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
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
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
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.		