UNIVERSITY OF MUMBAI

Syllabus for

Program: Bachelor of Science

Course: Computer Science

with effect from

Academic Year 2023-2024

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**	15

	Knowledge Representation and Reasoning: Knowledge Representation and different forms, Reasoning, Planning, Uncertainty in Knowledge Fuzzy 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	**15**

	details not expected), Multiple Encryption and Triple DES, Block Cipher Modes of Operation, Stream Ciphers **Public-Key Cryptography and RSA:** Principles of Public-Key Cryptosystems, The RSA Algorithm	
II	**Key Management:** Public-Key Cryptosystems, Key Management, Diffie-Hellman Key Exchange **Message Authentication and Hash Functions:** Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs, Secure Hash Algorithm, HMAC **Digital Signatures and Authentication:** Digital Signatures, Authentication Protocols, Digital Signature Standard **Authentication Applications:** Kerberos, X.509 Authentication, Public-Key Infrastructure	15
III	**Electronic Mail Security:** Pretty Good Privacy, S/MIME **IP Security:** Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management **Web Security:** Web Security Considerations, Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction **Intrusion:** Intruders, Intrusion Techniques, Intrusion Detection **Malicious Software:** Viruses and Related Threats, Virus Countermeasures, DDOS **Firewalls:** Firewall Design Principles, Types of Firewalls	15

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:** 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.	**15**	

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

	Software Testing Techniques and Strategies	
II	**Testing Fundamentals**: Basics of software testing process, Test case design principles and techniques, Test execution, reporting, and documentation	15
	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	
III	**Defect Management and Software Quality Assurance**	15
	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	

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 **3D Graphics for Game Programming:** 3D Transformations, Quaternions,	15

	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	**15**

	identification and analysis, Scope planning and definition, 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 & Limitations Linear 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
	Refer to the Project Guidelines at the end		