kalyan,
8. The Assistant Registrar, Ratnagiri sub-centre, Ratnagiri,
9. P.A to Hon'ble Vice-Chancellor,
10. P.A to Pro-Vice-Chancellor,
11. P.A to Registrar,
12. P.A to All Deans of all Faculties,
13. P.A to Finance & Account Officers, (F & A.O),
14. P.A to Director, Board of Examinations and Evaluation,
15. P.A to Director, Innovation, Incubation and Linkages,
16. P.A to Director, Department of Lifelong Learning and Extension (DLLE),
17. The Receptionist,
18. The Telephone Operator,

Copy with compliments for information to :-

19. The Secretary, MUASA
20. The Secretary, BUCTU.

UNIVERSITY OF MUMBAI

Revised Syllabus for

T.Y.B.Sc. (Computer Science)

(Sem. V & VI)

(CBCS)

(With effect from the academic year 2023-24)

University of Mumbai

Syllabus for Approval

Sr. No.	Heading	Particulars
1	O: _____ **Title of Course**	T.Y.B.Sc. (Computer Science)
2	O: _____ **Eligibility**	Amended Regulations 8438 & 8439 for all UG programs (aided and non-aided) in Faculties of Arts, Science & Commerce
3	R: _____ **Passing Marks**	40% Marks
4	**No. of years/Semesters:**	3 Years/ 6 Semesters
5	**Level:**	~~P.G.~~ / U.G./ ~~Diploma~~ / Certificate **(Strike out which is not applicable)**
6	**Pattern:**	~~Yearly~~ / Semester **(Strike out which is not applicable)**
7	**Status:**	Revised / ~~New~~ **(Strike out which is not applicable)**
8	**To be implemented from Academic Year :**	From Academic Year: 2023-24

Prof. Shivram S. Garje,
Dean,
Faculty of Science and Technology

Preamble

The revised and restructured syllabus aims to provide students with a comprehensive understanding of computer science concepts, theories, and practical skills, enabling them to excel in the dynamic and rapidly evolving field of technology. The revised and restructured curriculum for the Three-year integrated course is systematically designed considering the current industry needs in terms of skill sets demanded in the new technological environment. It also endeavors to align the program structure and course curriculum with student aspirations and corporate expectations. The proposed curriculum is contextual, industry-friendly, and suitable to cater to the needs of society and the nation in the present-day environment.

The TYBSc Computer Science syllabus is structured as follows:

Core Modules: The syllabus comprises core modules that cover essential topics in computer science, ensuring a strong foundation that aim to develop computational thinking, analytical abilities, and problem-solving skills among students. The Artificial Intelligence course provides in-depth knowledge of cutting edge AI concepts and techniques empowering them to develop intelligent systems and algorithms. With a focus on safeguarding information and systems the Cyber and Information Security course equips students with essential concepts and practices in cybersecurity. The Data Science course provides students with a solid foundation in data analysis and interpretation, enabling them to extract valuable insights and make data-driven decisions. In an era dominated by cloud-based technologies, the Cloud Computing course focuses on the principles, architectures, and applications of cloud computing.

Skill Enhancement Electives: Students are required to choose skill enhancement electives to deepen their knowledge in specific areas of interest. The electives offer specialized courses such as web development, cybersecurity, data science, or software engineering. By selecting these courses, students can tailor their learning experience according to their career aspirations and industry demands. Skill Enhancement courses such as Linux Server Administration, Software Testing and Quality Assurance, Cyber Forensics, Game Programming, Data Mining and Warehousing, Wireless and Sensor Networks, Ethical Hacking, and Information Retrieval cater to specialized areas of expertise and industry demands.

Generic Electives: The syllabus also includes generic electives, which provide students with the option to explore disciplines of interest beyond their choices in Core and Discipline-Specific Elective papers. These courses broaden their horizons and allow for interdisciplinary learning.

Project Work: A significant component of the syllabus involves hands-on project work. Through practical assignments and projects, students have the opportunity to apply their theoretical knowledge to real-world scenarios. This approach fosters creativity, problem-solving skills, and innovation in designing and developing software solutions.

Assessment methods for the TY Computer Science syllabus include written examinations, practical assignments, project evaluations, and presentations. This comprehensive approach ensures that students' understanding is evaluated through various mediums, emphasizing both theoretical knowledge and practical skills.

The newly designed TY Computer Science syllabus aims to equip students with the necessary competencies to pursue careers in software development, data analysis, research, or further studies in computer science-related disciplines. It seeks to empower students with the knowledge and skills required to thrive in the ever-evolving landscape of technology and contribute to the advancement of the field.

We sincerely believe that students who undertake this program will gain a strong foundation and exposure to the basics, advanced concepts, and emerging trends in the subject. We express our gratitude to all the experts who provided valuable feedback and suggestions to improve the curriculum. We have made sincere efforts to incorporate their inputs. Special appreciation goes to the University Department of

Computer Science and colleagues from various colleges who volunteered or indirectly contributed to designing certain specialized courses and the syllabus as a whole.

T.Y.B.Sc. Computer Science Syllabus
Choice Based Credit System (CBCS)
with effect from

Academic year 2023-2024

Course Code	Course Type	Course Title	Credits	Lectures/Week
		Semester – V		
USCS501	Core Subject	Artificial Intelligence	3	3
USCSP501	Core Subject Practical	Artificial Intelligence – Practical	1	3
USCS502	Core Subject	Information & Network Security	3	3
USCSP502	Core Subject Practical	Information & Network Security – Practical	1	3
USCS5031	Skill Enhancement Elective 1* (SEE)	Linux Server Administration	3	3
USCSP5031	Skill Enhancement Elective 1* Practical (SEEP)	Linux Server Administration – Practical	1	3
USCS5032	Skill Enhancement Elective 1* (SEE)	Software Testing & Quality Assurance	3	3
USCSP5032	Skill Enhancement Elective 1* Practical (SEEP)	Software Testing & Quality Assurance – Practical	1	3
USCS5041	Skill Enhancement Elective 2* (SEE)	Cyber Forensics	3	3
USCSP5041	Skill Enhancement Elective 2* Practical (SEEP)	Cyber Forensics – Practical	1	3
USCS5042	Skill Enhancement Elective 2* (SEE)	Game Programming	3	3
USCSP5042	Skill Enhancement Elective 2* Practical (SEEP)	Game Programming – Practical	1	3
USCS5051	Generic Elective**	Project Management	2	3
USCS5052	Generic Elective**	Operations Research	2	3
USCSP505	Project	Project Work – I	2	3

* One course each from Skill Enhancement Elective 1 and Skill Enhancement Elective 2 should be selected by the student.

** One course from Generic Elective should be selected by the student

T.Y.B.Sc. Computer Science Syllabus
Choice Based Credit System (CBCS)
with effect from

Academic year 2023-2024

Semester – VI				
Course Code	**Course Type**	**Course Title**	**Credits**	**Lectures/Week**
USCS601	Core Subject	Data Science	3	3
USCSP601	Core Subject Practical	Data Science – Practical	1	3
USCS602	Core Subject	Cloud Computing and Web Services	3	3
USCSP602	Core Subject Practical	Cloud Computing and Web Services – Practical	1	3
USCS6031	Skill Enhancement Elective 1* (SEE)	Wireless and Sensor Networks	3	3
USCSP6031	Skill Enhancement Elective 1* Practical (SEEP)	Wireless and Sensor Networks – Practical	1	3
USCS6032	Skill Enhancement Elective 1* (SEE)	Information Retrieval	3	3
USCSP6032	Skill Enhancement Elective 1* Practical (SEEP)	Information Retrieval – Practical	1	3
USCS6041	Skill Enhancement Elective 2* (SEE)	Data Mining & Warehousing	3	3
USCSP6041	Skill Enhancement Elective 2* Practical (SEEP)	Data Mining & Warehousing – Practical	1	3
USCS6042	Skill Enhancement Elective 2* (SEE)	Ethical Hacking	3	3
USCSP6042	Skill Enhancement Elective 2* Practical (SEEP)	Ethical Hacking – Practical	1	3
USCS6051	Generic Elective**	Customer Relationship Management	2	3
USCS6052	Generic Elective**	Cyber Laws and IPR	2	3
USCSP605	Project	Project Work – II	2	3

* One course each from Skill Enhancement Elective 1 and Skill Enhancement Elective 2 should be selected by the student.

** One course from Generic Elective should be selected by the student

Semester V

Course Code	Course Title	Credits	Lectures /Week
USCS501	**Artificial Intelligence**	**2**	**3**

About the Course: This course provides an introduction to the field of Artificial Intelligence (AI) and explores various topics related to intelligent agents, problem-solving, knowledge representation, reasoning, machine learning, and probabilistic models. The course covers both theoretical concepts and practical applications of AI techniques. Students will gain a solid foundation in AI and develop the skills to design and implement intelligent systems.

Course Objectives:
- Understand the foundations, history, and state of the art of AI.
- Learn about intelligent agents, their environments, and the structure of agents.
- Explore different problem-solving strategies, including uninformed and informed search techniques.
- Gain knowledge of knowledge representation and reasoning methods, and apply them to solve complex problems.
- Develop an understanding of machine learning techniques, including classification, regression, and ensemble learning.

Learning Outcomes:
After successful completion of this course, students would be able to
- Demonstrate knowledge of the foundations and key concepts in the field of AI.
- Analyze and design intelligent agents for specific environments.
- Apply problem-solving techniques and algorithms to find solutions to different types of problems.
- Construct knowledge representation models and use reasoning techniques to derive new knowledge.
- Implement machine-learning algorithms and evaluate their performance for classification and regression tasks.

Unit	Topics	No of Lectures
I	**Introduction to AI and Intelligent Agents** **What Is AI:** Foundations, History and State of the Art of AI **Intelligent Agents:** Agents and Environments, Nature of Environments, Structure of Agents. **Problem Solving by searching:** Problem-Solving Agents, Uninformed Search Strategies, Informed (Heuristic) Search Strategies	15
II	**Knowledge Representation, Reasoning, and Machine Learning** **Knowledge Representation and Reasoning:** Knowledge Representation and different forms, Reasoning, Planning, Uncertainty in Knowledge Fuzzy	15

		Logic & Fuzzification	
		Machine Learning: Forms of Learning, Parametric & Non-Parametric Models, Classification, Regression, Regularization, Decision Trees, SVM, Artificial Neural Networks, Ensemble Learning, Boosting, K-NN, Gradient Descent	
III		**Probabilistic Models, Unsupervised Learning, and Reinforcement Learning** **Probabilistic models:** Statistical Learning, Learning with Complete Data, Naive Bayes Classifier, Learning with Hidden Variables: The EM Algorithm **Unsupervised Learning**: Concept of Unsupervised learning,, Association Rule Mining **Reinforcement learning:** Concept of Reinforcement learning, Q-Learning, Hidden Markov Model	**15**

Textbook(s):
1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, 3rd Edition, Pearson, 2010.

Additional Reference(s):
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

Course Code	Course Title	Credits	Lectures /Week
USCSP501	**Artificial Intelligence – Practical**	**1**	**3**
1	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.		
2	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.		
3	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.		
4	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.		
5	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.		
6	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.		
7	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.		
8	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.		

9	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.
10	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.

Course Code	Course Title	Credits	Lectures /Week
USCS502	**Information & Network Security**	**2**	**3**

About the Course: This course provides an in-depth understanding of the principles and techniques used in computer and network security. Students will explore various security topics, including encryption techniques, public-key cryptography, key management, message authentication, digital signatures, authentication protocols, network security, web security, intrusion detection, malicious software, and firewall design principles. Through theoretical learning and practical exercises, students will develop the necessary knowledge and skills to analyze, design, and implement secure systems and protect against security threats.

Course Objectives:
- Familiarize students with the fundamental principles, models, and mechanisms of computer and network security.
- Explore various encryption techniques, including symmetric and public-key cryptography, and understand their strengths, weaknesses, and real-world applications.
- Examine different authentication and key management methods to ensure secure communication and protect against unauthorized access.
- Understand the concepts and techniques of message authentication, digital signatures, and authentication protocols used in secure communication systems.
- Investigate network security measures, including IP security, web security, intrusion detection, malicious software detection, and firewall design principles.

Learning Outcomes:
After successful completion of this course, students would be able to:
- Analyze and evaluate security trends, attacks, and mechanisms, and propose effective security solutions based on the OSI security architecture.
- Apply classical encryption techniques, such as substitution and transposition ciphers, to encrypt and decrypt messages and analyze their security implications.
- Implement public-key cryptography algorithms, including RSA, and demonstrate the ability to securely exchange keys and establish secure communication channels.
- Design and implement secure authentication mechanisms, including message authentication codes and digital signatures, to ensure data integrity and non-repudiation.
- Evaluate and implement various security measures, such as IP security, web security protocols (e.g., SSL/TLS), intrusion detection systems, and firewall configurations, to protect networks and systems from unauthorized access and attacks.

Unit	Topics	No of Lectures
I	**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	**15**

	Cipher Modes of Operation, Stream Ciphers	
	Public-Key Cryptography and RSA: Principles of Public-Key Cryptosystems, The RSA Algorithm	
II	**Key Management:** Public-Key Cryptosystems, Key Management, Diffie-Hellman Key Exchange	**15**
	Message Authentication and Hash Functions: Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs, Secure Hash Algorithm, HMAC	
	Digital Signatures and Authentication: Digital Signatures, Authentication Protocols, Digital Signature Standard	
	Authentication Applications: Kerberos, X.509 Authentication, Public-Key Infrastructure	
III	**Electronic Mail Security:** Pretty Good Privacy, S/MIME	**15**
	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	

Textbook(s):

1. Cryptography and Network Security: Principles and Practice 7th edition, William Stallings, Pearson

Additional Reference(s):

1. Cryptography and Network, 2nd edition, Behrouz A Fourouzan, Debdeep Mukhopadhyay, TMH.
2. Atul Kahate, "Cryptography and Network Security", Tata McGraw-Hill.

Course Code	Course Title	Credits	Lectures /Week
USCSP502	**Information & Network Security – Practical**	1	3
1	Implementing Substitution and Transposition Ciphers: Design and implement algorithms to encrypt and decrypt messages using classical substitution and transposition techniques.		
2	RSA Encryption and Decryption: Implement the RSA algorithm for public-key encryption and decryption, and explore its properties and security considerations.		
3	Message Authentication Codes: Implement algorithms to generate and verify message authentication codes (MACs) for ensuring data integrity and authenticity.		
4	Digital Signatures: Implement digital signature algorithms such as RSA-based signatures, and verify the integrity and authenticity of digitally signed messages.		
5	Key Exchange using Diffie-Hellman: Implement the Diffie-Hellman key exchange algorithm to securely exchange keys between two entities over an insecure network.		
6	IP Security (IPsec) Configuration: Configure IPsec on network devices to provide secure communication and protect against unauthorized access and attacks.		
7	Web Security with SSL/TLS: Configure and implement secure web communication using SSL/TLS protocols, including certificate management and secure session establishment.		
8	Intrusion Detection System: Set up and configure an intrusion detection system (IDS) to monitor network traffic and detect potential security breaches or malicious activities.		
9	Malware Analysis and Detection: Analyze and identify malware samples using antivirus tools, analyze their behavior, and develop countermeasures to mitigate their impact.		
10	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.		

Course Code	Course Title	Credits	Lectures /Week
USCS5031	**Linux Server Administration**	**2**	**3**

About the Course: This course provides an in-depth understanding of Linux server administration. Students will learn the technical aspects of various Linux distributions, software management, user and group administration, file systems, core system services, networking, security, and advanced internet services. Practical hands-on exercises and real-world examples will be used to enhance the learning experience.

Course Objectives:
- Develop a solid understanding of Linux server administration principles and concepts.
- Acquire practical skills in managing users, groups, and file systems in a Linux environment.
- Learn how to configure and secure network services such as DNS, FTP, Apache web server, SMTP, POP, IMAP, and SSH.
- Gain knowledge of advanced network administration topics including NFS, Samba, DFS, NIS, LDAP, DHCP, MySQL, LAMP applications, file services, email services, chat applications, and VPN.

Learning Outcomes:
After successful completion of this course, students would be able to
- Demonstrate proficiency in managing software packages and repositories in Linux.
- Configure and administer user accounts, groups, and permissions in a Linux system.
- Implement network services such as DNS, FTP, and web servers, ensuring proper security measures.
- Design and manage advanced network services including NFS, Samba, and LDAP for efficient file sharing and user authentication.
- Apply troubleshooting techniques to identify and resolve common issues in Linux server administration.

Unit	Topics	No of Lectures
I	**Introduction:** Technical Summary of Linux Distributions, Managing Software **Single-Host Administration:** Managing Users and Groups, Booting and shutting down processes. File Systems, Core System Services, Process of configuring, compiling, Linux Kernel **Networking and Security:** TCP/IP for System Administrators, basic network Configuration, Linux Firewall (Netfilter), System and network security	15
II	**Internet Services:** Domain Name System (DNS), File Transfer Protocol (FTP), Apache web server, Simple Mail Transfer Protocol (SMPT), Post Office Protocol and Internet Mail Access Protocol (POP and IMAP), Secure Shell (SSH), Network authentication system (Kerberos), Domain Name Service (DNS), Security	15
III	**Internet Services:**	15

Network File System (NFS), Samba, Distributed File Systems (DFS), Network Information Service (NIS), Lightweight Directory Access Protocol (LDAP), Dynamic Host configuration Protocol (DHCP), MySQL, LAMP Applications, File Services, Email Services, Chat applications, Virtual Private Networking.

Textbook(s):
1. Linux Administration: A Beginner's Guide, Wale Soyinka, Seventh Edition, McGraw-Hill Education, 2016
2. Ubuntu Server Guide, Ubuntu Documentation Team, 2016

Additional Reference(s):
1. Mastering Ubuntu Server, Jay LaCroix, PACKT Publisher, 2016

Course Code	Course Title	Credits	Lectures /Week
USCSP5031	**Linux Server Administration – Practical**	**1**	**3**
1	Install DHCP Server in Ubuntu 16.04		
2	Initial settings: Add a User, Network Settings, Change to static IP address, Disable IPv6 if not needed, Configure Service, display the list of services which are running. Stop and turn OFF auto-start setting for a service if you don't need it, Sudo Settings		
3	Configure NP Server (NTPd), Install and configure NTPd, Configure NTP Client (Ubuntu and Windows)		
4	SSH Server : Password Authentication Configure SSH server to manage a server from the remote computer, SSH Client : (Ubuntu and Windows)		
5	Install DNS server BIND, Configure DNS server which resolves domain name or IP address, Install BIND 9, Configure BIND, Limit ranges You allow to access if needed.		
6	Configure DHCP Server, Configure DHCP (Dynamic Host Configuration Protocol) Server, Configure NFS server to share directories on your NFS, Configure NFS Client. (Ubuntu and Windows Client OS)		
7	Configure LDDAP Server, Configure LDAP Server in order to share users' accounts in your local networks, Add LDAP User Accounts In the networks. Install phpLDAPadmin to operate LDAP server via Web browser.		
8	Configure NIS Server in order to share users; accounts in your local networks, Configure NIS Client to bind NIS Server.		
9	Install MySQL to configure database server, Install phpMyAdmin to operate MySQL on web browser from Clients.		
10	Install Samba to share folders or files between Windows and Linux		

Course Code	Course Title	Credits	Lectures /Week
USCS5032	**Software Testing & Quality Assurance**	2	3

About the Course: This course provides an in-depth understanding of software testing principles, techniques, and quality assurance practices. Students will learn the concepts and methodologies required to ensure the quality and reliability of software systems. The course covers various aspects of software testing, including test case design, verification and validation, defect management, software quality assurance, and quality improvement techniques. Students will gain hands-on experience with industry-standard tools and techniques used in software testing and quality assurance.

Course Objectives:
- Understand the significance of software testing in ensuring software quality and reliability.
- Familiarize with the fundamental concepts and principles of software quality assurance.
- Learn different software testing techniques and methodologies for effective test case design.
- Explore the role of verification and validation in software development and testing processes.
- Gain practical experience in using software testing tools and frameworks for automated testing.

Learning Outcomes:
After successful completion of this course, students would be able to
- Explain the importance of software testing and its impact on software quality.
- Apply appropriate software testing techniques to identify and mitigate software defects.
- Design and execute test cases to verify the functionality and performance of software systems.
- Understand the principles of verification and validation and their application in software testing.
- Utilize software testing tools and frameworks to automate testing processes and improve efficiency.

Unit	Topics	No of Lectures
I	**Introduction to Software Testing and Quality Assurance** **Introduction to Software Testing:** Nature of errors and the need for 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) **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 **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	15

II	**Software Testing Techniques and Strategies** **Testing Fundamentals**: Basics of software testing process, Test case design principles and techniques, Test execution, reporting, and documentation **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 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 **Software Metrics:** Concept of software metrics and their importance, Developing and utilizing different types of metrics, Complexity metrics and their significance in testing	**15**	
III	**Defect Management and Software Quality Assurance** **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, Software Reviews: Formal Technical Reviews and their benefits, Statistical Quality Assurance and Software Reliability **Statistical process control techniques for quality assurance:** Software reliability measurement and improvement, The ISO 9000 Quality Standards and their application in software development **Quality Improvement Techniques:** 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	**15**	

Textbook(s):
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

Additional Reference(s):
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.

Course Code	Course Title	Credits	Lectures /Week
USCSP5032	**Software Testing & Quality Assurance – Practical**	**1**	**3**
1	Install Selenium IDE and create a test suite containing a minimum of 4 test cases for different web page formats (e.g., HTML, XML, JSON, etc.).		
2	Conduct a test suite for two different websites using Selenium IDE. Perform various actions like clicking links, filling forms, and verifying content.		
3	Install Selenium Server (Selenium RC) and demonstrate its usage by executing a script in Java or PHP to automate browser actions.		
4	Write a program using Selenium WebDriver to automate the login process on a specific web page. Verify successful login with appropriate assertions.		
5	Write a program using Selenium WebDriver to update 10 student records in an Excel file. Perform data manipulation and verification.		
6	Write a program using Selenium WebDriver to select the number of students who have scored more than 60 in any one subject (or all subjects). Perform data extraction and analysis.		
7	Write a program using Selenium WebDriver to provide the total number of objects present or available on a web page. Perform object identification and counting.		
8	Write a program using Selenium WebDriver to get the number of items in a list or combo box on a web page. Perform element identification and counting.		
9	Write a program using Selenium WebDriver to count the number of checkboxes on a web page, including checked and unchecked counts. Perform checkbox identification and counting.		
10	Perform load testing on a web application using JMeter. Generate and analyze load scenarios. Additionally, explore bug tracking using Bugzilla as a tool for logging and tracking software defects.		

Course Code	Course Title	Credits	Lectures /Week
USCS5041	**Cyber Forensics**	**2**	**3**

About the Course: This course introduces computer forensics, focusing on the techniques and methodologies used in investigating and analyzing digital evidence. Students will learn the fundamentals of computer investigations, data acquisition, crime scene processing, and the use of specialized tools for analysis. The course covers various aspects of computer forensics, including network forensics, mobile device forensics, e-mail investigations, and report writing.

Course Objectives:
- Understand the principles and concepts of computer forensics.
- Develop skills in conducting computer investigations using a systematic approach.
- Gain proficiency in acquiring and preserving digital evidence from different storage formats.
- Explore the use of specialized tools and software for computer forensics analysis.
- Learn the techniques for investigating network-related incidents and conducting live acquisitions.

Learning Outcomes:

After successful completion of this course, students would be able to
- Demonstrate a solid understanding of the principles and techniques used in computer forensics investigations.
- Apply systematic approaches to acquire, preserve, and analyze digital evidence from various sources.
- Utilize specialized tools and software for conducting effective computer forensics analysis.
- Develop strong skills in investigating network-related incidents, including live acquisitions and network forensics.
- Generate comprehensive and well-written reports that accurately document the findings of computer forensic investigations.

Unit	Topics	No of Lectures
I	**Introduction:** Understanding Computer Forensics, Preparing for Computer Investigations, Maintaining Professional Conduct **Computer Investigations:** Preparing a Computer Investigation, Taking a Systematic Approach, Procedures for Corporate High-Tech Investigations, Understanding Data Recovery Workstations and Software, Conducting an Investigation **Data Acquisition:** Storage Formats for Digital Evidence, Determining the Best Acquisition Method, Contingency Planning for Image Acquisitions	15
II	**Processing Crime and Incident Scenes:** Identifying Digital Evidence, Preparing for a Search, Securing a Computer Incident or Crime Scene, Seizing Digital Evidence at the Scene, Storing Digital Evidence **Current Computer Forensics Tools:** Evaluating Computer Forensics Tool Needs, Computer Forensics Software Tools, Computer Forensics Hardware Tools	15

		Computer Forensics Analysis and Validation: Determining What Data to Collect and Analyze, Validating Forensic Data, Addressing Data-Hiding Techniques, Performing Remote Acquisitions	
		Recovering Graphics Files: Recognizing a Graphics File, Locating and Recovering Graphics Files, Identifying Unknown File Formats	
III		**Network Forensics and Live Acquisitions:** Network Forensics Overview, Performing Live Acquisitions, Developing Standard Procedures for Network Forensics, Using Network Tools	15
		E-mail Investigations: Role of E-mail in Investigations, Investigating E-mail Crimes and Violations, Using Specialized E-mail Forensics Tools	
		Cell Phone and Mobile Device Forensics: Overview, Acquisition Procedures for Cell Phones and Mobile Devices	
		Report Writing for Investigations: Importance of Reports, Guidelines for Writing Reports, Generating Report Findings with Forensics Software Tools	

Textbook(s):

1. Bill Nelson,Amelia Philips and Christopher Steuart, "Guide to computer forensics and investigations", course technology,6th edition

Additional Reference(s):

1. Kevin Mandia, Chris Prosise, "Incident Response and computer forensics",Tata McGrawHill

Course Code	Course Title	Credits	Lectures /Week
USCSP5041	**Cyber Forensics – Practical**	1	3
1	Creating a Forensic Image using FTK Imager/Encase Imager : - Creating Forensic Image - Check Integrity of Data - Analyze Forensic Image		
2	Data Acquisition: - Perform data acquisition using: - USB Write Blocker + Encase Imager - SATA Write Blocker + Encase Imager - Falcon Imaging Device		
3	Analyze the memory dump of a running computer system. Extract volatile data, such as open processes, network connections, and registry information.		
4	Capturing and analyzing network packets using Wireshark (Fundamentals) : - Identification the live network - Capture Packets - Analyze the captured packets		
5	Using Sysinternals tools for Network Tracking and Process Monitoring : - Check Sysinternals tools - Monitor Live Processes - Capture RAM - Capture TCP/UDP packets - Monitor Hard Disk - Monitor Virtual Memory - Monitor Cache Memory		
6	Recovering and Inspecting deleted files - Check for Deleted Files - Recover the Deleted Files - Analyzing and Inspecting the recovered files - Perform this using recovery option in ENCASE and also Perform manually through command line		
7	Steganography Detection - Detect hidden information or files within digital images using steganography analysis tools. - Extract and examine the hidden content.		
8	Mobile Device Forensics - Perform a forensic analysis of a mobile device, such as a smartphone or tablet. - Retrieve call logs, text messages, and other relevant data for investigative purposes.		
9	Email Forensics		

	• Analyze email headers and content to trace the origin of suspicious emails. • Identify potential email forgeries or tampering.
10	Web Browser Forensics • Analyze browser artifacts, including history files, bookmarks, and download records. • Analyze cache and cookies data to reconstruct user-browsing history and identify visited websites or online activities. • Extract the relevant log or timestamp file, analyze its contents and interpret the timestamp data to determine the user's last internet activity and associated details.

Course Code	Course Title	Credits	Lectures /Week
USCS5042	**Game Programming**	2	3

About the Course: This course provides an in-depth exploration of 3D game development and graphics, covering topics such as vectors, transformations, 3D modeling, rendering, physics-based simulation, and game engine architecture. Students will gain hands-on experience using industry-standard tools and technologies, including DirectX, Unity, and Python-Pygame. Through a combination of theoretical study, practical exercises, and project-based learning, students will develop the skills necessary to create immersive 3D games and interactive virtual worlds.

Course Objectives:

- Understand the fundamentals of vectors, transformations, and 3D graphics.
- Develop proficiency in using industry-standard tools and technologies for 3D game development.
- Implement advanced graphics techniques, including lighting, shading, and texturing, to create visually appealing game environments.
- Apply principles of game design and create engaging and immersive gaming experiences.

Learning Outcomes:
After successful completion of this course, students would be able to
- Apply vector manipulation techniques and transformations to create and manipulate objects in 3D space.
- Utilize industry-standard tools and technologies such as Unity and DirectX for 3D game development.
- Implement advanced graphics techniques, including lighting, shading, and texturing, to create visually stunning game environments.
- Design and develop games that incorporate principles of game design to create engaging and immersive experiences.
- Deploy and showcase 3D games on various platforms, demonstrating proficiency in game development.

Unit	Topics	No of Lectures
I	**Introduction to Vectors:** Vectors: Vector Manipulation, multiplying a Vector by a Scalar, Vector Addition and Subtraction, Position Vectors, Unit Vectors, Cartesian Vectors, Vector Multiplication, Scalar Product, Example of the Dot Product, The Dot Product in Lighting Calculations, The Dot Product in Back-Face Detection, The Vector Product, The Right-Hand Rule, deriving a Unit Normal Vector for a Triangle Areas, Calculating 2D Areas **Transformations:** 2D Transformations, Matrices, Homogeneous Coordinates, 3D Transformations, Change of Axes, Direction Cosines, rotating a Point about an Arbitrary Axis, Transforming Vectors, Determinants, Perspective Projection, Interpolation	15

	3D Graphics for Game Programming: 3D Transformations, Quaternions, 3D Modeling and Rendering, Ray Tracing, Shader Models, Lighting, Color, Texturing, Camera and Projections, Culling and Clipping, Character Animation, Physics-based Simulation, Scene Graphs.	
II	**Game Engines and Design:** Game engine architecture, Engine support systems, Resources and File systems, Game loop and real-time simulation, Human Interface devices, Collision and rigid body dynamics, Game profiling. Introduction to Dx Studio, Introduction to Unity, Application layer, Game logic, Game views, managing memory, controlling the main loop, loading and caching game data, User Interface management, Game event management. **Introduction To DirectX 11:** COM, Textures and Resources Formats, The swap chain and Page flipping, Depth Buffering, Texture Resource Views, Multisampling Theory and MS in Direct3D, Feature Levels **Game Platform and Development:** 2D and 3D Game development using Python-Pygame, ModernGL, Numpy, PyGLM,Ursina (OpenGL)	**15**
III	**Unity Development Environment:** IDE Basics, Unity Concepts, Sprites, Game Loops and Functions, Game Design Strategies. **Simple Movement, Operations and Object Oriented Concepts:** Simple Rotation and Scaling, Rigidbody Components, Unity Colliders, Physics Materials, Scripting Collision Events, Primitive Data and Math, Decisions and Flow Control, Loops and Arrays, Game Design Strategies, Exceptions and Debugging, Defining Classes, Functions, Organizing and Managing Game Objects, **Advanced Programming:** Virtual Worlds, Scrolling Games, Animation, Sound Effects, Advanced Game Physics, Multiple Scenes, Artificial Intelligence, User Interfaces, Game Art, Publishing Games.	**15**

Textbook(s):
1. Mathematics for Computer Graphics, John Vince, Springer-Verlag London, 5th Edition,2017
2. Mathematics for 3D Game Programming and Computer Graphic, Eric Lengyel, Delmar
3. Introduction To 3D Game Programming With Directx® 11,Frank D Luna, Mercury Learning And Information,2012.

Additional Reference(s):
1. Computer Graphics, C Version, Donald Hern and Pauline Baker, Pearson Education, 2nd Edition, 1997
2. HLSL Development Cookbook, Doron Feinstein, PACKT Publishing,2013
3. https://docs.unity3d.com/Manual/index.html

Course Code	Course Title	Credits	Lectures /Week
USCSP5042	**Game Programming – Practical**	1	3
1	Setup DirectX 11, Window Framework and Initialize Direct3D Device, Loading models into DirectX 11 and rendering		
2	Learn Basic Game Designing Techniques with pygame.		
3	Develop Snake Game using pygame		
4	Create 2D Target Shooting Game		
5	Creating 2D Infinite Scrolling Background		
6	Create Camera Shake Effect in Unity		
7	Design and Animate Game Character in Unity		
8	Create Snowfall Particle effect in Unity		
9	Develop Android Game with Unity		
10	Create Intelligent enemies in Unity		

Course Code	Course Title	Credits	Lectures /Week
USCS5051	**Project Managament**	**2**	**3**

About the Course: The Project Management course emphasizes on the importance of effective project management in the field. Students will gain a solid understanding of project management principles, processes, and knowledge areas while exploring their application in computer science projects. The course covers key topics such as project selection and initiation, scope management, time management, cost management, quality management, human resource management, risk management, procurement management, integration management, and advanced topics relevant to the digital age. Through a combination of theory and practical exercises, students will develop the skills necessary to successfully plan, execute, and control projects in the dynamic and rapidly evolving field of Computer Science.

Course Objectives:
- Understand the fundamental concepts and characteristics of project management, including project selection, initiation, and project governance.
- Develop skills in scope management, time management, cost management, quality management, and risk management to effectively plan and control projects.
- Acquire knowledge of human resource management techniques, including team development, conflict resolution, and communication management.
- Gain insight into agile project management methodologies and tools, enabling the successful management of iterative and incremental development.
- Explore advanced topics such as stakeholder management, project leadership, project governance, and the role of technology in project management.

Learning Outcomes:
After successful completion of this course, students would be able to
- Apply project management principles, processes, and best practices to plan, execute, and control projects effectively.
- Develop project charters, define project scopes, and create work breakdown structures (WBS) to establish project objectives and deliverables.
- Create project schedules, estimate resource requirements, and monitor project progress using appropriate project management techniques.
- Employ quality assurance and control measures to ensure project deliverables meet stakeholder expectations and industry standards.
- Demonstrate effective leadership and teamwork skills, as well as the ability to manage stakeholders, resolve conflicts, and make ethical decisions in project management settings.

Unit	Topics	No of Lectures
I	**Introduction to Project Management** **Introduction to Project Management:** Definition and characteristics of a project, Importance of project management, Project management processes and knowledge areas **Project Selection, Initiation and scope Management:** Project selection criteria and methods, Project initiation and charter development, Stakeholder identification and analysis, Scope planning and definition,	15

	Work Breakdown Structure (WBS) development, Scope verification and change control		
	Project Time & Cost Management: Activity definition and sequencing, Estimating activity durations and resources, Developing the project schedule, Schedule control and monitoring, Cost estimation techniques, Budget development and monitoring, Earned Value Management (EVM), Cost control and analysis		
II	**Project Execution and Control** **Project Quality & Risk Management:** Quality planning and standards, Quality assurance and control, Process improvement and Six Sigma concept, Risk Management-Risk identification and assessment, Risk response planning, Risk monitoring and control **Project Resource & Procurement Management:** Team development and roles, Staffing, training, and motivation, Conflict resolution and communication management, Procurement planning and contract types, Solicitation, source selection, and contract administration, Vendor management and relationship building **Project Integration Management:** Project integration processes and activities, Change management and project closure, Lessons learned and knowledge transfer	15	
III	**Advanced Topics in Project Management** **Agile Project & Management in the Digital Age:** Agile principles and methodologies, Agile project planning and execution, Managing iterative and incremental development, Role of technology in project management, Virtual teams and distributed project management,Tools and software for project planning and collaboration **Effective People Management in Projects:** Leadership styles and characteristics, Team building and motivation techniques, Emotional intelligence in project management, Stakeholder identification and analysis, Stakeholder engagement and communication strategies, Conflict resolution and negotiation skills **Project Governance and Ethics:** Project governance structures and accountability, Ethical considerations in project management, Professional responsibility and codes of conduct	15	

Textbook(s):

1. Project Management for Business and Technology, 3rd edition, Pearson Education. John M. Nicholas, 2000
2. Information Technology Project Management, by Jack T. Marchewka, 4th Wiley India 2013.
3. A Guide to the Project Management Body of Knowledge (PMBOK® Guide)–Sixth Edition 6th Edition, Project Management Institute, 2017

Additional Reference(s):

1. Introduction to Software Project Management by Adolfo Villafiorita · 2016, CRC press, e book format.
2. Project Management Professional Workbook, Claudia M. Baca, Patti M. Jansen, Sybex Publication, 2013
3. Project Management, by S. J. Mantel, J. R. Meredith and etal.,1st edition, Wiley India, 2009.

Course Code	Course Title	Credits	Lectures /Week
USCS5052	**Operations Research**	**2**	**3**

About the Course: This course introduces Operations Research (OR) and its application in decision-making. It covers the fundamental concepts, techniques, and methodologies used in OR, with a focus on linear programming and duality. Students will learn how to formulate and solve optimization problems, analyze sensitivity in linear programming, and explore advanced topics such as goal programming, transportation problems, and assignment problems. The course also introduces computer software commonly used in OR for problem solving.

Course Objectives:
- To understand the fundamental principles and approaches of Operations Research.
- To develop skills in formulating and solving linear programming problems.
- To analyze the duality in linear programming and its managerial significance.
- To apply sensitivity analysis techniques to assess the impact of changes in linear programming models.
- To explore advanced topics in Operations Research, including goal programming, transportation problems, and assignment problems.

Learning Outcomes:
After successful completion of this course, students would be able to
- Define and explain the key concepts and features of Operations Research.
- Formulate and solve linear programming models using appropriate techniques.
- Apply duality concepts to analyze and interpret the results of linear programming problems.
- Conduct sensitivity analysis to assess the robustness and flexibility of linear programming solutions.
- Apply advanced techniques such as goal programming, transportation problems, and assignment problems to real-world decision-making scenarios.

Unit	Topics	No of Lectures
I	**Introduction to Operations Research(OR)** **Exploring Operations Research** – A Quantitative Approach to Decision-Making, Definitions, Features, OR Approach to Problem Solving, Models and Modelling in Operations Research, Advantages of Model Building and Operations Research Study, Applications of Operations Research, Computer Software for Operations Research **Linear Programming and Duality:** Linear Programming: Applications and Model Formulation, Structure of Linear Programming Model, General Structure of an LP Model, Assumptions of an LP Model, Advantages & LimitationsLinear Programming, Application Areas, General Mathematical Model of Linear Programming Problem, Examples of LP Model Formulation **Linear Programming-The Graphical Method**: Important Definitions, Graphical Solution Methods of LP Problems, The Simplex Method: Introduction, Standard form of an LP Problem, Simplex Algorithm	15

		(Maximization Case), Simplex Algorithm (Minimization Case), Two-Phase Method, Big-M Method	
II		**Linear Programming** **Duality in Linear Programming:** Introduction, Formulation of Dual Linear Programming Problem, Symmetrical Form, Economic Interpretation of Dual Variables, Economic Interpretation of Dual Constraints, Rules for Constructing the Dual from Primal, Standard Results on Duality, Principle of Complementary Slackness, Managerial Significance of Duality **Sensitivity Analysis in Linear Programming:** Introduction, Sensitivity Analysis **Integer Linear Programming:** Introduction, Types of Integer Programming Problems, Enumeration and Cutting Plane Solution Concept, Branch and Bound Method, Applications of Zero-One Integer Programming	**15**
III		**Advanced Topics in Operations Research** **Goal Programming:** Introduction, Difference Between LP and GP Approach, Concept of Goal Programming, Distinction among Objectives, Goals and Constraints, Goal Programming Model Formulation **Transportation Problem:** Introduction, Mathematical Model of Transportation Problem, General Mathematical Model of Transportation Problem, The Transportation Algorithm, Methods for Finding Initial Solution **Assignment Problem:** Introduction, Mathematical Models of Assignment Problem, Solution Methods of Assignment Problem, Hungarian Method for Solving Assignment Problem	**15**

Textbook(s):

1. Operations Research: Theory and Applications, J K Sharma, Trinity Press, 6th Edition , 2017
2. Introduction to Operations Research, Frederick S. Hillier, Gerald J. Lieberman, McGraw Hill Education; 11th edition, 2021

Additional Reference(s):

1. Oeprations Research, P K Gupta, S. Chand Publications, 7th Edition, 2018
2. Operations Research, P. Rama Murthy, New Age Publication, 2nd Edition
3. Operations Research: An Introduction, 10th Edition, Hamdy A. Taha, Pearson Education, 2019
4. Operations Research (Schaums Outline Series), Richard Bronson and Govindasami Naadimuthu, McGraw Hill Education, 2nd Edition, 2017

Course Code	Course Title	Credits	Lectures /Week
USCSP505	**Project Work – I**	**2**	**3**
	<u>**Refer to the Project Guidelines at the end**</u>		

Semester VI

Course Code	Course Title	Credits	Lectures /Week
USCS601	Data Science	2	3

About the Course: This course introduces the field of Data Science, covering the fundamental concepts, techniques, and tools used for data analysis, machine learning, and data visualization. Students will learn how to preprocess and analyze data, build predictive models, evaluate model performance, and effectively communicate insights through visualizations. The course also explores data management principles and practices. Practical hands-on exercises and projects using popular Data Science libraries and technologies are included to reinforce the concepts learned.

Course Objectives:
- Understand the foundations and scope of Data Science, including its applications and comparison to related fields like Business Intelligence and Artificial Intelligence.
- Develop skills in data preprocessing, including cleaning, transforming, selecting, and merging data, to ensure data quality and suitability for analysis.
- Gain knowledge of machine learning algorithms and techniques, such as regression, classification, clustering, and ensemble learning, to build predictive models and make data-driven decisions.
- Learn how to evaluate and select models using appropriate evaluation metrics and cross-validation techniques to ensure reliable and robust model performance.
- Develop proficiency in data visualization techniques and tools to effectively communicate insights and tell compelling stories using data.

Learning Outcomes:

After successful completion of this course, students would be able to
- Apply data preprocessing techniques to clean and transform raw data, handle missing values and outliers, and merge datasets.
- Implement machine-learning algorithms to perform tasks such as regression, classification, clustering, and ensemble learning.
- Evaluate and compare different machine learning models using appropriate evaluation metrics and cross-validation techniques.
- Create informative and visually appealing data visualizations to communicate insights and patterns in data.
- Understand the principles and practices of data management, including data governance, data quality assurance, and data privacy considerations.

Unit	Topics	No of Lectures
I	**Introduction to Data Science and Data Preprocessing** **What is Data Science?:** Definition and scope of Data Science, Applications and domains of Data Science, Comparison with other fields like Business Intelligence (BI), Artificial Intelligence (AI), Machine	15

		Learning (ML), and Data Warehousing/Data Mining (DW-DM)	
		Data Types and Sources: Different types of data: structured, unstructured, semi-structured, Data sources: databases, files, APIs, web scraping, sensors, social media	
		Data Preprocessing: Data cleaning: handling missing values, outliers, duplicates, Data transformation: scaling, normalization, encoding categorical variables, Feature selection: selecting relevant features/columns, Data merging: combining multiple datasets	
		Data Wrangling and Feature Engineering: Data wrangling techniques: reshaping, pivoting, aggregating, Feature engineering: creating new features, handling time-series data Dummification: converting categorical variables into binary indicators, Feature scaling: standardization, normalization	
		Tools and Libraries: Introduction to popular libraries and technologies used in Data Science like Pandas, NumPy, Sci-kit Learn, etc.	
II		**Data Analysis and Machine Learning**	15
		Exploratory Data Analysis (EDA): Data visualization techniques: histograms, scatter plots, box plots, etc., Descriptive statistics: mean, median, mode, standard deviation, etc., Hypothesis testing: t-tests, chi-square tests, ANOVA, etc.	
		Introduction to Machine Learning: Supervised learning: classification and regression, Unsupervised learning: clustering and dimensionality reduction, Bias-variance tradeoff, underfitting, and overfitting	
		Regression Analysis: Simple linear regression, Multiple linear regression, Stepwise regression, Logistic regression for classification	
		Model Evaluation and Selection: Techniques for evaluating model performance: accuracy, precision, recall, F1-score, Confusion matrix and ROC curve analysis, Cross-validation: k-fold cross-validation, stratified cross-validation, Hyperparameter tuning and model selection	
		Machine Learning Algorithms: Decision Trees and Random Forests, Support Vector Machines (SVM), Artificial Neural Networks (ANN), Ensemble Learning: Boosting and Bagging, K-Nearest Neighbors (K-NN), Gradient Descent for optimization	
III		**Model Evaluation, Data Visualization, and Management**	15
		Model Evaluation Metrics: Accuracy, precision, recall, F1-score, Area Under the Curve (AUC), Evaluating models for imbalanced datasets	
		Data Visualization and Communication: Principles of effective data visualization, Types of visualizations: bar charts, line charts, scatter plots, etc. Visualization tools: matplotlib, seaborn, Tableau, etc. Data storytelling: communicating insights through visualizations	
		Data Management: Introduction to data management activities, Data pipelines: data extraction, transformation, and loading (ETL), Data governance and data quality assurance, Data privacy and security considerations	

Textbook(s):

1. Data Science from Scratch First Principles with Python- Joel Grus O'reilly, 2nd Edition
2. Advancing into Analytics From Excel to Python and R, George Mount, Oreilly, First Edition
3. Introduction to Machine Learning with Python, Andreas C. Muller, Sarah Guido, Oreilly, First Edition

Additional Reference(s):

1. Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly,2013
2. Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication,2015
3. Hands-On Programming with R, Garrett Grolemund,1st Edition, 2014
4. An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015

Course Code	Course Title	Credits	Lectures /Week
USCSP601	**Data Science – Practical**	**1**	**3**
1	Introduction to Excel • Perform conditional formatting on a dataset using various criteria. • Create a pivot table to analyze and summarize data. • Use VLOOKUP function to retrieve information from a different worksheet or table. • Perform what-if analysis using Goal Seek to determine input values for desired output.		
2	Data Frames and Basic Data Pre-processing • Read data from CSV and JSON files into a data frame. • Perform basic data pre-processing tasks such as handling missing values and outliers. • Manipulate and transform data using functions like filtering, sorting, and grouping.		
3	Feature Scaling and Dummification • Apply feature-scaling techniques like standardization and normalization to numerical features. • Perform feature dummification to convert categorical variables into numerical representations.		
4	Hypothesis Testing • Formulate null and alternative hypotheses for a given problem. • Conduct a hypothesis test using appropriate statistical tests (e.g., t-test, chi-square test). • Interpret the results and draw conclusions based on the test outcomes.		
5	ANOVA (Analysis of Variance) • Perform one-way ANOVA to compare means across multiple groups. • Conduct post-hoc tests to identify significant differences between group means.		
6	Regression and Its Types • Implement simple linear regression using a dataset. • Explore and interpret the regression model coefficients and goodness-of-fit measures. • Extend the analysis to multiple linear regression and assess the impact of additional predictors.		
7	Logistic Regression and Decision Tree • Build a logistic regression model to predict a binary outcome. • Evaluate the model's performance using classification metrics (e.g., accuracy, precision, recall). • Construct a decision tree model and interpret the decision rules for classification.		

8	K-Means Clustering • Apply the K-Means algorithm to group similar data points into clusters. • Determine the optimal number of clusters using elbow method or silhouette analysis. • Visualize the clustering results and analyze the cluster characteristics.
9	Principal Component Analysis (PCA) • Perform PCA on a dataset to reduce dimensionality. • Evaluate the explained variance and select the appropriate number of principal components. • Visualize the data in the reduced-dimensional space.
10	Data Visualization and Storytelling • Create meaningful visualizations using data visualization tools • Combine multiple visualizations to tell a compelling data story. • Present the findings and insights in a clear and concise manner.

Course Code	Course Title	Credits	Lectures /Week
USCS602	**Cloud Computing and Web Services**	**2**	**3**

About the Course: The course "Cloud Computing and Web Services" provides an in-depth understanding of cloud computing fundamentals and web service technologies. Students will learn about different types of clouds, cloud deployment models, and cloud platforms. They will also explore key concepts of virtualization, security in cloud computing, and popular cloud computing platforms such as OpenStack and AWS. Through practical exercises and hands-on projects, students will gain the skills required to design, deploy, and manage cloud-based applications and services.

Course Objectives:
- Understand the basics of cloud computing, including types of clouds, deployment models, and essential characteristics of cloud platforms.
- Explore web services technologies such as SOAP and REST and understand their role in distributed computing and parallel computing.
- Gain proficiency in utilizing virtualization technologies, including creating virtual machines and managing virtualized environments using tools like KVM and oVirt.
- Explore and utilize popular cloud computing platforms such as OpenStack and AWS to architect, deploy, and manage cloud-based applications and services.
- Learn about cloud security fundamentals, including confidentiality, integrity, availability, and secure development practices.

Learning Outcomes:
After successful completion of this course, students would be able to
- Demonstrate a comprehensive understanding of cloud computing concepts, including different types of clouds and their characteristics.
- Implement and utilize web service technologies, such as SOAP and REST, to develop distributed and parallel computing applications.
- Design, deploy, and manage cloud-based applications and services using popular cloud computing platforms such as OpenStack and AWS.
- Apply secure development practices and implement cloud security policies to ensure the confidentiality, integrity, and availability of cloud software solutions.
- Utilize virtualization technologies to create and manage virtualized environments, considering the benefits and drawbacks of virtualization.

Unit	Topics	No of Lectures
I	**Cloud Computing Basics** **Web Services** – Distributed Computing, Parallel Computing, WSDL structure, SOAP- Structure of SOAP Message (In JAX-WS), SOAP Messaging Architecture, SOAP Header, Client-side SOAP Handler, REST- What is REST? HTTP methods, Java API for RESTful Web Services (JAX-RS) **Virtulization:-** Characteristics of Virtualized Environments.. Pros and	**15**

		Cons of Virtualization. Virtualization using KVM, Creating virtual machines, oVirt - management tool for virtualization environment.	
II		**Introduction to Cloud Computing:** Definition, Types of Clouds, Deployment of software solutions and web applications, Types of Cloud Platforms, Essential characteristics – On-demand self-service, Broad network access, Location independent resource pooling ,Rapid elasticity , Measured service, Comparing cloud providers with traditional IT service providers **Cloud Computing Software Security fundamentals:** Cloud Information Security Objectives, Confidentiality, Integrity, Availability, Cloud Security Services, Relevant Cloud Security Design Principles, Secure Cloud Software Requirements, Secure Development practices, Approaches to Cloud Software Requirement Engineering, Cloud Security Policy Implementation.	15
III		**Cloud Applications** **CloudSim:** Introduction to Simulator, understanding CloudSim simulator, CloudSim Architecture(User code, CloudSim, GridSim, SimJava) Understanding Working platform for CloudSim, **OpenStack:** Introduction to OpenStack, OpenStack test-drive, Basic OpenStack operations, OpenStack CLI and APIs, Tenant model operations, Quotas, Private cloud building blocks, Controller deployment, Networking deployment, Block Storage deployment, Compute deployment, deploying and utilizing OpenStack in production environments, Building a production environment, Application orchestration using OpenStack Heat **AWS:** Architecting on AWS, Building complex solutions with Amazon Virtual Private Cloud (Amazon VPC)	15

Textbook(s):
1. Java Web Services Up and Running 2nd edition, Martin Kalin, O'Reilly (2013)
2. Pro Power Shell for Amazon Web Services, Brian Beach, Apress, 2014
3. Enterprise Cloud Computing Technology, Architecture, Applications, Gautam Shroff, Cambridge University Press, 2010
4. Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, Tata McGraw Hill Education Private Limited, 2013
5. OpenStack in Action, V. K. CODY BUMGARDNER, Manning Publications Co, 2016

Additional Reference(s):
1. OpenStack Essentials, Dan Radez, PACKT Publishing, 2015 2
2. OpenStack Operations Guide, Tom Fifield, Diane Fleming, Anne Gentle, Lorin Hochstein, Jonathan Proulx, Everett Toews, and Joe Topjian, O'Reilly Media, Inc., 2014
3. https://www.openstack.org

Course Code	Course Title	Credits	Lectures /Week
USCSP602	**Cloud Computing and Web Services – Practical**	**1**	**3**
1	Define a simple services like Converting Rs into Dollar and Call it from different platform like JAVA and .NET		
2	Create a Simple SOAP service.		
3	Create a Simple REST Service.		
4	Develop application to consume Google's search / Google's Map RESTful Web service.		
5	Installation and Configuration of virtualization using KVM.		
6	Develop application to download image/video from server or upload image/video to server using MTOM techniques		
7	Implement FOSS-Cloud Functionality VSI (Virtual Server Infrastructure) Infrastructure as a Service (IaaS), Storage		
8	Implement FOSS-Cloud Functionality - VSI Platform as a Service (PaaS),		
9	Using AWS Flow Framework develop application that includes a simple workflow. Workflow calls an activity to print hello world to the console. It must define the basic usage of AWS Flow Framework, including defining contracts, implementation of activities and workflow coordination logic and worker programs to host them		
10	Implementation of Openstack with user and private network creation.		

Course Code	Course Title	Credits	Lectures /Week
USCS6031	**Wireless Sensor Networks**	2	3

About the Course: This course provides a comprehensive understanding of Wireless Sensor Networks (WSNs) and their applications. It covers the fundamental concepts, architectural elements, advantages, and challenges of WSNs. Students will explore sensor node technology, network architecture, optimization goals, and design principles for WSNs. The course also delves into wireless transmission, telecommunication systems, and introduces the concepts of WSN operating systems and ad-hoc networks. Through practical examples and case studies, students will gain hands-on experience in medium access control protocols, routing strategies, transport control protocols, and WSN middleware architecture.

Course Objectives:
- Provide students with a comprehensive understanding of Wireless Sensor Networks (WSNs), including their basic architectural elements, advantages, and challenges.
- Introduce students to the key technologies and protocols used in WSNs, such as medium access control (MAC) protocols, routing strategies, and transport control protocols.
- Familiarize students with wireless transmission principles and telecommunication systems relevant to WSNs, including frequency, signals, antennas, and satellite systems.
- Develop students' practical skills in designing and implementing WSN solutions by exploring WSN operating systems, ad-hoc networks, and optimization goals.

Learning Outcomes:
After successful completion of this course, students would be able to
- Understand the fundamental concepts, architectural elements, and optimization goals of Wireless Sensor Networks (WSNs) and apply this knowledge to analyze and design WSN solutions.
- Evaluate and compare different medium access control protocols and routing strategies in WSNs, and make informed decisions to ensure efficient and reliable communication.
- Demonstrate knowledge of wireless transmission technologies, such as frequency, signals, antennas, and propagation, and analyze their impact on WSN performance.
- Assess the role of telecommunication systems, satellite, broadcast systems in WSNs, and understand their applications and implications for WSN deployments.

Unit	Topics	No of Lectures
I	**Introduction and Overview of Wireless Sensor Networks:** Basic Sensor Network Architectural Elements, Advantage and challenges, Applications, Sensor Node Technology, Sensor Taxonomy, WN Operating Environment, Radio Technology, Network architecture, Optimization goals and figures of merit, Design principles for WSNs, Service interfaces of WSNs, Gateway concepts. **Wireless Sensor Network Operating Systems and Ad-hoc Networks:** Overview of Wireless Sensor Network Operating Systems, Examples of WSN Operating Systems Ad-hoc Networks in Wireless Sensor Networks,	15

	Characteristics and Challenges of Ad-hoc Networks in WSNs, Energy Efficiency Considerations in Ad-hoc Networks, Security and Privacy in Ad-hoc Networks, Examples of WSN OS, Ad-hoc Network.	
II	**Medium Access Control Protocol:** Fundamentals of MAC Protocols, Sensor-MAC Case Study **Routing in WSN:** Routing Challenges and Design Issues in Wireless Sensor Networks, , IEEE 802.15.4 LR-WPANs Standard Case Study, Routing Strategies in Wireless Sensor Networks, **Transport Control Protocol:** Traditional Transport Control Protocols, Transport Protocol Design Issues, WSN Middleware Architecture	15
III	**Wireless Transmission:** Frequency for radio transmission, Signals, Antennas, Signal propagation, Multiplexing, Modulation, Spread spectrum, Cellular systems. **Telecommunication, Satellite and Broadcast Systems:** Satellite and Broadcast Systems: GSM: Mobile services, System architecture, Radio interface, Protocols, Localization And Calling, Handover, security, New data services; DECT: System architecture, Protocol architecture; ETRA, UMTS and IMT- 2000. Satellite Systems: History, Applications, Basics: GEO, LEO, MEO; Routing, Localization, Handover.	15

Textbook(s):

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

Additional Reference(s):

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

Course Code	Course Title	Credits	Lectures /Week
USCSP6031	**Wireless Sensor Networks – Practical**	**1**	**3**
1	Understanding the Sensor Node Hardware. (For Eg. Sensors, Nodes(Sensor mote), Base Station,Graphical User Interface.)		
2	Exploring and understanding TinyOS computational concepts: - Events, Commands and Task. - nesC model - nesC Components		
3	Understanding TOSSIM for - Mote-mote radio communication - Mote-PC serial communication		
4	Create and simulate a simple adhoc network		
5	Understanding, Reading and Analyzing Routing Table of a network.		
6	Create a basic MANET implementation simulation for Packet animation and Packet Trace		
7	Implement a Wireless sensor network simulation.		
8	Create MAC protocol simulation implementation for wireless sensor Network.		
9	Simulate Mobile Adhoc Network with Directional Antenna		
10	Create a mobile network using Cell Tower, Central Office Server, Web browser and Web Server. Simulate connection between them		

Course Code	Course Title	Credits	Lectures /Week
USCS6032	**Information Retrieval**	**2**	**3**

About the Course: This course introduces the principles, techniques, and technologies underlying information retrieval (IR) systems. Students will explore the fundamental concepts of document indexing, storage, and retrieval, as well as advanced topics such as retrieval models, text categorization, web information retrieval, and evaluation techniques. Through a combination of theoretical study, practical exercises, and reference to industry-standard books, students will gain a solid foundation in the field of information retrieval.

Course Objectives:
- To understand the fundamental principles and components of information retrieval systems.
- To explore various techniques for document indexing, storage, and retrieval.
- To analyze and compare different retrieval models and understand their strengths and limitations.
- To gain practical experience in implementing and evaluating information retrieval systems.
- To explore advanced topics in information retrieval, such as web information retrieval and machine learning techniques.

Learning Outcomes:

After successful completion of this course, students would be able to
- Explain the key components and principles of information retrieval systems.
- Apply indexing, storage, and retrieval techniques to efficiently retrieve relevant documents.
- Compare and contrast different retrieval models and select appropriate models for specific search scenarios.
- Develop practical skills in implementing and evaluating information retrieval systems.
- Demonstrate an understanding of advanced topics in information retrieval, including web search and machine learning techniques.

Unit	Topics	No of Lectures
I	**Foundations of Information Retrieval** **Introduction to Information Retrieval (IR) systems:** Definition and goals of information retrieval, Components of an IR system, Challenges and applications of IR **Document Indexing, Storage, and Compression:** Inverted index construction and compression techniques, Document representation and term weighting, Storage and retrieval of indexed documents, **Retrieval Models:** Boolean model: Boolean operators, query processing, Vector space model: TF-IDF, cosine similarity, query-document matching, Probabilistic model: Bayesian retrieval, relevance feedback **Spelling Correction in IR Systems**: Challenges of spelling errors in queries and documents, Edit distance and string similarity measures, Techniques for spelling correction in IR systems	**15**

	Performance Evaluation: Evaluation metrics: precision, recall, F-measure, average precision, Test collections and relevance judgments, Experimental design and significance testing		
II	**Advanced Topics in Information Retrieval** **Text Categorization and Filtering:** Text classification algorithms: Naive Bayes, Support Vector Machines, Feature selection and dimensionality reduction, Applications of text categorization and filtering **Text Clustering for Information Retrieval:** Clustering techniques: K-means, hierarchical clustering, Evaluation of clustering results, Clustering for query expansion and result grouping **Web Information Retrieval:** Web search architecture and challenges, Crawling and indexing web pages, Link analysis and PageRank algorithm **Learning to Rank:** Algorithms and Techniques, Supervised learning for ranking: RankSVM, RankBoost, Pairwise and listwise learning to rank approaches Evaluation metrics for learning to rank **Link Analysis and its Role in IR Systems:** Web graph representation and link analysis algorithms, HITS and PageRank algorithms, Applications of link analysis in IR systems	**15**	
III	**Advanced Topics in Information Retrieval** **Crawling and Near-Duplicate Page Detection:** Web page crawling techniques: breadth-first, depth-first, focused crawling, Near-duplicate page detection algorithms, Handling dynamic web content during crawling **Advanced Topics in IR:** Text Summarization: extractive and abstractive methods, Question Answering: approaches for finding precise answers, Recommender Systems: collaborative filtering, content-based filtering **Cross-Lingual and Multilingual Retrieval:** Challenges and techniques for cross-lingual retrieval, Machine translation for IR, Multilingual document representations and query translation, Evaluation Techniques for IR Systems **User-based evaluation:** user studies, surveys, Test collections and benchmarking, Online evaluation methods: A/B testing, interleaving experiments	**15**	

Textbook(s):
1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto, —Modern Information Retrieval: The Concepts and Technology behind Search, Second Edition, ACM Press Books
2. C. Manning, P. Raghavan, and H. Schütze, —Introduction to Information Retrieval, Cambridge University Press

Additional Reference(s):
1. Ricci, F, Rokach, L. Shapira, B. Kantor, —Recommender Systems Handbook‖, First Edition.
2. Bruce Croft, Donald Metzler, and Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson Education.
3. Stefan Buttcher, Charlie Clarke, Gordon Cormack, Information Retrieval: Implementing and Evaluating Search Engines, MIT Press.

Course Code	Course Title	Credits	Lectures /Week
USCSP6032	**Information Retrieval – Practical**	**1**	**3**

1	Document Indexing and Retrieval • Implement an inverted index construction algorithm. • Build a simple document retrieval system using the constructed index.
2	Retrieval Models • Implement the Boolean retrieval model and process queries. • Implement the vector space model with TF-IDF weighting and cosine similarity.
3	Spelling Correction in IR Systems • Develop a spelling correction module using edit distance algorithms. • Integrate the spelling correction module into an information retrieval system.
4	Evaluation Metrics for IR Systems • Calculate precision, recall, and F-measure for a given set of retrieval results. • Use an evaluation toolkit to measure average precision and other evaluation metrics.
5	Text Categorization • Implement a text classification algorithm (e.g., Naive Bayes or Support Vector Machines). • Train the classifier on a labelled dataset and evaluate its performance.
6	Clustering for Information Retrieval • Implement a clustering algorithm (e.g., K-means or hierarchical clustering). • Apply the clustering algorithm to a set of documents and evaluate the clustering results.
7	Web Crawling and Indexing • Develop a web crawler to fetch and index web pages. • Handle challenges such as robots.txt, dynamic content, and crawling delays.
8	Link Analysis and PageRank • Implement the PageRank algorithm to rank web pages based on link analysis. • Apply the PageRank algorithm to a small web graph and analyze the results.
9	Learning to Rank • Implement a learning to rank algorithm (e.g., RankSVM or RankBoost). • Train the ranking model using labelled data and evaluate its effectiveness.
10	Advanced Topics in Information Retrieval • Implement a text summarization algorithm (e.g., extractive or abstractive). • Build a question-answering system using techniques such as information extraction

Course Code	Course Title	Credits	Lectures /Week
USCS6041	**Data Mining & Warehousing**	**2**	**3**

About the Course: The course covers data warehousing, data mining, association rule mining, classification and prediction, and clustering. Students will learn about OLAP and OLTP, multi-dimensional data models, measures, concept of hierarchy, and data warehouse architecture. They will also explore different data mining functionalities. The course covers classification methods, prediction techniques, and classifier accuracy assessment. Students will learn various clustering methods and their applications to different data types, such as time-series data, text databases, and web mining.

Course Objectives:
- Understand the concept and framework of data warehousing and differentiate between OLAP and OLTP.
- Gain knowledge of data mining techniques and their applications in knowledge discovery.
- Acquire skills in data preprocessing, including handling missing data, cleaning, integration, and transformation.
- Apply association rules mining algorithms such as APRIORI and FP-Growth to discover frequent item sets.

Learning Outcomes:
After successful completion of this course, students would be able to
- Explain the purpose and components of a data warehouse and differentiate it from transactional databases.
- Perform OLAP operations on a multidimensional data model to analyze and query data.
- Implement data preprocessing techniques to address missing data and prepare the data for mining.
- Apply association rules mining algorithms to discover patterns and relationships in large datasets.

Unit	Topics	No of Lectures
I	**Introduction to Data Warehouse:** Introduction, Necessity, Framework of the data warehouse, options, developing data warehouses, Differences between OLAP and OLTP, OLAP Operations in the Multidimensional Data Model, Back-End Tools and Utilities, Metadata Repository, Types of OLAP servers. **DW Design Consideration And Dimensional Modeling:** Defining Dimensional Model, Granularity of Facts, Additivity of Facts, Functional dependency of the Data, Helper Tables, Implementation many-to-many relationships between fact and dimensional modeling. **Data Warehouse Models:** Enterprise Data Warehouse (EDW), Data Mart, Virtual Data Warehouse, Hybrid Data Warehouse.	**15**
II	**Data Mining:** Introduction to Data Mining, Definition, Knowledge Discovery in Data (KDD), Kinds of databases, Data to be mined, Basic mining techniques, Data Mining Issues, Data Mining Metrics, Social	**15**

		Implications of Data Mining, Overview of Applications of Data Mining.	
		Data Preprocessing: Data Processing prerequisites, Attributes and Data types, Statistical descriptions of data, Distance and similarity measures, Need for Preprocessing, Handling Missing data, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization.	
		Association Rules Mining: Problem Definition, Frequent item set generation, The APRIORI Principle, Support and confidence measures, Association rule generation: APRIORI algorithm, FP-Growth Algorithms, Compact Representation of Frequent item Set: Maximal Frequent item set, closed frequent item set.	
III		**Classification And Prediction:** Definition of classification, Model construction, Model Usage, Choosing algorithm, Decision tree Induction, Information gain, gain ratio, gini index, Bayesian Classification, Bayes Theorem, Naïve Bayes classifier, Linear Regression, Non-linear Regression, Logistic Regression. **Validating Model:** Measuring performance of classifiers, Precision, Recall, F-measure, confusion matrix, cross-validation, Bootstrap**.** **Clustering:** Types of data, Categorization of major clustering methods, K-means partitioning methods, Hierarchical methods, Density-based methods, Grid-based methods, Model-based clustering methods, Outlier analysis, Mining Time-Series and Sequence Data, Mining Text Databases, Mining the World Wide Web.	15

Textbook(s):

1. Data Warehousing : Design, Development And Best Practices by Soumendra Mohanty (Author), Tata McGraw Hill Education (Publisher).
2. Jiawei Han, Michelin Kamber, "Data Mining-Concepts and techniques", Morgan Kaufmann Publishers, Elsevier, 3nd Edition.
3. Alex Berson, Stephen J.Smith, "Data warehousing Data mining and OLAP", Tata McGraw-Hill, 2nd Edition.

Additional Reference(s):

1. Arum K Pujari, "Data Mining Techniques", 3rd Edition, Universities Press, 2005
2. PualrajPonnaiah, Wiley, "Data Warehousing Fundamentals", Student Edition, 2004.
3. Ralph Kimball, Wiley, "The Data warehouse Life Cycle Toolkit", Student Edition, 2006.

Course Code	Course Title	Credits	Lectures /Week
USCSP6041	**Data Mining & Warehousing – Practical**	**1**	**3**
1	Perform different operations of extraction, transformation, and loading (ETL) processes on a sample dataset using PowerBI.		
2	Integrate data from multiple sources by merging and transforming datasets using Python's pandas library and data manipulation techniques.		
3	Apply feature selection techniques like variance thresholding and correlation analysis using Python's **scikit-learn library** to reduce dimensionality in a dataset.		
4	**Discretize** continuous variables and create concept hierarchies for categorical variables in a market basket dataset using Python's pandas library.		
5	Implement the **Apriori algorithm** in Python to mine frequent itemsets from a retail transaction dataset and extract association rules.		
6	Build a **decision tree classifier** using Python's scikit-learn library to predict customer churn based on historical data.		
7	Implement a **Naive Bayes classifier in** Python using scikit-learn to classify emails as spam or non-spam based on their content.		
8	Implement a **linear regression** method to make predictions based on the sample data set using Python.		
9	Implement a **logistic regression** method to make predictions based on the sample data set using Python.		
10	Implement **K-means clustering** algorithm in Python using scikit-learn to group customers based on their purchasing behavior.		

Course Code	Course Title	Credits	Lectures /Week
USCS6042	**Ethical Hacking**	**2**	**3**

About the Course: This course provides an in-depth exploration of ethical hacking and penetration testing methodologies. Students will learn about hacking technology types, the phases of ethical hacking, footprinting, social engineering, system hacking, web server and application vulnerabilities, wireless hacking, and more. The course emphasizes hands-on lab exercises and real-world scenarios to develop practical skills in identifying and mitigating security vulnerabilities.

Course Objectives:
- Understand the terminology and concepts related to ethical hacking and penetration testing.
- Explore various hacking technologies and the skills required to become an ethical hacker.
- Learn the different phases involved in ethical hacking and the methodologies used in penetration testing.
- Gain knowledge of common hacking techniques, such as footprinting, scanning, enumeration, and session hijacking.
- Develop proficiency in identifying and exploiting vulnerabilities in web servers, web applications, and wireless networks.

Learning Outcomes:
After successful completion of this course, students would be able to
- Apply ethical hacking methodologies to conduct comprehensive security assessments and penetration tests.
- Perform effective footprinting and reconnaissance techniques to gather critical information about target systems.
- Identify and exploit vulnerabilities in various network and system components using appropriate tools and techniques.
- Evaluate the security posture of web servers, web applications, and wireless networks, and recommend appropriate countermeasures.
- Demonstrate an understanding of ethical and legal considerations in conducting ethical hacking activities and adhere to professional codes of conduct.

Unit	Topics	No of Lectures
I	**Introduction:** Terminology, Hacking Technology Types, Ethical Hacking Phases, Hacktivism, Hacker Classes, Skills Required for an Ethical Hacker, Vulnerability Research, Ways to Conduct Ethical Hacking **Footprinting:** Definition, ~~Information Gathering Methodology, Competitive Intelligence~~, DNS Enumeration, Whois and ARIN Lookups, Types of DNS Records, Traceroute in Footprinting, E-Mail Tracking **Social Engineering:** Common Types Of Attacks **Scanning and Enumeration:** Port Scanning, Network Scanning, Vulnerability Scanning, CEH Scanning Methodology, Ping Sweep Techniques, Nmap Command Switches, SYN, Stealth, XMAS, NULL, IDLE, FIN Scans, Anonymizers, HTTP Tunneling Techniques, IP Spoofing Techniques, SNMP Enumeration, Steps Involved in Enumeration	**15**

II	**System Hacking:** Password-Cracking Techniques, Types of Passwords, Keyloggers and Other Spyware Technologies, Escalating Privileges, Rootkits **Sniffers:** Protocols Susceptible to Sniffing, Active and Passive Sniffing, ARP Poisoning, MAC Flooding, DNS Spoofing Techniques, Sniffing Countermeasures **Denial of Service:** Types of DoS Attacks, Working of DoS Attacks, BOTs/BOTNETs, "Smurf" Attack, "SYN" Flooding, DoS/DDoS Countermeasures **Session Hijacking:** Spoofing vs. Hijacking, Types, Sequence Prediction, Steps, Prevention **Hacking Web Servers:** Web Server Vulnerabilities, Attacks against Web Servers, Patch Management Techniques, Web Server Hardening	**15**	
III	**Web Application Vulnerabilities:** Web Application Hacking, Web Application Threats, Google Hacking, Countermeasures **Web-Based Password Cracking Techniques:** Authentication Types, Password Crackers, Countermeasures **SQL Injection:** Steps, SQL Server Vulnerabilities, Countermeasures **Buffer Overflows:** Types, Stack-Based Buffer Overflows, Mutation Techniques **Wireless Hacking:** WEP, WPA Authentication Mechanisms, and Cracking Techniques, Wireless Sniffers, Rogue Access Points, Wireless Hacking Techniques, Securing Wireless Networks **Penetration Testing Methodologies:** Methodologies, Steps, Automated Tools, Pen-Test Deliverables	**15**	

Textbook(s):
1. CEH official Certfied Ethical Hacking Review Guide, Wiley India Edition

Additional Reference(s):
1. Certified Ethical Hacker: Michael Gregg, Pearson Education
2. Certified Ethical Hacker: Matt Walker, TMH.

Course Code	Course Title	Credits	Lectures /Week
USCSP6042	**Ethical Hacking - Practical**	**1**	**3**
1	Google and Whois Reconnaissance • Use Google search techniques to gather information about a specific target or organization. • Utilize advanced search operators to refine search results and access hidden information. • Perform Whois lookups to retrieve domain registration information and gather details about the target's infrastructure.		

2	Password Encryption and Cracking with CrypTool and Cain and Abel
	• Password Encryption and Decryption:
	  o Use CrypTool to encrypt passwords using the RC4 algorithm.
	  o Decrypt the encrypted passwords and verify the original values.
	• Password Cracking and Wireless Network Password Decoding:
	  o Use Cain and Abel to perform a dictionary attack on Windows account passwords.
	  o Decode wireless network passwords using Cain and Abel's capabilities.		

3	Linux Network Analysis and ARP Poisoning
	• Linux Network Analysis:
	  o Execute the ifconfig command to retrieve network interface information.
	  o Use the ping command to test network connectivity and analyze the output.
	  o Analyze the netstat command output to view active network connections.
	  o Perform a traceroute to trace the route packets take to reach a target host.
	• ARP Poisoning:
	  o Use ARP poisoning techniques to redirect network traffic on a Windows system.
	  o Analyze the effects of ARP poisoning on network communication and security.		

4	Port Scanning with NMap • Use NMap to perform an ACK scan to determine if a port is filtered, unfiltered, or open. • Perform SYN, FIN, NULL, and XMAS scans to identify open ports and their characteristics. • Analyze the scan results to gather information about the target system's network services.		

5	Network Traffic Capture and DoS Attack with Wireshark and Nemesy
	• Network Traffic Capture:
	  o Use Wireshark to capture network traffic on a specific network interface.
	  o Analyze the captured packets to extract relevant information and		

	identify potential security issues.
	• Denial of Service (DoS) Attack:
	o Use Nemesy to launch a DoS attack against a target system or network.
	o Observe the impact of the attack on the target's availability and performance.

6	**Persistent Cross-Site Scripting Attack** • Set up a vulnerable web application that is susceptible to persistent XSS attacks. • Craft a malicious script to exploit the XSS vulnerability and execute arbitrary code. • Observe the consequences of the attack and understand the potential risks associated with XSS vulnerabilities.
7	**Session Impersonation with Firefox and Tamper Data** • Install and configure the Tamper Data add-on in Firefox. • Intercept and modify HTTP requests to impersonate a user's session. • Understand the impact of session impersonation and the importance of session management.
8	**SQL Injection Attack** • Identify a web application vulnerable to SQL injection. • Craft and execute SQL injection queries to exploit the vulnerability. • Extract sensitive information or manipulate the database through the SQL injection attack.
9	**Creating a Keylogger with Python** • Write a Python script that captures and logs keystrokes from a target system. • Execute the keylogger script and observe the logged keystrokes. • Understand the potential security risks associated with keyloggers and the importance of protecting against them.
10	**Exploiting with Metasploit (Kali Linux)** • Identify a vulnerable system and exploit it using Metasploit modules. • Gain unauthorized access to the target system and execute commands or extract information. • Understand the ethical considerations and legal implications of using Metasploit for penetration testing.

Course Code	Course Title	Credits	Lectures /Week
USCS6051	**Customer Relationship Management**	2	3

About the Course: This course on Customer Relationship Management (CRM) provides an in-depth understanding of the principles, strategies, and tools necessary for managing customer relationships effectively. Students will explore the various forms of CRM and its impact on business performance. Additionally, the course covers customer acquisition, retention, and the measurement of customer-perceived value. Students will also gain insights into strategic and operational CRM, including customer portfolio management, marketing automation, and service automation. The course concludes with an examination of analytical CRM and the implementation of CRM strategies through real-life case studies.

Course Objectives:
- To provide students with a comprehensive understanding of CRM concepts, theories, and models.
- To equip students with the knowledge and skills to manage the customer journey, including customer acquisition and retention.
- To explore the factors that contribute to customer-perceived value and the role of CRM in enhancing the customer experience.
- To familiarize students with strategic and operational CRM approaches, including customer portfolio management and marketing automation.
- To introduce students to the analytical aspects of CRM, including data management, analytics for strategy and tactics, and the implementation of CRM systems.

Learning Outcomes:
After successful completion of this course, students would be able to
- Students will be able to define and explain the various forms of CRM and their relevance to business contexts.
- Students will acquire the skills to manage the customer journey effectively, including implementing customer acquisition and retention programs.
- Students will understand the importance of customer-perceived value and its impact on customer satisfaction, loyalty, and business performance.
- Students will be able to apply strategic and operational CRM approaches, such as customer portfolio management and marketing automation, to enhance organizational effectiveness.
- Students will develop proficiency in analytical CRM techniques, including data management, analytics for strategy and tactics, and the successful implementation of CRM systems. They will also be able to analyze and draw insights from real-life case studies and success stories related to CRM.

Unit	Topics	No of Lectures
I	**Understanding Customer Relationships** **Introduction to CRM:** Three forms of CRM, The changing face of CRM, Misunderstandings about CRM, Defining CRM, CRM constituencies, Commercial contexts of CRM, Models of CRM, Understanding	15

	relationships, Relationship Quality, Customer life-time value, Customer satisfaction, loyalty and business performance, Relationship management theories, Benefits of CRM	
	Managing the customer journey: customer acquisition, Portfolio purchasing, Prospecting, Key performance indicators of customer acquisition programs, Operational CRM tools that help customer acquisition, Customer retention, Economics and Strategies of customer retention, Key performance indicators of customer retention programs.	
	Managing customer-experienced value: Understanding value, modeling customer-perceived value, its sources, Customization, Value through the marketing mix, Customer Experience concepts, Service marketing, Total quality management, relationship management, CRM's influence on CX, How CRM software applications influence customer experience	
II	**Strategic and Operational CRM** **Customer portfolio management:** Portfolio, customer, Basic disciplines for CPM, CPM in the business-to-business context, customer portfolio management tools, strategically significant customers, The seven core customer management strategies **Marketing automation:** Introduction to marketing automation, Benefits, Software applications for marketing, Sales force automation **Service automation:** Introduction, customer service, Modeling service quality, Customer service standards, service automation, Benefits, Software applications for service	15
III	**Analytical CRM** **Developing and managing customer-related databases:** Corporate customer-related data, Structured and unstructured data, Developing a customer-related database, Data integration, Data warehousing, Data marts, Analytics for CRM strategy and tactics, Big data analytics, Analytics for structured data, ways to generate analytical insight, Data-mining procedures, Artificial intelligence (AI), machine learning (ML) and deep learning (DL) **Implementing CRM:** Introduction, develop the CRM strategy, build CRM project foundations, needs specification and partner selection, project implementation, performance evaluation **Case studies and success stories related to CRM**	15

Textbook(s):
1. Customer Relationship Management Concepts and Technologies, Francis Buttle, Stan Maklan, Routledge Taylor and Francis Group, 2019
2. Jagdish N Sheth, Parvatiyar Atul, G Shainesh, Customer Relationship Management- Emerging Concepts, Tools and Applications, 2017
3. Anderrson Kristin , Carol Kerr, Customer Relationship Management, Tata McGraw-Hill, 2017

Additional Reference(s):
1. V. Kumar & Werner J., CUSTOMER RELATIONSHIP MANAGEMENT, Willey India, 2008
2. S. Shanmugasundaram, CUSTOMER RELATIONSHIP MANAGEMENT, Prentice Hall of

Course Code	Course Title	Credits	Lectures /Week

India Private Limited, New Delhi, 2008

Course Code	Course Title	Credits	Lectures /Week
USCS6052	**Cyber Laws and IPR**	**2**	**3**

About the Course: The course on Cyber Laws and IPR provides a comprehensive understanding of the legal aspects and regulations related to cyberspace and information technology. The course covers a wide range of topics, including basic concepts, internet technology, network security, cyber law, e-commerce, electronic signatures, cyber crimes, privacy, intellectual property rights, and more. Students will explore the legal framework governing cyberspace and develop an understanding of the legal and ethical issues associated with information technology.

Course Objectives:
- Understand fundamental concepts of cyber laws and their relevance in the digital age.
- Examine legal frameworks and regulations in cyber laws, including the Information Technology Act 2000 in India.
- Explore key issues in cyber laws such as e-commerce, e-governance, and electronic records and contracts.
- Gain knowledge of cybercrimes, enforcement mechanisms, and the role of the Cyber Appellate Tribunal.
- Analyze emerging issues in cyber laws, including liability of ISPs, privacy concerns, and jurisdictional aspects.
- Understand intellectual property rights and online regulations, including copyrights, patents, and domain name disputes.

Learning Outcomes:
After successful completion of this course, students would be able to
- Demonstrate a comprehensive understanding of cyber laws and their application in the digital age.
- Evaluate legal frameworks and regulations governing cyber laws.
- Identify and assess key issues in cyber laws, such as e-commerce, e-governance, and electronic records and contracts.
- Understand cyber crimes, enforcement mechanisms, and the role of the Cyber Appellate Tribunal.
- Analyze emerging issues in cyber laws, including liability of ISPs, privacy concerns, and jurisdictional complexities.
- Recognize intellectual property rights and online regulations, including copyrights, patents, and domain name disputes.

Unit	Topics	No of Lectures
I	**Introduction to Cyber Laws and Technology:** Basic Concepts, Internet and Advantages and Disadvantages of Internet Technology, Network and Network Security **Legal Framework and Regulations:** Cyber Law & Components of Cyber Law, Cyber Law in India: An Overview of Information Technology Act	15

	2000, Cryptography, Encryption Technique & Algorithm and Digital Signature & Electronic Signature		
	Key Issues in Cyber Laws: E-Commerce, E-Governance, E-Record & E-Contract, Regulator, Certifying Authority, Electronic Signature Certificates		
II	**Cyber Crimes and Enforcement:** Cyber Appellate Tribunal, Cyber Crimes-Cyber Contraventions, Cyber Offences, Power of Investigation & Search, E-Evidence and Computer Forensic	15	
	Emerging Issues and Legal Considerations: ISP & Intermediary Not to be Liable in Certain Cases, Consequential Amendments in Various Conventional Laws in India, Grey Areas of Information Technology Act, 2000,		
	Jurisdiction and Privacy: Cyber Jurisdiction, E-Consumers, Privacy of Online Data and Information		
III	**Intellectual Property Rights and Online Regulations:** Free Speech Online or Online Freedom of Speech and Expression and Liability of Intermediary	15	
	Intellectual Property Rights (IPRs), Copyrights & Patents: International and Indian Scenario, Copyright Issues and Digital Medium, Patent Issues in Digital Medium		
	Disputes and Resolution: Domain Name Dispute & Resolution and Trademark Issues in Digital Medium, Spamming and Phishing		

Textbook(s):
1. Cyber Laws & Information Technology (For LL.B.) Paperback – 1 January 2020
2. Cyber Law in India, Satish Chandra, ABS Books, 2017
3. Cyber Security and Cyber Laws, Nilakshi Jain, Wiley India, October 2020

Additional Reference(s):
1. Cyber Laws, Justice Yatindra Singh, Universal Law Publishing, Universal Publishing, 2016
2. Cyber laws, Dr. Gupta & Agrawal, PREMIER PUBLISHING COMPANY, 2022
3. Cyber Law - An Exhaustive Section Wise Commentary On The Information Technology, Pavan Duggal, Universal Publishing (LexisNexis), 2nd Edition, 2017

Course Code	Course Title	Credits	Lectures /Week
USCSP605	**Project Work – II**	**2**	**3**
Refer to the Project Guidelines mentioned at the end			

Evaluation Scheme

I.   Internal Evaluation for Theory Courses – 25 Marks

(i) Mid-Term Class Test – 15 Marks

- It should be conducted using any **learning management system** such as **Moodle** (Modular object-oriented dynamic learning environment)
- The test should have **15 MCQ's** which should be solved in a time duration of **30 minutes.**

(ii) Assignment/ Case study/ Presentations – 10 Marks

- Assignment / Case Study Report / Presentation can be uploaded on any **learning management system**.

II.   External Examination for Theory Courses – 75 Marks
- Duration: **2.5 Hours**
- Theory question paper pattern:

	All questions are compulsory.		
Question	**Based on**	**Options**	**Marks**
Q.1	Unit I	*Any 4 out of 6*	20
Q.2	Unit II	*Any 4 out of 6*	20
Q.3	Unit III	*Any 4 out of 6*	20
Q.4	Unit I, II and III	*Any 5 out of 6*	15

- All questions shall be compulsory with internal choice within the questions.
- Each Question may be sub-divided into sub questions as a, b, c, d, etc. & the allocation of Marks depends on the weightage of the topic.

III.   Practical Examination
- Each subject carries 50 Marks

  40 marks + 05 marks (journal) + 05 marks (viva)
- Duration: **2 Hours** for each practical course.
- Minimum **80% practical** from each core subjects are required to be completed.
- **Certified Journal is compulsory for appearing at the time of Practical Exam**
- The final submission and evaluation of **journal in electronic form** using a Learning Management System / Platform can be promoted by college.

IV.   Project

The evaluation of the project will include a viva voce, which will assess the project based on the following parameters:

- **Documentation – 30 Marks:** The completeness, accuracy, and professionalism of the project documentation, including the project report and supporting materials, will be considered.

- **Quality of the Project – 15 Marks:** The overall quality of the project, including its design, implementation, and user experience, will be evaluated.

- Working of the Project – 20 Marks: The functionality and performance of the project will be assessed to determine how well it meets the specified requirements and objectives.

- **Project Presentation – 15 Marks:** The clarity, organization, and effectiveness of the project presentation will be evaluated.

- **Viva – 20 Marks:** The viva voce session will provide an opportunity for the student to demonstrate their knowledge and understanding of the project, as well as to answer questions and engage in a discussion with the evaluators.

Project Guidelines (*for USCSP505 and USCSP605*)

Aim:

The Project Work as part of B.Sc. Computer Science program provides students with practical experience in applying their knowledge and skills to real-world projects, emphasizing hands-on experience in industry-standard project practices. It focuses on project development, implementation, and deployment using computer science principles and techniques. Students will work individually or in teams to design, develop, and present a substantial software project, gaining exposure to real-life project scenarios. It also covers project planning, requirements gathering, software design, coding, testing, debugging, documentation, and project management, following industry best practices. Through these projects, students will enhance their problem-solving abilities, gain proficiency in software development methodologies, and strengthen their practical skills in computer science.

Objectives:

- Apply interdisciplinary knowledge to effectively solve real-life problems using acquired skills and concepts.
- Gain hands-on experience in the software development life cycle, encompassing requirements analysis, design, implementation, testing, and deployment.
- Familiarize with global IT industry standards, ethics, and professional practices to thrive in a professional environment.
- Develop teamwork and project management skills through structured collaboration, effective communication, and task delegation.
- Produce professional technical documentation aligning with industry practices, ensuring clarity, accuracy, and usability.
- Acquire time management, resource allocation, and personnel coordination skills for efficient project execution.

Project Types:

a) **Developing a solution for a real-life problem:** In this case, the project focuses on addressing an existing requirement for a computer-based solution that has practical applications. The project should successfully implement the different stages of the system development life cycle. Examples: Secure Online Banking System, Machine Learning-based Disease Diagnosis System, Cloud-based Document Management System.

b) **Innovative Product Development:** These projects involve exploring and developing a computer-based solution with a unique and innovative utility. Examples: Cybersecurity Monitoring and Threat Detection System, Machine Learning-powered Predictive Maintenance System for Industrial Equipment, IoT-based Smart Energy Management System.

c) **Research-Level Project:** These projects involve conducting research and development to explore advanced technologies and solve complex problems. Examples: Deep Learning-based Image Recognition System for Medical Imaging, Cloud Computing Infrastructure Optimization for Big Data Processing, Data Science-driven Predictive Analytics for Sales Forecasting. The methodology and reporting of such projects may vary based on the project supervisor's guidance.

Tools & Technologies:

In the project work, students are granted complete freedom to select platforms, tools, and programming languages without any imposed restrictions. This approach encourages creativity, flexibility, and exploration of various technologies. By prioritizing open-source technologies, students can leverage a vast array of resources and community support. Commonly employed tools include IDEs, version control systems (e.g., Git), programming languages (e.g., Python, Java), databases (e.g., MySQL), and web frameworks (e.g., Django, Ruby on Rails). The evaluation process focuses on the project's content and implementation rather than the specific tools chosen, ensuring a fair assessment of the students' skills and problem-solving abilities.

Project Guide:

Assigning a project guide to each project or group is a mandatory requirement to ensure the successful completion of the project work. The guide plays a crucial role as a mentor and technical expert, providing invaluable support and guidance to students. They are expected to facilitate effective communication and teamwork, review project proposals, assign schedules, and monitor progress on a regular basis. Additionally, guides are expected to offer timely feedback, provide guidance on project planning and implementation strategies, evaluate the quality of work, and promote professionalism and ethical conduct. Their expertise and involvement are essential in helping students navigate challenges, make informed decisions, and achieve their project goals effectively.

Project Team Size: 1 – 2 members

Project Proposal: The project proposal is a mandatory document that serves as a foundation for the project. It helps students define their project idea, receive early evaluation and feedback, establish clear communication with the project guide, and take ownership of the project's successful execution. A formal proposal ensures systematic and professional project planning, fostering critical thinking, effective communication, and project management skills. The proposal provides a roadmap and increases the chances of a successful outcome. Before initiating a project, it is mandatory to submit a project proposal for approval. **The original duly approved project proposal should be attached to the final project report.** The project proposal for UG computer science projects should include the following contents:
- Title
- Introduction
- Objectives: Clearly state the objectives of the project. What specific goals do you aim to achieve?
- Scope
- Methodology
- Tools and Technologies
- Timeline
- Resources
- Expected Outcomes
- References

Project Report:

The Certified Copy of Hard Bound Project Report must adhere to the following guidelines:

- No of Copies: Team Size + 1 (College / Department Copy)
- The project report should include the following
    - Title Page *(Sample attached in Appendix)*
    - Certificate *(Sample attached in Appendix)*
    - Declaration *(Sample attached in Appendix)*
    - Acknowledgement
    - Table of Contents
    - Original Copy of approved Project Proposal
    - Self-attested copy of Plagiarism Report from any open source tool.
    - Chapters / Sections depending upon the type of project
    - List of Tables and/or List of Figures
    - References (IEEE / Springer format)
    - Glossary
    - Appendices (Survey datasheets / Questionnaires, ect)
- Use of LaTeX for documentation purposes should be preferred.
- The text of the report should be set in 12 pt, Times New Roman font, and single-spaced.
- Chapter headings should be centered, written in 20 pt, Times New Roman font, bold, and in all caps.
- These guidelines ensure a standardized format for the project report, promoting clarity and readability.

<u>**SAMPLE TITLE PAGE FORMAT**</u>

A PROJECT REPORT

on

\<PROJECT NAME\>

Submitted by

Mr. XYZ

in partial fulfillment for the award of the degree

of

BACHELOR OF SCIENCE

in

COMPUTER SCIENCE

under the guidance of

\<Guide Name\>

Department of Computer Science

\<\<College Logo\>\>

\<\<College Name\>\>

(Sem V / VI)

(202- – 202-)

SAMPLE CERTIFICATE FORMAT

<<College Logo>>

<<College Name>>,

<<College Address>>

Department of Computer Science

CERTIFICATE

This is to certify that Mr./Ms. _____ of **T.Y.B.Sc. (Sem V/VI)** class has satisfactorily completed the Project _____, to be submitted in the partial fulfillment for the award of **Bachelor of Science** in **Computer Science** during the academic year **202- – 202-**.

Date of Submission:

Project Guide   **Head / Incharge,**
                                                          Department Computer Science
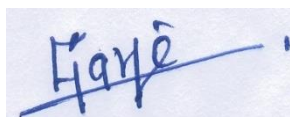
College Seal   **Signature of   Examiner**

DECLARATION

I, _____, hereby declare that the project entitled "_____" submitted in the partial fulfillment for the award of **Bachelor of Science** in **Computer Science** during the academic year **202- – 202-** is my original work and the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Signature of the Student:

Place:

Date

Prof. Shivram S. Garje,
Dean,
Faculty of Science and Technology