AC: 02.06.2025 ITEM NO: 2.3

Deccan Education Society's

Kirti M. Doongursee College of Arts, Science and Commerce (AUTONOMOUS)

Affiliated to

UNIVERSITY OF MUMBAI

Syllabus for
Program: Bachelor of Science
Course: T.Y.B.Sc.
Subject: Information Technology

Choice Based Credit System (CBCS)
with effect from
Academic Year 2025-2026

PROGRAM OUTCOMES

PO	Description
A student completing Bachelor's Degree in **Science** Program will be able to	
PO1	Disciplinary Knowledge: Demonstrate comprehensive knowledge of the disciplines that form a part of a graduate Programme. Execute strong theoretical and practical understanding generated from the specific graduate Programme in the area of work.
PO2	Critical Thinking and Problem solving: Exhibit the skills of analysis, inference, interpretation and problem-solving by observing the situation closely and design the solutions
PO3	Social competence: Display the understanding, behavioral skills needed for successful social adaptation, work in groups, exhibit thoughts and ideas effectively in writing and orally.
PO4	Research-related skills and Scientific temper: Develop working knowledge and applications of instrumentation and laboratory techniques. Able to apply skills to design and conduct independent experiments, interpret, establish hypothesis and inquisitiveness towards research
PO5	Trans-disciplinary knowledge: Integrate different disciplines to uplift the domains of cognitive abilities and transcend beyond discipline-specific approaches to address a common problem.
PO6	Personal and professional competence: Performing dependently and collaboratively as a part of a team to meet defined objectives and carry out work across interdisciplinary fields. Execute interpersonal relationships, self-motivation and adaptability skills and commit to professional ethics.
PO7	Effective Citizenship and Ethics: Demonstrate empathetic social concern and equity centered national development, and ability to act with an informed awareness of moral and ethical issues and commit to professional ethics and responsibility.
PO8	Environment and Sustainability: Understand the impact of the scientific solutions in societal and environmental contexts and demonstrate the knowledge of and need for sustainable development.

Year of implementation- 2025-2026
Name of the Department-Information Technology

SEM	Course Code	Course Title	Vertical	Credit
V	25ITMJ511	Enterprise Java	Major 1	2
	25ITMJP51	Enterprise Java Practical	Major 1	2
	25ITMJ512	Data Science	Major 2	2
	25ITMJP52	Data Science Practical	Major 2	2
	25ITMJ513	Research Methodology	Major 3	2
	25ITEL521 25ITEL522 25ITEL523	Go Programming Linux Administration Cyber Security	Elective	2
	25ITELP51 25ITELP52 25ITELP53	Go Programming Practical Linux Administration Practical Cyber Security Practical	Elective	2
	25ITMR531	Internet of Things	Minor	2
	25ITMRP51	Internet of Things Practical	Minor	2
	25ITVS551	AngularJS Practical	VSC	2
	25ITFP57	Project Dissertation	FEP	2
VI	25ITMJ611	Artificial Intelligence	Major 1	2
	25ITMJP61	Artificial Intelligence Practical	Major 1	2
	25ITMJ612	Business Intelligence & Data Analytics	Major 2	2
	25ITMJP62	Business Intelligence & Data Analytics Practical	Major 2	2
	25ITMJ613	Software Testing and Quality Assurance	Major 3	2
	25ITEL621 25ITEL622 25ITEL623	Information Security Mojo Programming IT Infrastructure Management	Elective	2
	25ITELP61 25ITELP62 25ITELP63	Information Security Practical Mojo Programming Practical Advanced Mobile Programming Practical	Elective	2
	25ITMR631	Fundamental of GIS	Minor	2
	25ITMRP61	Fundamental of GIS Practical	Minor	2
	25ITOJT67	Project Implementation	OJT	4

SEMESTER-V

Course Code	MAJOR SEM – V	Credits	Lectures/ Week
25ITMJ511	Paper I – Enterprise Java	2	2

Course Outcomes:
After successful completion of this course, students would be able to
CO1:Identify the key phases of the Maven lifecycle.
CO2:Explain the structure of a basic Maven project.
CO3:Execute common Maven lifecycle phases (clean, compile, package).
CO4:Compare and contrast different Maven build phases.

Unit	Topics	No of Lectures
I	**Maven:** Getting Started with Maven, Setting up Maven, Maven Dependency Management, Maven Project Basics, Maven Life Cycle. **Generics & Lambda Expression:** What are Generics?, Generics Example, Introducing Lambda Expressions. **Collection Framework:** Collections Overview, Collection Interfaces, Collection Classes, Accessing a Collection via an Iterator, Spliterators, Working with Maps, Comparators.	15
II	**Stream API & Regular Expressions:** Basics, Reduction Operations, Using Parallel Streams, Mapping, Collecting, Iterators & Streams, Regular Expression Processing. **Java Beans:** What is a Java Bean?, Advantages of Beans, Introspection, Bound & Constrained Properties, Persistence, Customizers	15

References:

- Java: The Complete Reference Herbert Schildt McGraw Hill 9th 2014
- Introducing Maven, Balaji Varanasi & Sudha Belida, Apress

Course Code	MAJOR SEM-V Practical	Credits	Lectures/ Week
25ITMJP51	Paper I – Enterprise Java Practical	2	4

Course Outcomes:
After successful completion of this course, students would be able to
CO1:Identify the basic components of a Maven project structure (pom.xml, src directory, etc.).
CO2 : Describe how Maven resolves project dependencies.
CO3 :Create a basic Maven project from scratch.
CO4: Troubleshoot dependency conflicts in a Maven project.

1	**Maven Project Setup:** Create a simple Maven project that includes dependencies for JUnit testing and a logging library (e.g., Log4j). Write a basic test case and configure Maven to execute the test during the build process.
2	**Generic Class for Data Storage:** Create a generic class that can store any type of data. Implement methods to add data, retrieve data, and check the type of data stored. Demonstrate its usage with different data types (e.g., Integer, String, Double).
3	**Lambda Expression for Filtering:** Given a list of objects (e.g., a class with name and age), use a lambda expression to filter the list and return a new list containing only the objects that meet a specific condition (e.g., age greater than 18).
4	**Using a Map to Count Word Occurrences:** Write a program that reads a text file, uses a Map to count the occurrences of each unique word, and then prints the word counts.
5	**Sorting a Collection with a Comparator:** Create a class representing a student with attributes like name, ID, and GPA. Create a List of student objects and sort it using a custom Comparator based on different attributes (e.g., sort by GPA, then by name).
6	**Processing a List with Stream API:** o Given a list of strings, use the Stream API to: ● Convert all strings to uppercase. ● Filter out strings that start with a specific letter. ● Sort the remaining strings. ● Collect the result into a new list.
7	**Regular Expression for Email Validation:** Write a regular expression to validate email addresses. Create a program that takes a list of email addresses and uses the regular expression to filter out the invalid ones.
8	**Creating a Simple JavaBean:** Create a simple JavaBean with a few properties (e.g., name, age, city) and appropriate getter and setter methods. Demonstrate how to set and retrieve these properties.
9	**Using Bound Properties in JavaBeans:** Create a JavaBean with a bound property. Implement a listener that listens for changes to this property and prints a message whenever the property changes.
10	**Data Processing Application with Maven, Collections, Stream API, and Regular Expressions:** o Create a Maven project that reads data from a CSV file (e.g., customer data with name, email, phone number, age).

	o Use regular expressions to validate the email and phone number fields. o Store the data in a List of objects. o Use the Stream API to: ● Filter customers based on age. ● Sort the customers by name. ● Extract a list of email addresses. o Print the processed data.

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	MAJOR SEM – V	Credits	Lectures/ Week
25ITMJ512	Paper II – Data Science	2	2

Course Outcomes:
After successful completion of this course, students would be able to
CO1: Recall the data science fundamentals.
CO2: Understand Statistical analysis skills.
CO3: Apply data management proficiency.
CO4: Analyze exploratory data analysis and machine learning awareness and Evaluate tools proficiency

Unit	Topics	No of Lectures
I	**What is Data Science?**: What is Data? Different kinds of data, Data Science Process or lifecycle, Definition and scope of Data Science, Applications and domains of Data Science. **Data Preprocessing:** Descriptive Data Summarization, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization and Concept Hierarchy Generation. Comparison with other fields like Business, Intelligence (BI), Artificial Intelligence (AI), Machine Learning (ML), and Data Warehousing/Data Mining (DW-DM) **Introduction to python data science**, Role of data science in business, Basic terminologies of DS, a. Data science b. Data scientist c. Data set d. Data mining e. Data visualization f. Data modeling g. Data wrangling h. big data i. Machine learning j. Algorithms k. Deep learning Algorithms: Machine Learning Algorithms – Supervised Learning, Unsupervised Learning and Reinforcement Learning, k-Nearest Neighbours (kNN), Simple Linear Regression, Naïve Bayes Algorithm, k-means clustering and Apriori Algorithm.	15
II	**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. **Business application for R and Python**: Web and Dashboard Development, healthcare, operations, finance, marketing and HR	15

References:

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

Additional Reference(s):

- Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly, 2013
- Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication, 2015
- Hands-On Programming with R, Garrett Grolemund,1st Edition, 2014

Course Code	MAJOR SEM-V	Credits	Lectures/Week
25ITMJP52	Data Science Practical	2	4

Course Outcomes:
After successful completion of this course, students would be able to
CO1:Recall and describe the functions of Excel features such as conditional formatting and pivot tables.
CO2:Explain the process of reading structured data into data frames and interpret basic preprocessing tasks.
CO3:Formulate null and alternative hypotheses and conduct hypothesis testing using statistical methods.
CO4: Analyze model performance using metrics such as R^2, RMSE, accuracy, precision, and recall.

1	**Introduction to Excel** ● Perform conditional formatting on a dataset using various criteria. ● Create a pivot table to analyze and summarize data.
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.
3	**Feature Scaling and Dummification** ● Apply feature-scaling techniques like standardization and normalization to numerical features.
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
5	**ANOVA (Analysis of Variance)** ● Perform one-way ANOVA to compare means across multiple groups.
6	**Regression and Its Types** ● Implement simple linear regression using a dataset.
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).
8	**K-Means Clustering** ● Apply the K-Means algorithm to group similar data points into clusters.
9	**Principal Component Analysis (PCA)** ● Perform PCA on a dataset to reduce dimensionality.
10	**Data Visualization and Storytelling** ● Create meaningful visualizations using data visualization tool

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	MAJOR SEM – V	Lectures /Week
25ITMJ513	Paper III – Research Methodology	2

Course Outcomes:

After successful completion of this course, students would be able to

CO1: Identify and recall the various terms associated with research methodology and statistics.

CO2: Describe and explain the various concepts, theories, terminologies and methods used in research and statistics.

CO3: Apply suitable methods of research and statistics to solve practical scenarios and numerical.

CO4: Analyse research related information and research ethics.

Unit	Topics	No of Lectures
I	**Introduction to Research Methodology** 1.1 Meaning and objectives of the research; characteristics; types of research– a) Pure and Fundamental b) Applied, c) Empirical d) Scientific & Social e) Historical f) Exploratory g) Descriptive h) Causal. 1.2 Concepts in Research: Variables, Qualitative and Quantitative Research, Review of Literature. 1.3 Stages in the research process. 1.4 Hypothesis - Meaning, Nature, Significance, Types of Hypotheses, Sources. 1.5 Research design– Meaning, Definition, Need and Importance, Steps in research design, Essentials of a good research design, Areas / Scope of research design and Types-Descriptive, Exploratory and causal. 1.6 Types of data -Primary and Secondary data sources. 1.7 Methods of collection of primary data. a. Observation b. Experimental c. Interview d. Survey 1.8 Methods of collection of secondary data.	15
II	**Advanced Techniques in Research** 1.1 Survey Instrument – Questionnaire Designing – Types of questions, Types of questionnaires. 1.2 Processing of Data – editing, coding and classification – meaning and purpose. 1.3 Analysis of Data – tabulation and graphical presentation – meaning and purpose. 1.4 Interpretation of Data. 1.5 Research Report Writing – Importance and Essentials of good report, content of research report, steps in writing research report, types of reports. 1.6 Citation & Referencing Styles – APA, MLA, Chicago, Harvard styles. Bibliography, Footnote, Endnote, Header and Footer, Appendix. 1.7 Plagiarism and ways to avoid plagiarism. 1.8 Ethical Practices in Research.	15

References:

1. *Research Methodology, C R Kothari, New Age Publication*
2. *Research methodology in Social sciences, O.R.Krishnaswamy, Himalaya Publication*
3. *Business Research Methods, Donald R Cooper, Pamela Schindler, Tata McGraw Hill*

Course Code	MAJOR ELECTIVE SEM – V	Credits	Lectures/ Week
25ITEL521	Paper I – Go Programming	2	2

Course Outcomes:
After successful completion of this course, students would be able to
CO1:List the built-in functions in Go for string manipulation.
CO2:Summarize the differences between pointers and values in Go.
CO3:Develop concurrent programs using Goroutines and channels.
CO4:Debug Go programs to identify and fix errors.

Unit	Topics	No of Lectures
I	**Go Fundamentals and Basic Structures** **Introduction to Go:** History, features, and benefits of Go, Setting up the Go development environment, Basic program structure, Running Go programs, Go tools. **Core Go Concepts:** Variables, declaration, initialization, data types, Constants, Operators, Control flow: if, else if, else, for, switch, defer, panic, and recover. **Composite Data Types:** Arrays, Slices, Maps, Structs and Pointers. **Functions:**Defining and calling functions, Multiple return values, Variadic functions, Anonymous functions and closures. **Methods, Structs and Interfaces:** Defining methods on structs, Interfaces, Empty interface, Type assertions and type switches, Error handling using interfaces.	15
II	**Advanced Go Concepts and Applications** **Concurrency:** Goroutines, go keyword, Channels, Buffered channels, select statement for multiplexing channel operations, Mutexes and WaitGroups for synchronization. **Error Handling:** Returning errors as values, Custom error types, Error wrapping and Logging. **Packages and Modules:** Creating and using packages, Importing standard library and third-party packages, Go Modules for dependency management. **Input/Output and Basic File Handling:** Reading from standard input, Printing to standard output, Basic file operations. **Introduction to Testing:** Writing unit tests with the testing package, Test functions and test cases, Running tests. **Introduction to Web Development:** Basic concepts of HTTP, Using the net/http package, Handling requests and responses, Basic routing.	15

References:

- The Go Programming Language by Alan A.A. Donovan and Brian W. Kernighan.
- Introducing Go: Build Reliable, Scalable Programs by Caleb Doxsey.

Course Code	ELECTIVE SEM-V	Credits	Lectures/Week
25ITELP51	Go Programming Practical	2	4

Course Outcomes:

After successful completion of this course, students would be able to

CO1:Identify the different ways to declare variables in Go.

CO2:Explain the concept of packages and modules in Go programming.

CO3:Use Go's standard library to work with files and input/output.

CO4:Compare and Contrast different Go data structures in terms of performance.

1	**Basic Input/Output and Conversion** a. Write a Go program that takes a temperature in Celsius or Fahrenheit as input, along with the conversion direction (C to F or F to C), and performs the conversion. b. Write a Go program that takes a string as input from the user and writes it to a new text file.
2	**String Manipulation** a. Write a Go function that takes a string as input and returns its reversed version. b. Create a Go function that takes a string as input and returns true if it's a palindrome and false otherwise.
3	**Basic Calculator** a. Create a command-line program that takes two numbers and an operator (+, -, *, /) as input and performs the corresponding calculation. b. Extend the calculator to include more operations (e.g., exponentiation, modulus) and implement robust error handling for invalid inputs, such as division by zero.
4	**File Handling** a. Create a program that reads the content of a text file and prints it to the console line by line. Handle file not found errors. b. Develop a command-line to-do list application that can add tasks, list tasks, and mark tasks as done. Store the to-do items in a file.
5	**Concurrency: Goroutines and WaitGroups** a. Write a program that takes a list of names and greets each name concurrently using goroutines. Use a WaitGroup to ensure all greetings are printed before the program exits. b. Create a program that processes multiple text files concurrently. For each file, count the number of words and lines, and print the results. Use goroutines and channels for efficient processing.
6	**Concurrency: Channels**

	a. Create a function that calculates the sum of a slice of numbers by dividing the slice into two parts, calculating the sum of each part in a separate goroutine, and then combining the results using channels. b. Write a Go program that finds prime numbers up to a given limit using multiple goroutines to speed up the process. Use channels to collect the prime numbers.
7	**Basic Web Server** a. Build a simple HTTP server that listens on a specific port and responds with "Hello, World!" when a user accesses it via a web browser.
8	**JSON Processing** a. Create a Go program that defines a struct representing a person (with name, age, and city). Encode an instance of this struct into JSON format and then decode a given JSON string back into the struct. b. Create a Go program that reads a JSON file containing information about books (title, author, publication year) and prints the details of each book.
9	**String Searching Utility** a. Write a Go program that takes a directory and a search string as input and finds all files in that directory (and its subdirectories) that contain the search string.
10	**Unit Testing** a. Choose one or more of the functions you implemented in the previous exercises (e.g., the string reversal function, the calculator functions, or the palindrome checker) and write a comprehensive set of unit tests for it using the testing package.

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	ELECTIVE SEM – V	Credits	Lectures/Week
25ITEL512	Paper II – Linux Administration	2	2

Course Outcomes:
After successful completion of this course, students would be able to
CO1:Recognize the structure of the Linux file system hierarchy and the purpose of key directories.
CO2:Describe the significance of file permissions and ownership in a multi-user environment.
CO3:Navigate the Linux file system using the command-line interface.
CO4:Differentiate between various Linux distributions based on their target use cases and package management systems.

Unit	Topics	No of Lectures
I	**Fundamentals of Linux** **Introduction to Linux**: What is Linux? History and evolution, Understanding different Linux distributions, Open Source philosophy and licensing, Basic Linux architecture. **Getting Started with the Command Line:** Basic commands, Working with files and directories, Understanding file permissions, types and attributes, Using the chmod and chown commands to manage permissions and ownership, Accessing the Linux terminal, File and directory manipulation. **Navigating the File System:** Linux file system hierarchy, Absolute and relative paths, Using commands like find and locate to search for files. **Text File Handling:** Viewing file content, Basic text editing using nano or vi, Input/output redirection (>, >>, <), Piping commands (|) and using filters like grep.	15
II	**Basic System Administration** **User and Group Management:** Understanding user accounts and groups, Creating, modifying, and deleting users and groups, Managing user passwords, Using the sudo command for administrative privileges. **Package Management:** Introduction to package management systems, Searching for, installing, upgrading, and removing packages, Working with software repositories. **Process Management:** Understanding processes and their states, Viewing running processes, Controlling processes, Running commands in the background (&, nohup). **Basic System Configuration and Monitoring:** Understanding IP addresses and gateways, Checking disk space (df, du), Monitoring memory usage, Basic network configuration (viewing network interfaces: ip addr, ifconfig, ping) and Understanding system logs (/var/log/syslog, /var/log/auth.log, journalctl, systemd).	15

References:

- Linux Administration Handbook" by Evi Nemeth, Garth Snyder, Trent R. Hein, and Ben Whaley.
- LPIC-1 Linux Professional Institute Certification Study Guide: Exams 101-500 and 102-500" by Christine Bresnahan and Richard Blum.

Course Code	ELECTIVE SEM-V	Credits	Lectures/Week
25ITELP52	Linux Administration Practical	2	4

Course Outcomes:
After successful completion of this course, students would be able to
CO1:Identify fundamental command-line commands for file and directory manipulation.
CO2:Interpret absolute and relative file paths.
CO3:Utilize redirection and piping to manipulate command output.
CO4:Determine the appropriate file permissions for different scenarios to ensure security and access control.

1	**Command Line Mastery & File Operations:** ● Navigate through the file system using various cd commands (absolute and relative paths). ● Create a directory structure: projects/web/public_html. ● Create empty files named index.html, style.css, and script.js within the public_html directory using touch. ● Copy index.html to your home directory. ● Move style.css into a new directory named css within public_html. ● Remove the empty script.js file. ● Remove the entire web directory structure using a single command.
2	**File Permissions Deep Dive:** ● Create a new file named secret.txt. ● Examine its default permissions using ls -l. ● Change the permissions so only the owner can read and write to secret.txt (using octal mode). ● Change the permissions so the owner can read, write, and execute, and the group can read and execute (using octal mode). ● Use symbolic mode to remove write permission for the group and others from secret.txt. ● Create a script file run_me.sh and use symbolic mode to make it executable by the owner.
3	**Text File Analysis & Redirection:** ● Create a text file named data.log with at least five lines of varying information. ● Display the first three lines of data.log. ● Display the last two lines of data.log. ● Find all lines in data.log that contain the word "error" (case-insensitive). ● Redirect the output of the command ls -l /home to a file named homelist.txt.
4	**User and Group Account Management:** ● Create a new user account named tester. ● Set a password for the tester user. ● Create a new group named operations. ● Add the tester user to the operations group as a secondary group. ● Switch to the tester user account. After verifying, switch back to your original user.
5	**Software Package Management (Distribution Specific):** ● **(Debian/Ubuntu):** Update the package lists. Search for a package related to system monitoring (e.g., nload). Install the nload package. Check if it's installed. Then, remove the nload package.

	• **(Red Hat/Fedora):** Update the package lists. Search for a command-line web browser (e.g., links). Install the links package. Verify installation. Then, remove the links package.
6	**Process Control and Monitoring:** • Start a long-running process in the background (e.g., ping -c 10 8.8.8.8 &). Note the process ID (PID). • List all running processes and filter the output to find the ping process using grep. • Use the PID to terminate the background ping process using the kill command. • Use top or htop to observe CPU and memory usage, and identify the PID of a running application.
7	**File System Searching Techniques:** • Use the find command to locate all files named exactly report.txt under the /tmp directory. • Use find to locate all files in your home directory that are larger than 1MB. • Use find to locate all directories under /var/log that were modified in the last 7 days. • If locate is available, update its database and then use it to find a file you recently created.
8	**Advanced Redirection and Piping:** • List all files and directories in /usr/bin and count the number of entries using piping. • Append the output of the date command to the homelist.txt file you created earlier. • Use input redirection to provide a list of commands from a file to the bash interpreter.
9	**Basic Network Configuration Analysis:** • Display the IP configuration of your network interfaces using ip addr (or ifconfig). Identify the IP address and MAC address of your primary network interface. • Test network connectivity to a specific hostname (e.g., example.com) using the ping command. Observe the output. • Examine the contents of your system's DNS resolver configuration file (usually /etc/resolv.conf).
10	**System Logging and Resource Monitoring:** • Use the df -h command to identify the file system with the highest disk usage. • Use the du -sh command to find the size of the /var/log directory. • Use the free -m command to determine the amount of free and used RAM. • View the last 15 lines of the system log file (/var/log/syslog or /var/log/messages). • Use journalctl to view logs specifically for the systemd-user-sessions.service.

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	ELECTIVE SEM – V	Credits	Lectures /Week
25ITEL513	Paper III – Cyber Security	2	2

Course Outcomes:
After successful completion of this course, students would be able to
CO1:Define key concepts of cyber security, threats, and security principles.
CO2:Describe various types of cyber-attacks like phishing, malware, and DoS.
CO3: Demonstrate the configuration and purpose of firewalls, IDS/IPS, and VPNs.
CO4:Analyze the security mechanisms in wireless networks like WPA2 and identify potential risks.

Unit	Topics	No of Lectures
I	**Introduction to Cyber Security:** Difference between cyber security, information security, and network security Goals of cybersecurity: protection against threats, ensuring safe and secure data environments **Types of Cyber Threats:** Malware (Virus, Worm, Trojan Horse, Spyware, Adware) Phishing (Spear phishing, Email spoofing) Ransomware (Working, real-world attacks like WannaCry) SQL Injection (Types, prevention, and examples) DoS and DDoS (Tools, types: volumetric, protocol, application layer attacks)	15
II	**Network Security:** Definition and role of firewalls Types: Packet-filtering, Stateful Inspection, Application-level, Next-Generation Firewalls (NGFW) Rule-based configuration and zones Concept of DMZ (Demilitarized Zone) **Security in Wireless Networks:** Wireless security protocols: WEP, WPA, WPA2, WPA3 Encryption standards: RC4, TKIP, AES Common wireless attacks: Evil twin, Packet sniffing, Deauthentication Wireless security best practices	15

References:

- **Cybersecurity Essentials** – Charles J. Brooks, Christopher Grow, Philip Craig
- **Cybersecurity for Beginners** – Raef Meeuwisse
- **Computer Security: Principles and Practice** – William Stallings & Lawrie Brown
- **Network Security Essentials: Applications and Standards** – William Stallings

Course Code	ELECTIVE SEM-V	Credits	Lectures/Week
25ITELP53	**Cyber Security Practical**	**2**	**4**

Course Outcomes:
After successful completion of this course, students would be able to
CO1:Identify common cyber threats such as phishing, malware, and insecure websites.
CO2:Demonstrate how to secure browsers, analyze traffic using Wireshark, and use basic firewalls.
CO3:Simulate and observe basic security attacks like SQL injection in a controlled environment.
CO4: Analyze network packets and firewall behavior to detect potential threats.

1	Exploring and analyzing different types of malware using VirusTotal or Hybrid Analysis.
2	Capturing and analyzing network packets using Wireshark.
3	Setting up a basic firewall using iptables or pfSense
4	Demonstrating SQL Injection on a test website
5	Footprinting and Reconnaissance using tools like Nmap & Whois
6	Setting up a Secure File Transfer Using SFTP
7	Analyzing raw email headers to identify phishing or spoofing.
8	Setting up Two-Factor Authentication (2FA) on accounts for secure login.
9	Demonstrate brute-force attack using dictionary files.
10	Password cracking using tools like John the Ripper or Hashcat

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	MINOR SEM – V	Credits	Lectures/ Week
25ITMR531	Internet of Things	2	2

Course Outcomes:
After successful completion of this course, students would be able to
CO1: Explain the various concepts, terminologies, and architecture of IoT systems
CO2: Remember the knowledge of various IP based protocols for design of IoT systems
CO3: Implement the various protocols for design of IoT systems
CO4: Compare and Contrast various Industrial and other applications of IoT

Unit	Topics	No of Lectures
I	**Fundamentals of IoT:** Introduction, Definitions & Characteristics of IoT, IoT Architectures, Physical & Logical Design of IoT, Enabling Technologies in IoT, History of IoT, About Things in IoT, The Identifiers in IoT, About the Internet in IoT, IoT frameworks, IoT and M2M. **Sensors Networks** : Definition, Types of Sensors, Types of Actuators, Examples and Working, IoT Development Boards: Arduino IDE and Board Types, Raspberry Pi Development Kit, RFID Principles and components, Wireless Sensor Networks: History and Context, The node, Connecting nodes, Networking Nodes, WSN and IoT	15
II	**Wireless Technologies for IoT:** WPAN Technologies for IoT: IEEE 802.15.4, Zigbee, HART, NFC, Z-Wave, BLE, Bacnet, Modbus. IP Based Protocols for IoT IPv6, 6LowPAN, RPL, REST, AMPQ, CoAP, MQTT. Edge connectivity and protocols **Data Handling & Analytics:** Introduction, Big Data, Types of data, Characteristics of Big data, Data handling Technologies, Flow of data, Data acquisition, Data Storage, Introduction to Hadoop. Introduction to data Analytics, Types of Data analytics, Local Analytics, Cloud analytics and applications **Applications of IoT:** Home Automation, Smart Cities, Energy, Retail Management, Logistics, Agriculture, Health and Lifestyle, Industrial IoT, Legal challenges, IoT design Ethics, IoT in Environmental Protection.	15

References:

- Hakima Chaouchi, — “The Internet of Things Connecting Objects to the Web” ISBN : 978-1- 84821-140-7, Wiley Publications
- Olivier Hersent, David Boswarthick, and Omar Elloumi, — “The Internet of Things: Key Applications and Protocols”, WileyPublications
- Daniel Minoli, — “Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications”, ISBN: 978-1-118-47347-4, Willy Publications

Course Code	MINOR SEM-V Practical	Credits	Lectures/Week
25ITMRP51	**Internet of Things Practical**	**2**	**4**

Course Outcomes:
After successful completion of this course, students would be able to
CO1:Choose the hardware required for building an IoT system
CO2:Understand the concept of M2M with necessary protocols.
CO3:Develop embedded applications using IoT development boards, sensors and actuators.
CO4: Examine the python programming with Raspberry pi.

0	Starting Raspbian OS, Familiarising with Raspberry Pi Components and interface, Connecting to ethernet, Monitor, USB.
1	Displaying different LED patterns with Raspberry Pi.
2	Displaying Time over 4-Digit 7-Segment Display using Raspberry Pi
3	Home Automation using Node-RED and Raspberry Pi
4	Controlling Raspberry Pi with WhatsApp.
5	Setting up Wireless Access Point using Raspberry Pi
6	Temperature and Humidity monitoring using Node-Red and Raspberry Pi
7	Raspberry Pi GPS Module Interfacing
8	IoT based Web Controlled Home Automation using Raspberry Pi
9	Visitor Monitoring with Raspberry Pi and Pi Camera
10	a. Interfacing Raspberry Pi with RFID. b. Building Google Assistant with Raspberry Pi. c. Installing Windows 10 IoT Core on Raspberry Pi

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	VSC SEM-V	Credits	Lectures/Week
25ITVSP51	AngularJS Practical	2	4

Course Outcomes:
After successful completion of this course, students would be able to
CO1:Recall fundamental AngularJS concepts and the purpose of core components.
CO2:Explain the principles of data binding, client-side validation, RESTful API interaction, service-based data sharing and basic state management within the AngularJS framework.
CO3:Develop practical AngularJS applications to create reusable and interactive UI components using AngularJS services.
CO4:Evaluate the structure and functionality of AngularJS applications.

1	**Dynamic Form Generation and Validation:** Create an Angular.js application that dynamically generates a form based on a JSON configuration. The JSON should define the form fields (text, select, checkbox), their data types, and validation rules (required, email, pattern). Implement client-side validation using Angular.js directives and display appropriate error messages. **Focus**: $scope, ng-repeat, ng-model, ng-required, ng-pattern, custom validation.
2	**RESTful API Integration and Data Display:** Build an application that fetches data from a public RESTful API (e.g., JSONPlaceholder, OpenWeatherMap). Display the data in a user-friendly format (table, list, cards). Implement filtering and sorting functionality. **Focus**: $http, $resource (if applicable), data binding, filters, sorting.
3	**Custom Directive for Interactive Components:** Develop a custom directive that creates an interactive component, such as a star rating system or a color picker. The directive should handle user interaction, update the model, and display the changes. **Focus**: directive definition object, isolated scope, event handling, DOM manipulation.
4	**Routing and Navigation with $routeProvider:** Create a single-page application (SPA) with multiple views. Implement routing using $routeProvider to navigate between these views. Each view should display different content. **Focus**: $routeProvider, ng-view, partial views.
5	**Service for Data Sharing and Logic Encapsulation:** Build an application with a service that manages shared data or performs complex business logic. Demonstrate how different controllers can interact with the service and update the shared data. **Focus**: service, data sharing, logic encapsulation.
6	**Implementing a Simple Search Functionality:** Create an application that displays a list of items. Add a search input field that filters the list based on user input. Implement a case-insensitive search. **Focus**: ng-model, filters, data binding.
7	**File Upload with Progress Tracking:** Implement a file upload functionality using Angular.js. Display a progress bar to show the upload progress. **Focus**: $http with file upload, progress events, DOM manipulation.

8	**Creating a Reusable Modal Dialog:** Develop a reusable modal dialog component using a custom directive or service. The modal should be customizable with title, content, and buttons. Demonstrate how to open and close the modal from different parts of the application. **Focus**: Custom directives or services, DOM manipulation, event handling. 9.
9	**Implementing a Drag-and-Drop Interface:** Build an application that allows users to drag and drop elements within a list or between lists. **Focus**: Custom directives, event handling (dragstart, dragover, drop), DOM manipulation.
10	**Implementing a simple state management using a service:** Create a service that manages the application state. Implement functions to update and retrieve the state. Create a small application that uses this service to change and display state values. **Focus**: Services, state management, data sharing.

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	SEM-V	Credits	Hours
25ITFP57	Project Dissertation	2	60

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)*

    - *o* Declaration *(Sample attached in Appendix)*
    - o Acknowledgement
    - o Table of Contents
    - o Original Copy of approved Project Proposal
    - o Self-attested copy of Plagiarism Report from any open source tool.
    - o Chapters / Sections depending upon the type of project
    - o List of Tables and/or List of Figures
    - o References (IEEE / Springer format)
    - o Glossary
    - o 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.

SAMPLE TITLE PAGE FORMAT

A PROJECT REPORT

on

<PROJECT NAME>

Submitted by

Mr. XYZ

in partial fulfillment for the award of the degree of

BACHELOR OF SCIENCE

in

Information Technology

under the guidance of

<Guide Name>

Department of Computer Science and IT

<<College Logo>>

<<College Name>>

(Sem V / VI)

(202- – 202-)

SAMPLE CERTIFICATE FORMAT

<<College Logo>>

<<College Name>>,

<<College Address>>

Department of Information Technology

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 **IT** during the academic year **202- – 202-**.

Date of Submission:

Project Guide

Head / Incharge,
Department Computer Science and IT

College Seal

Signature of Examiner

SAMPLE DECLARATION FORMAT

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:

SEMESTER-VI

Course Code	MAJOR SEM – VI	Credits	Lectures/ Week
25ITMJ611	Paper I– Artificial Intelligence	2	2

Course Outcomes:

After successful completion of this course, students would be able to

CO1: Identify and to gain perspective of AI and its foundations.

CO2: Understand the basic principles of AI towards problem solving, inference, perception, knowledge representation, and learning and how heuristic functions guide informed search strategies towards the goal.

CO3: Apply basic evaluation metrics to assess the performance of machine learning models.

CO4: Analyze the factors that influence the choice of a particular learning algorithm.

Unit	Topics	No of Lectures
I	**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, Example Problems, Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions. **Expert System and Applications:** Phases in Building ExpertSystem,ExpertSystem Architecture, Expert System versus Traditional Systems, Rule based Expert Systems, Blackboard Systems, Truth Maintenance System, Application of ExpertSystems,Shell and Tools	15
II	**Learning from Examples:** Forms of Learning, Supervised Learning, Unsupervised Learning, Decision Trees, Evaluating and Choosing the Best Hypothesis, Theory of Learning, Regression and Classification with Linear Models, Artificial Neural Networks, Nonparametric Models, Support Vector Machines, Ensemble Learning, Practical Machine Learning	15

References:

- Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig,3rd Edition, Pearson, 2010.
- 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	MAJOR SEM-VI	Credits	Lectures /Week
25ITMJP61	Artificial Intelligence Practical	2	4

Course Outcomes:
After successful completion of this course, students would be able to
CO1: Articulate the historical development and current trends in Artificial Intelligence, demonstrating a comprehensive understanding of its foundations and principles.
CO2: Understand the proficiency in implementing and analyzing various search algorithms, utilizing both uninformed and informed strategies to solve complex problems efficiently.
CO3: Apply adversarial search techniques to decision-making in competitive environments, including games, and effectively manage uncertainty and partial observability.
CO4: Demonstrate competency in logical reasoning and inference, utilizing propositional and first-order logic to represent and solve real-world problems in AI applications.

	Write programs for the following:
1	Implement depth first search algorithm.
2	Implement breadth first search algorithm.
	Write programs for the following:
3	Simulate 4-Queen / N-Queen problem
4	Solve the Tower of Hanoi problem.
	Write programs for the following:
5	Implement alpha beta search.
6	Implement hill climbing problems.
	Write programs for the following:
7	Implement A* algorithm.
8	Solve the water jug problem.
	Write programs for the following:
9	Simulate tic – tac – toe game using min-max algorithm.
10	Shuffle deck of cards.

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	MAJOR SEM – VI	Credits	Lectures/ Week
25ITMJ612	Paper II – Business Intelligence and Data Analytics	2	2

Course Outcomes:

After successful completion of this course, students would be able to

CO1: Identify the key components of a Business Intelligence (BI) architecture.

CO2: Describe the typical stages involved in a data mining project.

CO3: Apply data validation, transformation, and reduction techniques to a given dataset.

CO4: Evaluate the effectiveness of different marketing and sales force management models.Fact findings using Strategic Organizational Intelligence: Bridging Gaps, Cultivating Knowledge, and Embracing Artificial Intelligence

Unit	Topics	No of Lectures
I	**Business intelligence:** Effective and timely decisions, Data, information and knowledge, The role of mathematical models, Business intelligence architectures, Ethics and business intelligence **Decision support systems:** Definition of system, Representation of the decision-making process, Evolution of information systems, Definition of decision support system, Development of a decision support system	15
II	**Mathematical models for decision making**: Structure of mathematical models, Development of a model, Classes of models **Data mining:** Definition of data mining, Representation of input data, Data mining process, Analysis methodologies Data preparation: Data validation, Data transformation, Data reduction **Business intelligence applications:** **Marketing models:** Relational marketing, Sales force management, Logistic and production models: Supply chain optimization, Optimization models for logistics planning, Revenue management systems.Data envelopment analysis, The CCR model, Identification of good operating practices	15

References:

- “Business Intelligence: A Managerial Approach" by Efraim Turban, Ramesh Sharda, and Dursun Delen
- “Data Mining: Concepts and Techniques" by Jiawei Han, Micheline Kamber, and Jian Pei
- “Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software" by William W. Cooper, Lawrence M. Seiford, and Kaoru Tone

Course Code	MAJOR SEM-VI	Credits	Lectures/Week
25ITMJP62	Business Intelligence Practical	2	4

Course Outcomes:
After successful completion of this course, students would be able to
CO1: Learners can explore the concepts of Strategic Decision Support and Harnessing Data for Informed Business Decisions
CO2: Application used for Data-Driven Mathematical Models and Data Mining for Informed Decision Making
CO3: Managing data through Advanced Data Analysis Techniques: Classification, Clustering, and Model Evaluation
CO4: Analyzing Strategic Information Management: Enhancing Decision-Making Across Marketing, Logistics, and Production

	Perform the analysis for the following
1	Import the data warehouse data in Microsoft Excel and create the Pivot table and Pivot Chart
2	Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to perform data analysis
3	Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data. Use Excel.
	Perform data visualization
4	Perform data visualization using Tableau on any sales data.
5	Perform data visualization using PowerBI on any sales data.
6	The dataset ToyotaCorolla.xls contains data on used cars on sale during the late summer of 2004 in The Netherlands. It has 1436 records containing details on 38 attributes, including Price, Age, Kilometers, HP, and other specifications. a. Explore the data using the data visualization (matrix plot) capabilities of XLMiner. Which of the pairs among the variables seem to be correlated? b. We plan to analyze the data using various data mining techniques described in future chapters. Prepare the data for use as follows: i. The dataset has two categorical attributes, Fuel Type and Metallic. (a) Describe how you would convert these to binary variables. (b) Confirm this using XLMiner's utility to transform categorical data into dummies..
7	Creating Dashboard for Account System
8	Creating Dashboard for E-commerce
9	Creating Dashboard for Supply Chain Management
10	**Mini Project**

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	MAJOR SEM – VI	Credits	Lectures/ Week
25ITMJ613	Paper III – Software Testing & Quality Assurance	2	2

Course Outcomes:

After successful completion of this course, students would be able to

CO1:Define the core concepts of software testing and its importance.

CO2:Explain why software testing is a crucial part of the software development process.

CO3:Apply basic techniques within functional, non-functional, and static testing scenarios.

CO4:Compare and contrast different types of software testing.

Unit	Topics	No of Lectures
I	**Importance, Testing Types & Techniques:** Introduction, Functional Testing, Nonfunctional Testing, Static Testing. **Software Development Life Cycle:** Planning Phase Requirements, Gathering Phase, Design Phase, Development Phase, Deployment Phase, Maintenance Phase, The Role of Testing in the SDLC. V-MODEL **Software Testing Life Cycle:** Introduction, V-Model in software development and testing projects, Verification Phases, Validation Phases, Cross-Cutting Topics Related to the V-Model **Test Planning:** Defining Testing Objectives, Determining the Scope of Testing, Selecting the Testing Approach, Identifying Testing Resources, Developing the Test Schedule, Defining Test Cases, Identifying Test Data, Defect Management Process, Stop Testing Criteria, Reviewing and Approving the Test Plan, Benefits of Test Planning, Test Plan Document, Test Planning Tools and Techniques	15
II	**Test Design Techniques:** Black-Box Testing Introduction, White-Box Testing Introduction, Static Testing Introduction. **Test Execution:** Test Execution Process, Test Environment Setup, Defect Reporting and Retesting, Regression Testing, Test Case Status Reporting, Test Case Completion, Techniques and Tools Used in Test Execution, Quality Metrics. **Test Automation:** Benefits of Test Automation, Record and Playback Tools, Scripting Tools, Hybrid Tools, Frameworks. **Testing in Agile Environment:** Agile Testing Principles, Agile Testing Quadrants, Test-Driven Development, Behavior-Driven Development, Acceptance Test-Driven Development, Continuous Integration and Continuous Delivery, Test Automation in Agile, Agile Testing Best Practices.	15

References:

- Introduction to Software Testing, Panagiotis Leloudas, Apress.
- Software Testing: Principles and Practices" by Srinivasan Desikan and Gopalaswamy Ramesh.
- Effective Software Testing: A Developer's Guide" by Maurício Aniche.

Course Code	ELECTIVE SEM – VI	Credits	Lectures/ Week
25ITEL621	Paper I – Information Security	2	2

Course Outcomes:

After successful completion of this course, students would be able to

CO1: Define the key terms and concepts related to information security, threats, and types of attacks.

CO2:Describe the CIA triad and different defense models used in secure system design.

C03:Apply authentication and authorization techniques in designing secure systems.

CO4:Analyze firewall design and functions to enhance network perimeter security.

Unit	Topics	No of Lectures
I	**Information Security Overview** : The Importance of Information Protection, The Evolution of Information Security, Threat Definition, Types of Attacks, Risk Analysis. **Secure Design Principles**: The CIA Triad and Other Models, Defense Models, Zones of Trust, Best Practices for Network Defense. **Authentication and Authorization**: Encryption: A Brief History of Encryption, Symmetric-Key Cryptography, Public Key Cryptography, Public Key Infrastructure. **Database Security**: General Database Security Concepts, Database Backup and Recovery	15
II	**Secure Network Design**: Introduction to Secure Network Design, Performance, Availability, Security. **Network Device Security**: Switch and Router Basics, Network Hardening. **Firewalls:** Overview, The Evolution of Firewalls, Core Firewall Functions, Additional Firewall Capabilities, Firewall Design. **Intrusion Detection and Prevention Systems**: IDS Concepts, IDS Types and Detection Models, IDS Features, IDS Deployment Considerations	15

References:

- TheCompleteReference: Information Security, Mark Rhodes- Ousley, McGraw- Hill, 2nd Edition, 2013
- Essential Cybersecurity Science: Josiah Dykstra, O"Reilly, Fifth Edition, 2017
- Principles of Computer Security: CompTIA Security+ and Beyond, Wm.Arthur Conklin, Greg White, McGraw- Hill, 2nd Edition, 2010

Course Code	ELECTIVE SEM-VI	Credits	Lectures/Week
25ITELP61	**Information Security Practical**	**2**	**4**

Course Outcomes:
After successful completion of this course, students would be able to
CO1:Recall the purpose and functions of router configuration, AAA, ACLs, VPNs, firewalls, and IPS.
CO2: Explain the significance of access control and VPNs in a secure network.
CO3:Apply commands to configure routers and implement AAA authentication using the CLI.
CO4:Analyze the effectiveness of site-to-site IPsec VPN configuration by verifying secure communication.

1	Configure Routers
2	Configure AAA Authentication
3	Configuring Extended ACLs
4	Configure IP ACLs to Mitigate Attacks and IPV6 ACLs
5	Configuring a Zone-Based Policy Firewall
6	Configure IOS Intrusion Prevention System (IPS) Using the CLI
7	Configure Layer 2 Security
8	Configure Layer 2 VLAN Security
9	Configure and Verify a Site-to-Site IPsec VPN Using CLI
10	Configuring ASA Basic Settings and Firewall Using CLI

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	ELECTIVE SEM – VI	Credits	Lectures/ Week
25ITEL622	Paper II – Mojo Programming	2	2

Course Outcomes:
After successful completion of this course, students would be able to
CO1: Identify the mechanisms for error handling and modularization in Mojo.
CO2: Summarize the potential advantages of using Mojo in areas like AI/ML and systems programming.
CO3: Develop Mojo programs that utilize fundamental data structures like lists and dictionaries to store and manipulate data.
CO4: Analyze the memory usage and performance implications of different coding patterns in Mojo.

Unit	Topics	No of Lectures
I	**Mojo Fundamentals and Python Interoperability** **Introduction to Mojo and Development Environment:** Mojo's design philosophy and key features for AI/ML and systems programming, Setting up the Mojo development environment (SDK, Jupyter, CLI), "Hello, World!" in Mojo: Syntax, structure, and execution, Brief comparison with Python and other relevant languages. **Core Language Constructs:** Data types, Variables and constants, Operators, Control flow, Basic function definition with def and fn, parameters, return types. **Fundamental Data Structures:** Tuples, Lists (or dynamic arrays), Dictionaries (or hash maps), basic operations. **Python Interoperability:** Importing and using Python modules, Interacting with Python functions and objects, Understanding basic performance considerations of interoperability.	15
II	**Advanced Mojo Concepts and Potential Applications** **Structures and Methods:** Defining custom data types using struct. Fields and initialization, Implementing methods on structs using fn and def. Receiver types. **Traits and Basic Generics:** Defining and implementing traits for shared behavior, Introduction to basic generic functions. **Performance Considerations:** Deeper dive into the differences between def and fn for performance, Role of type annotations in fn., Basic memory management concepts (value semantics), Introduction to profiling (if tools are available). **Error Handling and Modules:** Basic error handling mechanisms in Mojo, Organizing code into modules. **Potential Applications and Future Directions:** Overview of Mojo's potential in AI/ML, data science, and systems programming, Discussion on the future roadmap of Mojo.	15

References:

- Machine Learning with MOJO Programming Language: A Developer's Guide to Building High-Performance Language Model & Rapid Ai Development.
- Mojo Programming for Everyone: Hands-on Guide to Large Language Model Development with Functional Programming.

Course Code	ELECTIVE SEM-VI Practical	Credits	Lectures/ Week
25ITELP62	Mojo Programming Practical	2	4

Course Outcomes:
After successful completion of this course, students would be able to
CO1: Identify the syntax for importing and using Python modules within Mojo.
CO2: Summarize the basic principles of performance considerations when writing Mojo code.
CO3: Develop Mojo programs that utilize fundamental data structures like lists and dictionaries to store and manipulate data.
CO4: Determine the appropriate data structure in Mojo for solving specific data manipulation problems.

1	**Optimized Mathematical Functions:** Implement mathematical functions (e.g., factorial, Fibonacci) using both def and fn, and compare their performance using basic timing mechanisms. Analyse the performance differences.
2	**Data Structure Implementation:** Implement a basic data structure like a linked list or a binary search tree using Mojo structs and methods (fn for performance-critical operations).
3	**File Processing and Data Analysis:** Write a Mojo program that reads a large CSV file and performs basic data analysis (e.g., calculating averages, finding maximums/minimums) using efficient Mojo code.
4	**Image Manipulation (Interoperability):** Use Mojo in conjunction with a Python image processing library (like Pillow) to perform basic image manipulations (e.g., resizing, filtering). Analyse the performance implications of crossing the language boundary.
5	**Implementing a Trait-Based System:** Design a system using traits (e.g., Shape with area() and perimeter()) and implement different geometric shapes as structs that adhere to this trait.
6	**Basic Concurrency:** Write a Mojo program that uses concurrency (e.g., goroutines-like) to perform a task in parallel, such as processing elements of a large list or performing network requests.
7	**Low-Level Memory Operations:** Explore and demonstrate basic pointer manipulation or memory allocation for performance-critical tasks.
8	**Building a Simple CLI Application:** Design and implement a basic command-line interface (CLI) tool in Mojo that takes user input and performs a specific task.
9	**Interfacing with System APIs :** Explore if Mojo allows interaction with basic system-level APIs (e.g., getting system information) and write a program to demonstrate this.

10	**Mini-Project: Performance-Critical Task:** Choose a computationally intensive task (e.g., a simple machine learning algorithm, a simulation) and implement it in Mojo, focusing on optimizing its performance using fn and other potential Mojo features. Compare its performance to a Python implementation.

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	ELECTIVE SEM – VI	Credits	Lectures/ Week
25ITEL623	Paper III – IT Infrastructure Management	2	2

Course Outcomes:

After successful completion of this course, students would be able to

CO1:Apply the principles and concepts of ITIL 4 to enhance IT service management practices

CO2: Demonstrate proficiency in implementing ITIL management practices

CO3:Successfully drive stakeholder value and engage in value co-creation

CO4:Develop the skills to create, deliver, and support services within the ITIL service value system

Unit	Topics	No of Lectures
I	**Introduction:** IT service management in the modern world, About ITIL 4, The structure and benefits of the ITIL 4 framework. **Key concepts of service management** : Value and value co-creation, Organizations, service providers, service consumers, and other stakeholders, Products and services, Service relationships, Value: outcomes, costs, and risks. **The four dimensions of service management**: Organizations and people, Information and technology, Partners and suppliers, Value streams and processes, External factors. **The ITIL service value system** : Service value system overview, Opportunity, demand, and value, The ITIL guiding principles, Governance, Service value chain, Continual improvement.	15
II	**ITIL management practices:** General management practices,Service management practices, Technical management practices. **Drive Stakeholder Value**: Introduction, The customer journey, Step 1: Explore, Step 2: Engage, Step 3: Offer, Step 4: Agree, Step 5: Onboard, Step 6: Co-create, Step 7: Realize.	15

References:

- ITIL Foundation 4 Edition-Roman-Jouravlev, Akashay Anand, et. al-TSO, AXELOS-2 Edition-2019
- ITIL 4 High-Velocity IT-AkashayAnand-Dan Ashby, et. al--TSO, AXELOS-1 Edition-2020

Course Code	ELECTIVE SEM-VI Practical	Credits	Lectures/Week
25ITELP63	**Advanced Mobile Programming Practical**	**2**	**4**

Course Outcomes:
After successful completion of this course, students would be able to
CO1: Recall basic concepts of Android, Android Studio, and app components.
CO2: Understand the Android development environment and build simple apps.
CO3: Apply Android resources like themes, strings, colors, and images in apps.
CO4: Analyze the activity and fragment life cycles to manage app behavior during state transitions.

1	**Introduction to Android, Introduction to Android Studio IDE, Application Fundamentals**: Creating a Project, Android Components, Activities, Services, Content Providers, Broadcast Receivers, Interface overview, Creating Android Virtual device, USB debugging mode, Android Application Overview. Simple "Hello World" program.
2	**Programming Resources** Android Resources: (Color, Theme, String, Drawable, Dimension, Image)
3	**Programming Activities and fragments** Activity Life Cycle, Activity methods, Multiple Activities, Life Cycle of fragments and multiple fragments.
4	**Programs related to different Layouts** Coordinate, Linear, Relative, Table, Absolute, Frame, List View, Grid View.
5	**Programming UI elements** AppBar, Fragments, UI Components
6	**Programming menus, dialog, dialog fragments**
7	**Programs on Intents, Events, Listeners and Adapters** The Android Intent Class, Using Events and Event Listeners
8	**Programs on Services, notification and broadcast receivers**
9	**Database Programming with SQLite**
10	**Programming threads, handles and asynchronized programs**

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	MINOR SEM – VI	Credits	Lectures/ Week
25ITMR631	Fundamental of GIS	2	2

Course Outcomes:
After successful completion of this course, students would be able to
CO1:Recall the definitions of key GIS terminology and the importance of error propagation in spatial data processing.
CO2:Explain how the real world is represented in GIS through models, maps, and spatial databases.
CO3:Utilize basic GIS software functionalities for data viewing, querying, and manipulation.
CO4:Critically assess the effectiveness of different spatial data presentation methods.

Unit	Topics	No of Lectures
I	**Foundations of Geographic Information Systems** **Introduction to GIS**: Defining GIS, GISystems, GIScience, and GIApplications. **Spatial Data and Geoinformation:** The nature of spatial data, types of geographic phenomena (fields and objects), and geoinformation. **Representations of the Real World:** Models and modeling, maps, databases, and spatial databases. **GIS Architecture and Functionality:** Hardware and software trends, GIS software components, and essential GIS functionalities.Spatial Data Infrastructure (SDI). **Stages of Spatial data handling:** Spatial data handling and preparation, Spatial Data Storage and maintenance. **Database management Systems:** Reasons for using a DBMS, Alternatives for data management, The relational data model. **GIS and Spatial Databases:** Linking GIS and DBMS, Spatial database functionality.	15
II	**Spatial Analysis and GIS Applications** **Spatial Query and Analysis**: Techniques for querying and analyzing spatial data. **Overlay Functions:** Vector and raster overlay operations. **Neighborhood Functions:** Proximity computations, diffusion, flow computation, and raster-based surface analysis. **Advanced Analysis:** Network analysis, interpolation, and terrain modeling. Spatial Data Presentation. **GIS and Application models:** GPS, Open GIS Standards, GIS. Applications and Advances. **Error Propagation in spatial data processing:** How Errors propagate, Quantifying error propagation.	15

References:

- Principles of Geographic Information Systems, P.A Burrough and R.A.McDonnell, Oxford University Press, Third 1999
- Fundamentals of Spatial Information Systems, R.Laurini and D. Thompson, Academic Press 1994

Course Code	MINOR SEM-VI	Credits	Lectures/ Week
25ITMRP61	Fundamental of GIS Practical	2	4

Course Outcomes:
After successful completion of this course, students would be able to
CO1:Recall the fundamental concepts related to GIS data types (vector and raster), map elements, coordinate systems, georeferencing, spatial analysis techniques (overlay, neighborhood, interpolation), and automation tools within GIS software.
CO2:Explain the concept of map projections and coordinate reference systems (CRS) in GIS.
CO3:Perform basic advanced GIS analysis, including nearest neighbor analysis, raster sampling using points or polygons, and point interpolation.
CO4:Compare and contrast the characteristics and appropriate uses of vector and raster data models.

1	Familiarizing Quantum GIS: Installation of QGIS, datasets for both Vector and Raster data, Maps.
2	Creating and Managing Vector Data: Adding vector layers, setting properties, formatting, calculating line lengths and statistics
3	Exploring and Managing Raster data: Adding raster layers, raster styling and analysis, raster mosaicking and clipping.
4	Making a Map, Working with Attributes, Importing Spreadsheets or CSV files Using Plugins, Searching and Downloading OpenStreetMap Data
5	Working with attributes, terrain Data
6	Working with Projections and WMS Data
7	Georeferencing TopoSheets and Scanned Maps Georeferencing Aerial Imagery Digitizing Map Data
8	Managing Data Tables and Spatial data Sets: Table joins, spatial joins, points in polygon analysis, performing spatial queries
9	Advanced GIS Operations 1: Nearest Neighbor Analysis, Sampling Raster Data using Points or Polygons, Interpolating Point Data
10	Advance GIS Operations 2: Batch Processing using Processing Framework. Automating Complex Workflows using Processing Modeler. Automating Map Creation with Print Composer Atlas.

Note: Students are required to complete a minimum of 80% of all the practicals.

Course Code	SEM-VI	Credits	Hours
25ITOJT67	Project Implementation	4	120

OJT- Project Implementation Semester VI

Project Dissertation Semester V and Project Implementation Semester VI

Chapter 1 to 4 should be submitted in Semester V in spiral binding. These chapters have also to be included in the Semester VI report. Semester VI report has to be hard bound with golden embossing. Students will be evaluated based on the dissertation in semester V and dissertation and viva voce in Semester VI.

I. OBJECTIVES

- Describe the Systems Development Life Cycle (SDLC).
- Evaluate systems requirements.
- Complete a problem definition.
- Evaluate a problem definition.
- Determine how to collect information to determine requirements.
- Perform and evaluate feasibility studies like cost-benefit analysis, technical feasibility, time feasibility and Operational feasibility for the project.
- Work on data collection methods for fact finding.
- Construct and evaluate data flow diagrams.
- Construct and evaluate data dictionaries.
- Evaluate methods of process description to include structured English, decision tables and decision trees.
- Evaluate alternative tools for the analysis process.
- Create and evaluate such alternative graphical tools as systems flow charts and state transition diagrams.
- Decide the S/W requirement specifications and H/W requirement specifications.
- Plan the systems design phase of the SDLC.
- Distinguish between logical and physical design requirements.
- Design and evaluate system outputs.
- Design and evaluate systems inputs.
- Design and evaluate validity checks for input data.
- Design and evaluate user interfaces for input.
- Design and evaluate file structures to include the use of indexes.
- Estimate storage requirements.
- Explain the various file update processes based on the standard file organizations.
- Decide various data structures.
- Construct and evaluate entity-relationship (ER) diagrams for RDBMS related projects.
- Perform normalization for the unnormalized tables for RDBMS related projects
- Decide the various processing systems to include distributed, client/server, online and others.
- Perform project cost estimates using various techniques.
- Schedule projects using both GANTT and PERT charts.
- Perform coding for the project.

- Documentation requirements and prepare and evaluate systems documentation.
- Perform various systems testing techniques/strategies to include the phases of testing.
- Systems implementation and its key problems.
- Generate various reports.
- Be able to prepare and evaluate a final report.
- Brief the maintenance procedures and the role of configuration management in operations.
- To decide the future scope and further enhancement of the system.
- Plan for several appendices to be placed in support with the project report documentation.
- Decide the various processing systems to include distributed, client/server, online and others.
- Perform project cost estimates using various techniques.
- Schedule projects using both GANTT and PERT charts.
- Perform coding for the project.
- Documentation requirements and prepare and evaluate systems documentation.
- Perform various systems testing techniques/strategies to include the phases of testing.
- Systems implementation and its key problems.
- Generate various reports.
- Be able to prepare and evaluate a final report.
- Brief the maintenance procedures and the role of configuration management in operations.
- To decide the future scope and further enhancement of the system.
- Plan for several appendices to be placed in support with the project report documentation.
- Work effectively as an individual or as a team member to produce correct, efficient, well organized and documented programs in a reasonable time.
- Recognize problems that are amenable to computer solutions, and knowledge of the tools necessary for solving such problems.
- Develop of the ability to assess the implications of work performed.
- Get good exposure and command in one or more application areas and on the software
- Develop quality software using the software engineering principles
- Developing the ability to communicate effectively.

II. Type of the Project

The majority of the students are expected to work on a real-life project preferably in some industry/ Research and Development Laboratories/Educational Institution/Software Company.

Students are encouraged to work in the areas listed below. However, it is not mandatory for a student to work on a real-life project. The student can formulate a project problem with the help of her/his Guide and submit the project proposal of the same. Approval of the project proposal is mandatory. If approved, the student can commence working on it, and complete it. Use the latest versions of the software packages for the development of the project.

III SOFTWARE AND BROAD AREAS OF APPLICATION

FRONT END / GUI Tools	.Net Technologies, Java
DBMS/BACK END	Oracle, SQL Plus, My SQL, SQL Server
LANGUAGES	C, C++, Java, [1] VC++, C#, R, Python
SCRIPTING LANGUAGES	PHP, JSP, SHELL Scripts (Unix), TCL/TK
.NET Platform	F#, C#, .Net Visual [2] C#, ASP.Net
MIDDLEWARE (COMPONENT) TECHNOLOGIES	COM/DCOM, Active-X, EJB
UNIX INTERNALS	Device Drivers, RPC, Threads, Socket programming
NETWORK WIRELESS TECHNOLOGIES	-
REALTIME OPERATING SYSTEM/ EMBEDDED SKILLS	LINUX, Raspberry [3] Pi, Arduino, 8051
APPLICATION AREAS	Financial / Insurance / Manufacturing / Multimedia / Computer Graphics / Instructional Design / Database Management [4] System / Internet / Intranet / Computer Networking-Communication / Software development / E-Commerce / ERP / MRP / TCP-IP programming / Routing protocols programming / Socket programming

IV. Introduction

The project report should be documented with a scientific approach to the solution of the problem that the students have sought to address. The project report should be prepared in order to solve the problem in a methodical and professional manner, making due references to appropriate techniques, technologies and professional standards. The student should start the documentation process from the first phase of software development so that one can easily identify the issues to be focused upon in the ultimate project report. The student should also include the details from the project diary, in which they will record the progress of their project throughout the course.

The project report should contain enough details to enable examiners to evaluate the work. The important points should be highlighted in the body of the report, with details often referred to appendices.

1.1 PROJECT REPORT:

4.6 Test Cases Design

The documentation should use tools like star UML, Visuo for windows, Rational Rose for design as part of Software Project Management Practical Course. The documentation should be spiral bound for semester V and the entire documentation should be hard bound during semester VI.

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

5.2 Coding Details and Code Efficiency

5.2.1 Code Efficiency

5.3 Testing Approach

5.3.1 Unit Testing

5.3.2 Integrated Testing

5.3.3 Beta Testing

5.4 Modifications and Improvements

5.5 Test Cases

CHAPTER 6: RESULTS AND DISCUSSION

6.1 Test Reports

6.2 User Documentation

CHAPTER 7: CONCLUSIONS

7.1 Conclusion

7.1.1 Significance of the System

7.2 Limitations of the System

7.3 Future Scope of the Project REFERENCES

GLOSSARY APPENDIX A APPENDIX B

V. EXPLANATION OF CONTENTS

Title Page

Sample format of Title page is given in Appendix 1 of this block. Students should follow the given format.

Original Copy of the Approved Proforma of the Project Proposal

Sample Proforma of Project Proposal is given in Appendix 2 of this block. Students should follow the given format.

Certificate of Authenticated work

Sample format of Certificate of Authenticated work is given in Appendix 3 of this block. Students should follow the given format.

Role and Responsibility Form

Sample format for Role and Responsibility Form is given in Appendix 4 of this block. Students should follow the given format.

Abstract

This should be one/two short paragraphs (100-150 words total), summarising the project work. It is important that this is not just a restatement of the original project outline. A suggested flow is background, project aims and main achievements. From the abstract, a reader should be able to ascertain if the project is of interest to them and, it should present results of which they may wish to know more details.

Acknowledgements:

This should express the student‟s gratitude to those who have helped in the preparation of the project.

Table of Contents:

The table of contents gives the readers a view of the detailed structure of the report. The students would need to provide section and subsection headings with associated pages. The formatting details of these sections and subsections are given below. Table of Figures: List of all Figures, Tables, Graphs, Charts etc. along with their page numbers in a table of figures.

Chapter 1: Introduction

The introduction has several parts as given below:

Background: A description of the background and context of the project and its relation to work already done in the area. Summarise existing work in the area concerned with the project work.

Objectives: Concise statement of the aims and objectives of the project. Define exactly what is going to be done in the project; the objectives should be about 30 /40 words.

Purpose, Scope and Applicability: The description of Purpose, Scope, and Applicability are given below:

- Purpose: Description of the topic of the project that answers questions on why this project is being done. How the project could improve the system's significance and theoretical framework.

- Scope: A brief overview of the methodology, assumptions and limitations. The students should answer the question: What are the main issues being covered in the project? What are the main functions of the project?

- Applicability: The student should explain the direct and indirect applications of their work. Briefly discuss how this project will serve the computer world and people.

- Achievements: Explain what knowledge the student achieved after the completion of the work. What contributions has the project made to the chosen area?

Goals achieved - describes the degree to which the findings support the original objectives laid out by the project. The goals may be partially or fully achieved, or exceeded.

- Organisation of Report: Summarizing the remaining chapters of the project report, in effect, giving the reader an overview of what is to come in the project report.

Chapter 2: Survey of Technologies

In this chapter Survey of Technologies should demonstrate the students awareness and understanding of Available Technologies related to the topic of the project. The student should give the details of all the related technologies that are necessary to complete the project. They should describe the technologies available in the chosen area and present a comparative study of all those Available Technologies. Explain why the student selected the one technology for the completion of the objectives of the project.

Chapter 3: Requirements and Analysis

Problem Definition: Define the problem on which the students are working in the project.

Provide details of the overall problem and then divide the problem into subproblems. Define each sub- problem clearly.

Requirements Specification: In this phase the student should define the requirements of the system, independent of how these requirements will be accomplished. The Requirements Specification describes the things in the system and the actions that can be done on these things.Identify the operation and problems of the existing system.

Planning and Scheduling: Planning and scheduling is a complicated part of software

development. Planning, for our purposes, can be thought of as determining all the small tasks that must be carried out in order to accomplish the goal. Planning also takes into account rules, known as constraints, which control when certain tasks can or cannot happen. Scheduling can be thought of as determining whether adequate resources are available to carry out the plan. The student should show the Gantt chart and Program Evaluation Review Technique (PERT).

Software and Hardware Requirements: Define the details of all the software and hardware needed for the development and implementation of the project.

- Hardware Requirement: In this section, the equipment, graphics card, numeric coprocessor, mouse, disk capacity, RAM capacity etc. necessary to run the software must be noted.

- Software Requirements: In this section, the operating system, the compiler, testing tools, linker, and the libraries etc. necessary to compile, link and install the software must be listed.

Preliminary Product Description: Identify the requirements and objectives of the new system. Define the functions and operation of the application/system the students are developing as a project.

Conceptual Models: The student should understand the problem domain and produce a model of the system, which describes operations that can be performed on the system, and the allowable sequences of those operations. Conceptual Models could consist of complete Data Flow Diagrams, ER diagrams, Object-oriented diagrams, System Flowcharts etc.

Chapter 4: System Design

Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudocode and other documentation.

Basic Modules: The students should follow the divide and conquer theory, so divide the overall problem into more manageable parts and develop each part or module separately. When all modules are ready, the student should integrate all the modules into one system. In this phase, the student should briefly describe all the modules and the functionality of these modules.

Data Design: Data design will consist of how data is organised, managed and manipulated.

- Schema Design: Define the structure and explanation of schemas used in the project.
- Data Integrity and Constraints: Define and explain all the validity checks and constraints provided to maintain data integrity.

Procedural Design: Procedural design is a systematic way for developing algorithms or procedurals.

- Logic Diagrams: Define the systematic flow of procedure that improves its comprehension and helps the programmer during implementation. e.g., Control Flow Chart, Process Diagrams etc.
- Data Structures: Create and define the data structure used in procedures.
- Algorithms Design: With proper explanations of input data, output data, logic of processes, design and explain the working of algorithms.

User Interface Design: Define user, task, environment analysis and how to map those requirements in order to develop a “User Interface”. Describe the external and internal

components and the architecture of the user interface. Show some rough pictorial views of the user interface and its components.

Security Issues: Discuss Real-time considerations and Security issues related to the project and explain how the student intends avoiding those security problems. What are the security policy plans and architecture?

Test Cases Design: Define test cases, which will provide easy detection of errors and mistakes within a minimum period of time and with the least effort. Explain the different conditions in which the students wish to ensure the correct working of the project.

Chapter 5: Implementation and Testing

Implementation Approaches: Define the plan of implementation, and the standards the students have used in the implementation. Coding Details and Code Efficiency: Students do not need to include full source code, instead, include only the important codes (algorithms, applets code, forms code etc). The program code should contain comments needed for explaining the work a piece of code does. Comments may be needed to explain why it does it, or why it does a particular way. The student can explain the function of the code with a shot of the output screen of that program code.

- Code Efficiency: The student should explain how the code is efficient and how the students have handled code optimisation.

- Testing Approach: Testing should be according to the scheme presented in the system design chapter and should follow some suitable model – e.g., category partition, state machine-based. Both functional testing and user-acceptance testing are appropriate. Explain the approach of testing.

- Unit Testing: Unit testing deals with testing a unit or module as a whole. This would test the interaction of many functions but, do confine the test within one module.

- Integrated Testing: Brings all the modules together into a special testing environment, then checks for errors, bugs and interoperability. It deals with tests for the entire application. Application limits and features are tested here.

Modifications and Improvements: Once the students finish the testing they are bound to be faced with bugs, errors and they will need to modify your source code to improve the system. Define what modifications are implemented in the system and how it improved the system.

Chapter 6: Results and Discussion

Test Reports: Explain the test results and reports based on the test cases, which should show that the project is capable of facing any problematic situation and that it works fine in different conditions. Take the different sample inputs and show the outputs.

User Documentation: Define the working of the software; explain its different functions, components with screenshots. The user document should provide all the details of the product in such a way that any user reading the manual, is able to understand the working and functionality of the document.

Chapter 7: Conclusions

Conclusion: The conclusions can be summarised in a fairly short chapter (2 or 3 pages). This chapter brings together many of the points that would have been made in the other chapters. Limitations of the System: Explain the limitations encountered during the testing of the project that the students were not able to modify. List the criticisms accepted during the demonstrations of the project.

Future Scope of the Project describes two things: firstly, new areas of investigation prompted by developments in this project, and secondly, parts of the current work that was not completed due to time constraints and/or problems encountered.

REFERENCES

It is very important that the students acknowledge the work of others that they have used or adapted in their own work, or that provides the essential background or context to the project.

The use of references is the standard way to do this. Please follow the given standard for the references for books, journals, and online material. The citation is mandatory in both the reports.

E.g:

Linhares, A., & Brum, P. (2007). Understanding our understanding of strategic scenarios: What role do chunks play? Cognitive Science, 31(6), 989-1007.

https://doi.org/doi:10.1080/03640210701703725

Lipson, Charles (2011). Cite right : A quick guide to citation styles; MLA, APA, Chicago, the sciences, professions, and more (2nd ed.). Chicago [u.a.]: University of Chicago Press. p. 187.

ISBN 9780226484648.

Elaine Ritchie, J Knite. (2001). Artificial Intelligence, Chapter 2 ,p.p 23 - 44. Tata McGrawHill.

GLOSSARY

If you give the students any acronyms, abbreviations, symbols, or uncommon terms in the project report then their meaning should be explained where they first occur. If they go on to use any of them extensively then it is helpful to list them in this section and define the meaning.

APPENDICES

These may be provided to include further details of results, mathematical derivations, certain illustrative parts of the program code (e.g., class interfaces), user documentation etc.

In particular, if there are technical details of the work done that might be useful to others who wish to build on this work, but that are not sufficiently important to the project as a whole to justify being discussed in the main body of the project, then they should be included as appendices.

VI. SUMMARY

Project development usually involves an engineering approach to the design and development of a software system that fulfils a practical need. Projects also often form an important focus for discussion at interviews with future employers as they provide a detailed example of what the students are capable of achieving. In this course the

students can choose your project topic from the lists given in Unit 4: Category-wise Problem Definition.

VII. FURTHER READINGS

1. Modern Systems Analysis and Design; Jeffrey A. Hoffer, Joey F. George, Joseph,S. Valacich; Pearson Education; Third Edition; 2002.

2. ISO/IEC 12207: Software Life Cycle Process (http://www.software.org/quagmire/descriptions/iso-iec12207.asp).

3. IEEE 1063: Software User Documentation (http://ieeexplore.ieee.org).

4. ISO/IEC: 18019: Guidelines for the Design and Preparation of User Documentation for Application Software.

5. http://www.sce.carleton.ca/squall.

6. http://en.tldp.org/HOWTO/Software-Release-Practice-HOWTO/documentation.html.

7. http://www.sei.cmu.edu/cmm/

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

(Note:All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

PNR No.: ..... Rollno: _____

1. Name of the Student

_____

2. Title of the Project

_____

3. Name of the Guide-
4. Teaching experience of the Guide _____
5. Is this your first submission? Yes No

Signature of the Student
Date: .....

Signature of the Guide
Date: .....

Signature of the Coordinator
Date: .....

(All the text in the report should be in times new roman)

TITLE OF THE PROJECT
(NOT EXCEEDING 2 LINES, 24 BOLD, ALL CAPS) A Project Report (12 Bold)

Submitted in partial fulfillment of the
Requirements for the award of the Degree of (size-12)

BACHELOR OF SCIENCE
(INFORMATION TECHNOLOGY) (14 BOLD, CAPS)
By(12 Bold)

Name of The Student (size-15, title case)
Seat Number (size-15)

Under the esteemed guidance of (13 bold)
Mr./Mrs. Name of The Guide (15 bold, title case)
Designation (14 Bold, title case)

COLLEGE LOGO
DEPARTMENT OF INFORMATION TECHNOLOGY(12 BOLD, CAPS)
COLLEGE NAME (14 BOLD, CAPS)
(Affiliated to University of Mumbai) (12, Title case, bold, italic)
CITY, PIN CODE(12 bold, CAPS)
MAHARASHTRA (12 bold, CAPS)

YEAR (12 bold)

COLLEGE NAME (14 BOLD, CAPS)
(Affiliated to University of Mumbai) (13, bold, italic)
CITY-MAHARASHTRA-PINCODE(13 bold, CAPS)
DEPARTMENT OF INFORMATION TECHNOLOGY (14 BOLD, CAPS)

College Logo

CERTIFICATE (14 BOLD, CAPS, underlined, centered)

This is to certify that the project entitled, **"Title of The Project "**, is bonafied work of **NAME OF THE STUDENT** bearing Seat.No: (**NUMBER**) submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai. (12, times new roman, justified)

Internal Guide (12 bold) **Coordinator**

(Don‟t write names of lecturers or HOD)

External Examiner

Date: **College Seal**

COMPANY CERTIFICATE (if applicable)

(Project Abstract page format)

Abstract (20bold, caps, centered)

Content (12, justified)

Note: Entire document should be with 1.5 line spacing and all paragraphs should start with 1 tab space.

ACKNOWLEDGEMENT

(20, BOLD, ALL CAPS, CENTERED)

The acknowledgement should be in times new roman, 12 font with 1.5 line spacing, justified.

(Declaration page format)

DECLARATION (20 bold, centered, allcaps)

Content (12, justified)

I hereby declare that the project entitled, "**Title of the Project**" done at the place **where the project is done**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Name and Signature of the Student

TABLE OF CONTENTS (20 bold, caps, centered) Should be generated automatically using word processing software.

List of Tables (20 bold, centered, Title Case)

Should be generated automatically using word processing software.

List of Figures (20 bold, centered, Title Case)
Should be generated automatically using word processing software.

***** The plagiarism should be maintained as per the UGC guidelines.***

**** NOTE ABOUT PROJECT VIVA VOCE:**

Students may be asked to write code for problems during VIVA to demonstrate his coding capabilities and he/she may be asked to write any segment of coding used in the project.
The project can be done in groups of at most four students. However, the length and depth of the project should be justified for the projects done in group. A big project can be modularised and different modules can be assigned as separate projects to different students.

Marks Distribution:
Semester V: 50 Marks
Documentation: 50 marks

Semester VI: 150 Marks
Documentation: 50 Marks:
Implementation and Viva Voce: 100 Marks

Evaluation Scheme for Third Year (UG) under NEP 2020

I. Internal Evaluation for Theory Courses – 20 Marks

1) Continuous Internal Assessment(CIA) Assignment – 10 marks

2) Continuous Internal Assessment(CIA) ONLINE Unit Test – 10 marks

II. External Examination for Theory Courses – 30 Marks

Duration: 1 Hours

Theory question paper pattern: All questions are compulsory.

Question	Based on	Marks
Q.1	Unit I	15
Q.2	Unit II	15

- Each Question may be subdivided into sub questions as A, B, C, D & P, Q, R, S and the allocation of Marks depends on the weightage of the topic and course outcomes.
- Students have to attempt either A, B, C, D or P, Q, R, S sub-questions.

Paper Pattern of Theory Paper:

DES's Kirti M. Doongursee College (AUTONOMOUS), Dadar (W), Mumbai-28			
Regular / Additional / ATKT Examination			
Duration: 1 Hour			**Max Marks: 30**
Date:		**Time:**	**Code:**
(For office use)			
N. B.	i)	*All Questions are compulsory.*	
	ii)	*Mixing of sub-questions is not allowed*	
	iii)	*Draw neat labeled diagrams wherever necessary.*	
Q. No.			**Marks**

Q.1 A)		**04**
Q.1 B)		**04**
Q.1 C)		**04**
Q.1 D)		**03**
OR		
Q.1 P)		**04**
Q.1 Q)		**04**
Q.2 R)		**04**
Q.2 S)		**03**
Q.2 A)		**04**
Q.2 B)		**04**
Q.2 C)		**04**
Q.2 D)		**03**
OR		
Q.2 P)		**04**
Q.2 Q)		**04**
Q.2 R)		**04**
Q.2 S)		**03**

III. Practical Examination

- Each core subject carries 50 Marks (30 marks External + 20 marks Internal)

Sr. No.	Undergraduate Practical Internal Evaluation:	Marks
1	Short Experiment/Field Trip/Excursion/Industrial Visit Report	15
2	Journal	5

Sr. No.	Undergraduate Practical External Evaluation:	Marks
1	Experiment/s	25
2	Viva	5

- 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.

Note:
Two short field excursions for habitat studies are compulsory. Field work of not less than eight hours duration is equivalent to one period per week for a batch of 15 students.

- A candidate will be allowed to appear for the practical examinations only if he/she submits a **certified journal** of T.Y.B.Sc. or a certificate from the Head of the department / Institute to the effect that the candidate has completed the practical course of T.Y.B.Sc. as per the minimum requirements. In case of loss of journal, a candidate must produce a certificate from the Head of the department/Institute that the practicals for the academic year were completed by the student. However, such a candidate will be allowed to appear for the practical examination, but the marks allotted for the journal will not be granted.

NOTE:

1. **To pass the examination, attendance is compulsory in both Internal & External (Theory + Practical) Examinations.**
2. **There is separate passing in internal and external theory courses.**

****