Kurukshetra University, Kurukshetra

(Established by the State Legislature Act-XII of 1956)
("A++" Grade, NAAC Accredited)

Syllabus
for

Post Graduate Programme

M.Sc. Computer Science (Software)
as per NEP-2020
Curriculum and Credit Framework for Postgraduate Programme

CBCS-LOCF
(3rd and 4th Semesters)
With effect from the session 2025-26

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS
FACULTY OF SCIENCES

KURUKSHETRA UNIVERSITY, KURUKSHETRA -136119

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	3rd		
Name of the Course	Mobile Applications Development		
Course Code	M24-CSE-301		
Course Type	CC-9		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The course aims to provide a comprehensive understanding of mobile application development, focusing on the Android platform. It will cover fundamental concepts, user interface design, data persistence, and advanced features like location-based services and graphics, equipping students with the skills to build robust and interactive mobile applications.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO 1:** Students will gain a foundational understanding of mobile application concepts and the Android ecosystem, including its architecture and development environment. **CLO 2:** Students will learn to design and implement interactive user interfaces using various UI components and handle user events effectively, along with understanding how to use Intents for inter-component communication and deploy applications. **CLO 3:** Students will be able to implement modular and multi-screen user interfaces using Fragments and integrate location-based services and multimedia functionalities into their applications. **CLO 4:** Students will be able to incorporate graphics and manage data persistence using file storage and SQLite databases, including interacting with Content Providers.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Introduction:** Mobile Applications, Characteristics and Benefits, Application Models, Mobile devices Profiles. Basics of Android, Importance and scope, Android Versions, Features of Android, Android Architecture, Android Stack, Android Applications Structure, Android Emulator, Android SDK, Overview of Android Studio, Android and File Structure, Android Virtual Device Manager, DDMS, LogCat, Understanding Activities. **Android User Interface:** Measurements – Device and pixel density independent measuring units. Layouts – Linear, Relative, Grid and Table Layouts.	15

II	**User Interface (UI) Components** – Editable and non-editable Text Views, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers, List View, Spinner View. **Event Handling** – Handling clicks or changes of various UI components. **Intents and Broadcasts:** Intent – Using intents to launch Activities, Explicitly starting new Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS **Services**- Callbacks and Override in application, Application Signing, API keys for Google Maps, Publishing application to the Android Market.	15
III	**Fragments** – Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities **Location and Mapping:** location based services, Mapping, Google Maps activity, Working with MapView and MapActivity; Playing and Recording of Audio and Video in application; Sensors and Near Field Communication; Native libraries and headers, Building client server applications.	15
IV	**Using Graphics:** Canvas Drawing, Shadows, and Gradients. **Persisting Data to files:** Saving to Internal Storage, Saving to External Storage **Introduction to SQLite database:** creating and opening a database, creating tables, inserting retrieving and deleting data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update)	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) Zigurd Mednieks, Laird Dornin, G,Blake Meike and Masumi Nakamura “Programming Android”, O’Reilly Publications.
2) Wei-Meng Lee, “Beginning Android Application Development”, Wiley India Ltd.
3) James C.S. “Android Application development for Java Programmer”, CENGAGE Learning.
4) Pradeep Kothari, “Android Application Development: Black Book”, Wiley India Ltd.
5) Gargenta M., Nakamura M., “Learning Android”, O’Reilly Publications.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	3rd		
Name of the Course	Machine Learning using Python		
Course Code	M24-CSE-302		
Course Type	CC-10		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The course aims to equip students with a strong foundation in Python programming, covering core data structures, file handling, and error management. It will then delve into essential Python libraries for data manipulation, analysis, and visualization, culminating in an introduction to various machine learning algorithms and their theoretical underpinnings.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Students will master fundamental Python data structures like strings, lists, tuples, and dictionaries, along with file operations and exception handling. **CLO-2:** Students will gain proficiency in using key Python libraries (Pandas, NumPy, Scikit-learn, Matplotlib) for data manipulation, analysis, visualization, and implementing various machine learning algorithms. **CLO-3:** Students will develop an understanding of core machine learning concepts, including problem definition, system design, and algorithms like Concept Learning and Decision Trees. **CLO-4:** Students will comprehend advanced machine learning topics such as Bayesian learning, computational learning theory, and instance-based learning algorithms.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Python Programming: Strings - String slices, immutability, string functions and methods, string module; Lists, Tuples, Dictionaries: Lists - Lists as arrays Traversing a List, list operations, list slices, list methods, Map, Filter and Reduce, list loop, mutability, aliasing, cloning lists, list parameters; Dictionaries - operations and methods; advanced list processing - list comprehension; Tuples - tuple assignment, tuple as return value. Files and Modules: Files and exception - text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules.	15

II	Packages in Python: PANDAS, NUMPY, SCIKIT-LEARN, MATPLOTLIB. NumPy - Introduction, Ndarray Object ,Data types, Array Attributes, Array Creation Routines, Indexing & Slicing, Advanced Indexing, Broadcasting, Iterating Over Array, Array Manipulation, Binary Operators, String Functions, Mathematical Functions, Mathematical Functions, Arithmetic Operations, Statistical Functions, Linear Algebra, Matplotlib(Used for data visualization), Histogram Using Matplotlib. Pandas: Performing data cleaning and analysis, Loading data with Pandas (data manipulation and analysis), Working with and Saving data with Pandas. Using Scikit-Learn for Linear Regression, Logistic Regression, Decision Tree, Naive Bayes, KNN, SVN, k Mean Clustering, Random Forest.	15
III	Introduction to Machine Learning – Well defined learning problems, Designing a Learning System, Issues in Machine Learning. The Concept Learning Task - General-to-specific ordering of hypotheses, Find-S, List then eliminate algorithm, Candidate elimination algorithm, Inductive bias Decision Tree Learning - Decision tree learning algorithm-Inductive bias- Issues in Decision tree learning.	15
IV	Bayesian Learning: Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier, Bayesian belief networks, EM algorithm. Computational Learning Theory: Sample Complexity for Finite Hypothesis spaces, Sample Complexity for Infinite Hypothesis spaces, The Mistake Bound Model of Learning. Instance-Based Learning – k-Nearest Neighbour Learning, Locally Weighted Regression, Radial basis function networks, Case-based learning.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) Tom M. Mitchell, Machine Learning, McGraw-Hill Education (India) Private Limited.
2) Ethem Alpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press.
3) John V Guttag, Introduction to Computation and Programming Using Python, MIT Press.
4) Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python: An Interdisciplinary Approach, Pearson India Education Services Pvt. Ltd.
5) Stephen Marsland, Machine Learning: An Algorithmic Perspective, CRC Press.
6) Allen B. Downey, Think Python: How to Think Like a Computer Scientist, Updated for Python 3, Shroff/O,,Reilly Publishers.
7) Peter Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press.
8) Python Machine Learning, Sebastian Raschka.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	3rd		
Name of the Course	Data Mining and Analytics using R		
Course Code	M24-CSE-303		
Course Type	DEC-1		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The course aims to provide a comprehensive understanding of data warehousing and data mining concepts, techniques, and their practical application. It will cover data preprocessing, various data mining algorithms, and data analytics methodologies, concluding with an introduction to R programming for data analysis and mining.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Students will gain a foundational understanding of data warehousing concepts, architecture, and the knowledge discovery process, including various data preprocessing techniques and data visualization. **CLO-2:** Students will learn about key data mining techniques such as clustering, decision trees, and association rule mining, along with their underlying principles and performance considerations. **CLO-3:** Students will develop an understanding of data analytics methodologies, including qualitative and quantitative data analysis, data strategies, and exploratory data analysis techniques, as well as working with external data sources. **CLO-4:** Students will acquire practical skills in R programming for data manipulation, exploration, and the application of data mining algorithms like decision trees and clustering.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Data Warehouse: A Brief History, Characteristics, Architecture for a Data Warehouse. Fact and Dimension Tables, Data Mining: Introduction, Motivation, Importance, Knowledge Discovery Process, Data Mining Functionalities, Interesting Patterns, Classification of Data Mining Systems, Major issues, Data Preprocessing: Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization, Data Visualization, Outliers.	15
II	Data Mining Techniques: Statistical Perspective on Data Mining, Similarity Measures, Clustering-Requirement for Cluster Analysis, Clustering Methods, Decision Tree- Decision Tree Induction, Attribute Selection Measures, Tree Pruning. Association Rule Mining: Frequent Item-set Mining using Apriori Algorithm, Nearest Neighbour Classification: Performance of Nearest Neighbour Classifiers.	15
III	Data Analytics: Ways of Thinking About Data, Qualitative and Quantitative Data, And Data Strategies, Conceptualizing Data Analysis as a Process, Managing Data Analysis Process, Exploratory Data Analysis: Exploring a New Dataset, Summarizing Numeric Data, Anomalies in Numeric Data, Visualizing Relations between Variables. Working with External Data: Manual Data Entry, CSV Files, Other Files, Merging Data from Different Sources.	15

IV	R Programming: Advantages of R over other Programming Languages, Working with Directories and Data Types in R, Control Statements, Loops, Data Manipulation and integration in R, Exploring Data in R: Data Frames, R Functions for Data in Data Frame, Loading Data Frames, Decision Tree packages in R, Issues in Decision Tree Learning, Hierarchical and K-means Clustering functions in R, Mining Algorithm interfaces in R.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) J Hanes, M. Kamber, Data Mining Concepts and Techniques, Elsevier India.
2) Ronald K. Pearson, Exploratory Data Analysis Using R, CRC Press.
3) S. Acharya, Data Analytics Using R, McGraw Hill Education (India) Private Limited.
4) G.S. Linoff, M.J.A. Berry, Data Mining Techniques, Wiley India Pvt. Ltd.
5) Berson, S.J. Smith, Data Warehousing, Data Mining & OLAP, Tata McGraw-Hill.
6) J.Horbulyk, Data Integration Best Practices, elastic.io
7) Jared P. Lander, R For Everyone, Perason India Education Services Pvt. Ltd.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	3rd		
Name of the Course	Linux Administration		
Course Code	M24-CSE-304		
Course Type	DEC-1		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The course aims to provide a fundamental understanding of the Linux operating system, covering its architecture, file system, and essential command-line operations. It will then delve into managing users, groups, and permissions, followed by comprehensive coverage of process and package management. Finally, the course will equip students with knowledge of Linux networking, security configurations, and backup strategies.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Students will grasp the core architecture and file system hierarchy of Linux, enabling them to effectively navigate and manipulate files and directories using basic shell commands, and manage disk partitions and file systems. **CLO-4:** Students will be able to competently manage users and groups, understand and apply file permissions and ownership, including special permissions, and configure user environments while utilizing disk quotas and log file management. **CLO-4:** Students will master the concepts of process and job scheduling, manage system services, and effectively utilize various package management tools for software installation and maintenance, along with employing system monitoring utilities. **CLO-4:** Students will acquire the skills to configure Linux networking interfaces, implement firewall rules for security, set up and use SSH for secure remote access and file transfer, and understand basic backup strategies.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Introduction to Linux and File System:** Linux architecture and kernel overview; Boot process and run levels; Basic shell commands (ls, cp, mv, rm, cat, more, less, etc.); File system hierarchy (/, /home, /etc, /var, /tmp, etc.); File types and permissions; Creating and managing files and directories; Disk partitions; File systems (ext3, ext4, xfs); Mounting and unmounting file systems	15
II	**User and Group Management, Permissions:** Managing users and groups (useradd, usermod, passwd, groupadd, etc.); Understanding /etc/passwd, /etc/shadow, /etc/group; File permissions and ownership (chmod, chown, chgrp); Special permissions (SUID, SGID, Sticky Bit); User environment configuration files; Switching users (su, sudo); Disk quotas; Log files and log management (/var/log/)	15

III	**Process and Package Management:** Understanding processes (ps, top, nice, renice, kill, killall); Scheduling jobs (cron, at, anacron); System services (systemd, service, chkconfig); Package management using apt and dpkg (Debian-based systems); Package management using yum, dnf and rpm (Red Hat-based systems); Compiling and installing software from source; System monitoring tools (htop, iotop, vmstat, netstat, ss)	15
IV	**Networking, Security and Backup:** Linux networking basics (IP, ifconfig/ip, ping, netstat, ss); Configuring network interfaces and hostnames; Network configuration files; Firewall configuration using iptables and firewalld; SSH server and client configuration; File transfer using scp, rsync, sftp;	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) UNIX and Linux System Administration Handbook by Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, Dan Mackin – Pearson Education
2) Linux Administration: A Beginner's Guide by Wale Soyinka – McGraw-Hill Education
3) Linux Command Line and Shell Scripting Bible by Richard Blum and Christine Bresnahan – Wiley
4) Red Hat RHCSA/RHCE 8 Cert Guide: Red Hat Enterprise Linux 8 (EX200 and EX294) by Sander van Vugt – Pearson IT Certification
5) How Linux Works: What Every Superuser Should Know by Brian Ward – No Starch Press

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	3rd		
Name of the Course	Theory of Computation		
Course Code	M24-CSE-306		
Course Type	DEC-2		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The course aims to provide a rigorous theoretical foundation in the principles of computation, starting with finite automata and formal languages, progressing through pushdown automata and Turing machines, and culminating in an exploration of decidability, computability, and computational complexity.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Students will gain a deep understanding of finite state machines, including their design, equivalence, and the properties of regular languages and grammars, along with their limitations. **CLO-2:** Students will learn about formal grammars, specifically context-free grammars, their construction, properties, and simplification, alongside the introduction and design of Pushdown Automata and their relationship to CFGs. **CLO-3:** Students will comprehend the concept of Turing Machines as a general model of computation, exploring their capabilities as language acceptors and function computers, and understanding the nature of recursive and recursively-enumerable languages. **CLO-4:** Students will delve into advanced topics such as the Halting Problem, decidability of language properties, and the fundamentals of computable functions and computational complexity, including tractable and intractable problems.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Finite State Machines: Finite Automata, Designing of DFA and NDFA, NFA with E-Transitions, Equivalence of DFA and NFA with proof, Regular Expressions and Regular languages, Laws of Regular Expressions, Kleene's Theorem 1 and 2, Properties and Limitations of FSM FSM with Output: Moore and Mealy Machines, Arden's Theorem with proof, Closure Properties of Regular Sets, Pumping Lemma for Regular Grammers, Minimization of FA.	15
II	Formal Grammars: Definition, Construction of Regular & Context Free Grammar, Derivation, Parse Trees, Ambiguity, Removal of Ambiguity, Simplification of Context Free Grammar, CNF and GNF, Closure properties of CFL, Pumping Lemma for CFL. Pushdown Automaton: Introduction, Types of PDA, Designing of PDA's, Conversion from PDA to CFG and vice-versa.	15

III	Linear Bounded Automata (LBA), Turing Machines (TM), General Model of Computation, TM as Language Acceptors, TM as Computing Partial Functions, Combining TM, Multi-Tape TM, Restricted and Universal TM; TM and Computers. Recursive and recursively-enumerable languages and Properties, More General Grammars	15
IV	Reductions and the Halting Problem, Post's correspondence problem, Rice's theorem, Cook's Theorem, decidability of membership, emptiness and equivalence problems of languages, Decidable languages and problems, Diagonalization method. Computable Functions: Primitive recursive functions, Godel Numbering, Tractable and Intractable problems, Computable Complexity.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) John C. Martin, Introduction to Languages and the Theory of Computation, McGraw Hill.
2) Peter Linz, An introduction to formal language & automata, Jones & Bartlett publications.
3) Hopcroft J. E. & Ullman J. D, Formal languages and their relation to Automata, Pearson Education.
4) Lewis, H.R. & Papadimitrious, C. H., Elements of the theory of computation, PHI Learning.
5) Michael Sipser, Introduction to the Theory of Computation, Cengage Learning.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	3rd		
Name of the Course	Principles of programming Languages		
Course Code	M24-CSE-307		
Course Type	DEC-2		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The course aims to provide a comprehensive understanding of programming language design and implementation. It will cover fundamental concepts such as language syntax and semantics, formal language theory, data types, object-oriented principles, control flow mechanisms, and various advanced topics including concurrency and distributed processing.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Students will gain a foundational understanding of programming language history, design principles, translation processes, and formal syntax description using BNF grammars and parsing techniques. **CLO-2:** Students will learn about formal languages and automata theory, including the Chomsky hierarchy, and delve into language semantics through attribute grammars and denotational semantics, alongside a detailed study of data objects and types. **CLO-3:** Students will acquire a deep understanding of object-oriented programming concepts such as inheritance, polymorphism, and abstract data types, and explore various sequence control mechanisms within programs and subprograms. **CLO-4:** Students will be introduced to advanced and miscellaneous topics in programming languages, including parameter passing, scoping rules, storage management, exception handling, and an overview of parallel, network, and scripting paradigms.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Preliminaries: History, Impact of Programming Paradigms, Role of Programming Languages, Good Language, Effects of Programming Environment, Translators and virtual architectures, Binding and Binding time, Language Syntax, Analysis of Program, Synthesis of Object program, Formal translation models: BNF Grammars, General parsing, Language translation, Recursive descent parsing.	15
II	Formal languages and automata: The Chomsky hierarchy of formal languages, regular grammars, Regular expressions, Finite State Automata, Context-free grammars, Pushdown automata, Ambiguous grammars. Language Semantics: Attribute grammars, Denotational semantics, Program verification and validation, Data objects, variables, constants, data types, declaration, type checking, type casting, type promotion, Enumerators, Composite data types.	15

III	Object Orientated concepts: Structured data types, Abstract data types, Information hiding, Subprogram concepts, Good program design, Type definitions, Type equivalence, Inheritance, Derived classes, Abstract classes, Polymorphism, Inheritance and software reuse. Sequence control: Implicit and explicit sequence control, Sequence control within arithmetic expressions, sequence control between statements, sequencing with non-arithmetic expressions, Subprogram Sequence control.	15
IV	Miscellaneous topics: Parameter passing techniques, Static & Dynamic Scoping, Storage of variables, Static storage, Heap Storage management, Distributed Processing, Exceptions and Exception handlers, Co-routines, Scheduled subprograms, Parallel programming, Processor design, Hardware and Software architectures, Network Programming, Evolution of scripting languages, Applets, XML.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) Pratt T.W., Zelkowitz M.V., Gopal T.V., Programming Languages Design and Implementation, Pearson Education.
2) Sebesta W. Robert, Concepts of Programming Languages, Pearson Education.
3) Appleby Doris & Vande Kopple J. Julius, Programming Languages-Paradigm and practice", Tata McGraw Hill.
4) Sethi Ravi, Programming Languages: Concepts & Constructs, Pearson Education
5) Scott M., Programming Language Pragmatics, Elsevier India.

With effect from Session: 2024-25			
Part A - Introduction			
Name of the Programme	M. Sc. Computer Science (Software)		
Semester	3rd		
Name of the Course	Practical-5		
Course Code	M24-CSE-309		
Course Type	PC-5		
Level of the course	500-599		
Pre-requisite for the course (if any)			
Course objectives	This practical course aims to provide students with hands-on experience in developing Android applications. Students will learn to set up their development environment, design various user interfaces using different layout managers and UI components, handle user interactions, and implement inter-component communication using Intents. Furthermore, they will gain practical skills in incorporating advanced features such as background services, fragments, Google Maps integration, multimedia playback, and data persistence using file I/O and SQLite databases, including interaction with Content Providers. This course aims to equip students with practical skills in both Linux system administration and data analytics using R. Students will learn to navigate and manage Linux file systems using essential shell commands, administer users and groups, configure file and directory permissions, manage processes, and schedule tasks using tools such as cron and at.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Design and implement user interfaces in Android using various layout managers to create interactive components such as login forms, calculators, radio buttons, and dropdowns. **CLO-2:** Demonstrate the ability to handle user interaction, activity transitions, fragment management, and basic data storage and retrieval using event listeners, intents, fragments, and SQLite in Android applications. **CLO-3:** Apply data warehousing and preprocessing techniques in R, including schema design, handling missing values, normalization, discretization, and data integration, to prepare structured datasets for analysis. **CLO-4:** Use R packages and functions to implement and interpret data mining and analysis techniques such as decision trees, clustering, K-NN classification, association rule mining, and exploratory data visualization. **OR** **CLO-3:** Operate essential Linux commands to navigate the file system, manage files and directories, and explore system configuration, boot status, and kernel-level information. **CLO-4:** Administer users, groups, permissions, and scheduled tasks while effectively managing system processes, access control mechanisms, and system monitoring in a Linux environment.		
Credits	Theory	Practical	Total
	0	4	4
Teaching Hours per week	0	8	8
Internal Assessment Marks	0	30	30
End Term Exam Marks	0	70	70
Max. Marks	0	100	100
Examination Time	0	4 hours	

Part B- Contents of the Course	
Practicals	**Contact Hours**
Practical course will consist of two components Part-A and Part-B. The examiner will set 5 questions at the time of practical examination asking 2 questions from the Part-A and 3 questions from the Part-B by taking course learning outcomes (CLO) into consideration. The examinee will be required to solve one problem from the Part-A and to write and execute 2 programs from the Part-B.	120

Part-A

1) Design a simple login screen using `LinearLayout`. Include `EditText` for username and password, and two `Button`s for "Login" and "Cancel." Experiment with `android:orientation` (vertical/horizontal) and `android:layout_weight` to distribute space.
2) Design a basic calculator interface using `GridLayout`. Include `Button`s for numbers (0-9) and basic operations (+, -, *, /). Focus on layout arrangement, not functionality.
3) Create an application with a `Button`. When the button is clicked, change the text of a `TextView` to "Button Clicked!". Implement both an `OnClickListener` and a method in the XML layout for handling the click.
4) Design a layout with a `RadioGroup` containing three `RadioButton`s representing different programming languages (e.g., Java, Kotlin, Python). Display a `Toast` message indicating the selected language when a radio button is chosen.
5) Implement a `Spinner` (dropdown list) with a list of countries. When a country is selected, display the selected country in a `TextView`.
6) Create two activities. From the main activity, use an explicit `Intent` to launch the second activity when a button is clicked. Pass a simple string message from the first to the second activity.
7) Create an activity that hosts two `Fragment`s. Initially display one fragment. Add a button that, when clicked, replaces the currently displayed fragment with the other fragment using `FragmentTransaction`.
8) Create an Android application with a `Button` to insert a record into an SQLite database. Add another button to retrieve and display all records from the database in a `TextView` or `ListView`.

30

Part-B

1) Design and document a star schema, then simulate fact and dimension tables in R using `data.frame()`.
2) Perform data preprocessing in R: handle missing values, normalize data, and discretize using `cut()` or `scale()`.
3) Use the `arules` package to apply the Apriori algorithm on transaction data and generate association rules.
4) Build a decision tree using the `rpart` package, visualize it with `rpart.plot`, and interpret the model.
5) Apply K-means and hierarchical clustering using `kmeans()` and `hclust()` on a multivariate dataset, and plot the results.
6) Perform EDA using `summary()`, `str()`, `hist()`, and `ggplot2` to explore patterns and outliers.
7) Manipulate data frames using `dplyr` functions like `filter()`, `select()`, `mutate()`, and `arrange()`.
8) Implement K-NN classification using the `class` package and evaluate performance with a confusion matrix.
9) Read and merge datasets from CSV/Excel using `read.csv()` or `readxl`, then merge using `merge()` or `dplyr::left_join()`.
10) Detect and visualize outliers with `boxplot()`, `ggplot2`, and statistical functions like `scale()` or `IQR()`.

OR

1) Use `ls`, `cd`, `pwd`, `cat`, `cp`, `mv`, `rm`, `mkdir`, and `rmdir` to navigate and manipulate files across directories like `/home`, `/etc`, and `/tmp`.
2) Run commands like `uname -a`, `dmesg`, `runlevel`, and `systemctl` to explore the Linux kernel, boot process, and current run level.
3) Create a directory structure, assign read/write/execute permissions using `chmod`, and change ownership using `chown` and `chgrp`.
4) Add users with `useradd`, modify them using `usermod`, create groups with `groupadd`, and verify entries in `/etc/passwd`, `/etc/shadow`, and `/etc/group`.
5) Apply and test SUID, SGID, and Sticky Bit on executable files and shared directories to control access behavior.
6) Use `ps`, `top`, `htop`, `kill`, `nice`, and `renice` to monitor system processes, adjust priorities, and terminate specific jobs.
7) Schedule recurring tasks using `crontab -e` and one-time tasks using `at` and `anacron`; verify execution logs in `/var/log`.

90 (Lab hours include instructions for writing programs and demonstration by a teacher and for running the programs on computer by students.)

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Practicum**	**30**	➢ **Practicum**	**70**
• Class Participation:	5	Lab record, Viva-Voce, write-up and execution of the programs	
• Seminar/Demonstration/Viva-voce/Lab records etc.:	10		
• Mid-Term Examination:	15		

Part C-Learning Resources

Recommended Books:

1) Zigurd Mednieks, Laird Dornin, G,Blake Meike and Masumi Nakamura "Programming Android", O'Reilly Publications.
2) Wei-Meng Lee, "Beginning Android Application Development", Wiley India Ltd.
3) James C.S. "Android Application development for Java Programmer", CENGAGE Learning.
4) Pradeep Kothari, "Android Application Development: Black Book", Wiley India Ltd.
5) Gargenta M., Nakamura M., "Learning Android", O'Reilly Publications.

With effect from Session: 2024-25			
Part A - Introduction			
Name of the Programme	M. Sc. Computer Science (Software)		
Semester	3rd		
Name of the Course	Practical-6		
Course Code	M24-CSE-310		
Course Type	PC-6		
Level of the course	500-599		
Pre-requisite for the course (if any)			
Course objectives	This practical course aims to provide students with hands-on proficiency in core Python programming concepts, including string manipulation, list and dictionary operations, file handling, and exception management. Students will gain practical experience in leveraging powerful Python libraries such as NumPy for numerical computing, Pandas for data manipulation and analysis, and Matplotlib for data visualization. Furthermore, the course will equip students with the ability to implement and evaluate various machine learning algorithms using Scikit-Learn for tasks including regression, classification, and clustering.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Students will be able to perform advanced string manipulations, effectively utilize lists, tuples, and dictionaries for data organization and processing, and manage file input/output with robust exception handling. **CLO-2:** Students will be proficient in writing Python scripts that interact with command-line arguments, create and utilize custom modules, and perform efficient numerical operations, array manipulations, and statistical computations using NumPy. **CLO-3:** Students will be able to load, clean, analyze, and save data using Pandas DataFrames, and create informative data visualizations including line plots, histograms, and scatter plots using Matplotlib. **CLO-4:** Students will gain practical experience in implementing and evaluating a range of supervised and unsupervised machine learning algorithms, including linear and logistic regression, decision trees, KNN, Naïve Bayes, SVM, K-Means clustering, and Random Forest, using the Scikit-learn library.		
Credits	Theory	Practical	Total
	0	4	4
Teaching Hours per week	0	8	8
Internal Assessment Marks	0	30	30
End Term Exam Marks	0	70	70
Max. Marks	0	100	100
Examination Time	0	4 hours	

Part B- Contents of the Course	
Practicals	**Contact Hours**
Practical course will consist of two components Part-A and Part-B. The examiner will set 5 questions at the time of practical examination asking 2 questions from the Part-A and 3 questions from the Part-B by taking course learning outcomes (CLO) into consideration. The examinee will be required to solve one problem from the Part-A and to write and execute 2 programs from the Part-B.	120
Part-A 1) Write a Python program to extract substrings, reverse strings, and demonstrate string immutability using slicing and string methods. 2) Use Python's `string` module to perform operations like removing punctuation, checking character types, and converting case formats. 3) Create a list of integers, perform operations like append, insert, delete, slice, and use list comprehensions to generate a list of squares and even numbers. 4) Implement basic array-like operations using lists and demonstrate the difference between aliasing and cloning lists.	

5) Use `map()`, `filter()`, and `reduce()` to process a list of numbers (e.g., square, filter primes, calculate product of all elements). 6) Write a program that uses dictionaries to count character frequencies in a string and update values dynamically. 7) Demonstrate tuple assignment, packing and unpacking, and use a tuple as a return value from a function. 8) Write a program to read a text file, count word occurrences, and write the output to a new file using the `with` statement.	30
Part-B 9) Create a Python script that takes input from command line arguments and prints formatted results using `%` and `format()` methods. 10) Create a calculator program that gracefully handles exceptions such as division by zero and invalid input types using try-except blocks. 11) Create a custom module with mathematical functions and import it into another Python file for computation. 12) Create a NumPy ndarray and perform slicing, reshaping, broadcasting, and mathematical operations. 13) Use NumPy to compute mean, median, standard deviation, and solve a system of linear equations. 14) Generate line plots, histograms, and bar charts using `matplotlib.pyplot` for a given dataset. 15) Load a CSV dataset into a Pandas DataFrame, perform missing data handling, filtering, sorting, and summary statistics. 16) Modify and analyze data from a CSV file and save the cleaned and processed dataset into a new file. 17) Use Scikit-Learn to implement linear regression on a dataset and logistic regression on a binary classification dataset. 18) Implement Decision Tree, KNN, Naïve Bayes, and SVM classifiers using Scikit-Learn on a suitable dataset and evaluate performance using accuracy, precision, and recall. 19) Apply K-Means clustering on a synthetic dataset and visualize the clusters using `matplotlib`.	90 (Lab hours include instructions for writing programs and demonstration by a teacher and for running the programs on computer by students.)

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Practicum**	**30**	➢ **Practicum**	**70**
• Class Participation:	5	Lab record, Viva-Voce, write-up and execution of the programs	
• Seminar/Demonstration/Viva-voce/Lab records etc.:	10		
• Mid-Term Examination:	15		

Part C-Learning Resources

Recommended Books:

1) Tom M. Mitchell, Machine Learning, McGraw-Hill Education (India) Private Limited.
2) Ethem Alpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press.
3) John V Guttag, Introduction to Computation and Programming Using Python, MIT Press.
4) Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd.
5) Stephen Marsland, Machine Learning: An Algorithmic Perspective, CRC Press.
6) Allen B. Downey, Think Python: How to Think Like a Computer Scientist, Updated for Python 3, Shroff/O,,Reilly Publishers.
7) Peter Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press.
8) Raschka, S., & Mirjalili, V. (2017). Python Machine Learning (2nd ed.). Packt Publishing.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	3rd		
Name of the Course	Python Programming		
Course Code	M24-OEC-307		
Course Type	OEC		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	This course introduces the core concepts of Python programming to help students develop logical problem-solving skills using an easy-to-learn high-level language. Students will learn Python syntax, control structures, data types, functions, modular programming, and error handling. The course prepares students to write well-structured programs suitable for automation, academic research, and general-purpose computing tasks.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-I:** Students will be able to construct Python programs using variables, expressions, and control flow structures. **CLO-II:** Students will be able to manipulate and process data using Python's built-in data structures such as strings, lists, tuples, and dictionaries. **CLO-III:** Students will be able to design modular Python programs using functions and implement recursive solutions **CLO-IV:** Students will be able to reuse code through modules and handle runtime exceptions effectively.		
Credits	Theory	Tutorial	Total
	2		2
Teaching Hours per week	2		2
Internal Assessment Marks	15		15
End Term Exam Marks	35		35
Max. Marks	50		50
Examination Time	3		3

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Introduction to Python Programming:** History and features of Python; Installing Python and working with IDEs; Python syntax and indentation; Variables and data types; Type conversion; Input and output functions; Arithmetic, relational, logical and assignment operators; Conditional statements – if, if-else, if-elif-else; Looping structures – for loop, while loop; Loop control statements – break, continue, pass	15
II	**Strings and Lists:** String creation and indexing; String operations and built-in methods; String slicing and formatting; List creation, indexing, updating and deletion; List operations and methods; Nested lists; Iterating through strings and lists; List comprehensions	15
III	**Tuples, Sets and Dictionaries:** Tuples – creation, indexing and immutability; Tuple operations; Set creation and operations; Set methods and membership testing; Dictionary creation and access; Dictionary methods – update, pop, keys, values, items; Iterating through tuples, sets and dictionaries	15
IV	**Functions:** Defining and calling functions; Function arguments – positional, keyword, default, variable-length; Return values and scope of variables; Recursive functions; Lambda functions.	15
	Total Contact Hours	60

Suggested Evaluation Methods			
Internal Assessment: 15		**End Term Examination: 35**	
➢ **Theory**	**15**	➢ **Theory:**	**35**
➢ Class Participation:	4	Written Examination	

➢ Seminar/presentation/assignment/quiz/class test etc.:	4	
➢ Mid-Term Exam:	7	

Part C-Learning Resources

Reference Books

1) Learning Python by Mark Lutz – O'Reilly Media
2) Python Programming: An Introduction to Computer Science by John Zelle – Franklin, Beedle & Associates
3) Python Crash Course by Eric Matthes – No Starch Press
4) Think Python: How to Think Like a Computer Scientist by Allen B. Downey – O'Reilly Media
5) Automate the Boring Stuff with Python by Al Sweigart – No Starch Press

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	4th		
Name of the Course	Object Oriented Analysis and Design using UML		
Course Code	M24-CSE-401		
Course Type	DEC-3		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The course aims to provide a comprehensive understanding of the Unified Modeling Language (UML) and its application in software engineering. It will cover various UML diagrams for structural and behavioral modeling, as well as essential concepts in requirement engineering and system design.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1** Understands basics of modelling and UML and applying usecase modeling for requirements elicitation **CLO-2** Understand and apply concepts of class modeling and advanced structural modeling through UML diagrams. **CLO-3** Understand and apply behavioural modeling through state diagram, sequence diagram, activity diagram to represent software systems. **CLO-4** Have a working ability and grasping attitude to analyse and design software systems based on object-oriented thinking using UML.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Introduction to UML:** Principles of Modeling, Conceptual Model of UML - Things, Relationships, Diagrams; Mechanisms in The UML – Specifications, Adornments, Common Divisions, Extensibility Mechanisms. **Usecase Modeling:** Identifying Use Cases & Actors, Relationships - between Actors, between Use Cases and between Actor and Use Case, Elements of Use Case Diagram, Constructing Use Case Diagram. **Requirement Engineering:** Gathering and Documenting Requirements, Functional Vs. Non-Functional Requirements, Requirement Validation and Verification.	15
II	**Structural Modeling with UML:** Class Modeling - Generalization & Inheritance, Association, Multiplicity, Sets, Bags & Sequences, Role Names, Qualified Association, Aggregation, Composition, Association Attributes, Association Classes, Abstract Class, Metadata, Reification, Elements of Class Diagrams, Constructing Class Diagrams. **Advanced Structural Modeling with UML:** Object Diagrams - Instances and Links; Package Diagrams - Organizing Classes, Components Diagram, Deployment Diagram.	15

III	**Behavioral Modeling with UML:** State Modeling - Events, States, Transitions & Conditions, Activity Effects, Do-Activities, Entry & Exit Activities, Completion Transitions, Elements of State Diagrams, Nested State Diagrams, Concurrency, Constructing State Diagrams. **Sequence Diagram:** Lifelines, Messages, and Activations, Constructing Sequence Diagram and Communication Diagram **Activity Diagram:** Actions, Control Flows, and Object Flows, Swim Lanes, Constructing Activity Diagram	15
IV	**System Design:** Estimating Performance, Make A Reuse Plan, Organize The System into Subsystem, Identifying Concurrency, Allocating Subsystems to Processors and Tasks, Management of Data Stores, Handling Global Resources, Choosing Software Control Strategies, Handling Boundary Conditions, Setting Trade-Off Priorities, Selecting an Architect Style. **Class Design:** bridging gap, realize use cases with operations, designing algorithms, design optimization, adjustment of inheritance.	15
	Total Contact Hours	60

Suggested Evaluation Methods			
Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Textbook

1) Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Pearson education.
2) M. Blaha, J. Rumbaugh, Object-Oriented Modeling and Design with UML, Pearson Education.

Reference Books

1) Brend Bruegge, Allen H. Dutoit, Object-Oriented Software Engineering, Using UML, Patterns and Java, Pearson Education.
2) J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, Object-Oriented Modeling and Design, Prentice Hall of India.
3) Satzinger, Jackson, Burd, Object-Oriented Analysis & Design with the Unified Process, Thomson.
4) Grady Booch, Object Oriented Analysis & Design, Pearson Education.
5) Craig Larman , Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Prentice Hall

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	4th		
Name of the Course	Computer Graphics and Animation		
Course Code	M24-CSE-402		
Course Type	DEC-3		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The objective of this course is to provide students with a solid foundation in the principles, techniques, and applications of computer graphics. The course covers both 2D and 3D graphics, focusing on fundamental concepts such as raster and random scan systems, graphics hardware, drawing algorithms, transformations, and curve generation. Students learn how to create and manipulate graphical primitives, perform geometric and viewing transformations, and apply clipping algorithms. Advanced topics include 3D modeling, projections, visible surface detection, lighting, shading, and animation techniques. Through this course, students will develop the skills needed to build interactive visual applications and understand the role of graphics in modern computing fields such as augmented reality, computer vision, and animation.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Students will understand the basic components, functions, and devices of computer graphics and explore its practical applications. **CLO-2:** Students will implement scan conversion algorithms for lines, circles, and ellipses and use curve and area-filling techniques to create graphic primitives. **CLO-3:** Students will apply 2D geometric transformations, clipping techniques, and coordinate system manipulations for effective 2D rendering. **CLO-4:** Students will explore 3D modeling, transformations, surface rendering, and animation techniques to simulate realism and motion in computer graphics.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Introduction to Computer Graphics and its Components:** Overview of Computer Graphics, its functions & elements; Introduction to GUI, Computer Vision, Augmented Reality and other Applications of Graphics; Popular Graphics Software; Components and Working of Interactive Graphics; Raster Scan and Random Scan systems and Display Processors; Look-up table; Loading the Frame Buffer; Coordinate Systems. **Graphics Devices**: Display Technologies: Resolution, Aspect Ratio, Refresh CRT, Color CRT, Flat Panel Displays; Interactive Input Devices for Graphics, Image and Video Input Devices.	15

II	**Scan Conversion:** Drawing Geometry; Output Primitives; Lines and Pixel Graphics; AntiAliasing; Scan Converting Lines: DDA line drawing algorithms, Bresenham's line Algorithm; Scan Converting Circles: Polynomial method for circle drawing, circle drawing using polar coordinates, Bresenham's circle drawing; Algorithms for Generation of ellipse; Line Styles; Generation of Bar Charts, Pie-Charts. **Curve Representation**: Parametric Curves, Parametric Representation of a Circle, Parametric representation of cubic curves, drawing Bezier curves. **Filled-Area Primitives**: Basic Stack based fill algorithms: Flood fill algorithm, Boundary fill algorithm; Scan-line polygon fill algorithm and its computational structures.	15
III	**Two-Dimensional Transformations**: Coordinate and Geometric Transformations; Translation, Rotation, Scaling; Matrix representations and Homogeneous coordinates, Composite transformations, General Pivot Point rotation, General Fixed Point Scaling, Shearing; Reflection ; Reflection about an arbitrary line. **2-D Viewing**: Viewing pipeline; Window, Viewport, Window-to-Viewport transformation; Zooming, Panning; Pointing and Positioning techniques; Rubber band technique; Dragging. **Clipping operations**: Point and Line clipping, Cohen-Sutherland line clipping, Mid-Point Subdivision line clipping, Liang-Barsky line clipping, Sutherland-Hodgman polygon clipping; Weiler-Atherton polygon clipping.	15
IV	**3-D Graphics & Modeling**: Visualization techniques for Realism; 3D Object Representation; Solid Model Representation Schemes; Euclidean Geometry methods: Regularized Boolean Set Operations, Primitive Instancing, Boundary Representations, Curved lines and surfaces, Sweep Representations, Spatial-Partitioning Representations - Octree representation, Constructive Solid Geometry; Procedural Methods: Fractals, Shape Grammars, Particle systems, Physically Based modeling, Visualization techniques; 3D transformations. **Three-Dimensional Viewing:** Viewing Pipeline**;** Parallel Projection: Orthographic and Oblique Projection; Perspective Projection. **Visible-Surface Determination:** Z-buffer, Depth-Sorting, Area Subdivision, BSP-Tree method; Ray casting. **Introduction to Animation**: Designing of Animation Sequences; Key-Frame Systems; Animation Techniques: Tweening, Morphing.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) Donald Hearn, M. Pauline Baker, Computer Graphics, Pearson Education.
2) J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes, Computer Graphics - Principles and Practice, Pearson Education.
3) Newmann & Sproull, Principles of Interactive Computer Graphics, McGraw Hill.
4) Rogers, David F., Procedural Elements of Computer Graphics, McGraw Hill.
5) Zhigang Xiang, Roy Plastock, Computer Graphics, Tata McGraw Hill.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	4th		
Name of the Course	Big Data Analytics		
Course Code	M24-CSE-403		
Course Type	DEC-3		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	This course aims to provide students with a comprehensive understanding of the foundational concepts of big data, its characteristics, and the challenges it presents. Students will learn various technologies and analytical techniques for processing, storing, and analyzing large datasets, including distributed computing frameworks and advanced analytical methods, enabling them to derive actionable insights from complex data.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-I:** Students will be able to articulate the fundamental concepts of big data, differentiate it from traditional data processing methods, and understand the Hadoop ecosystem and its components. **CLO-II:** Students will gain proficiency in distributed data processing using MapReduce and will be able to design and work with various types of NoSQL databases, including Cassandra. **CLO-III:** Students will learn to use Hive and Pig for data analysis and ETL processing, and will be introduced to the Apache Spark framework for efficient big data processing. **CLO-IV:** Students will be able to apply advanced analytical techniques, including clustering, association rule mining, regression, and classification, and will understand the applications of big data analytics in real-world scenarios.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction to Big Data; Characteristics of Big Data (Volume, Velocity, Variety, Veracity, Value); Evolution of Big Data; Big Data vs. Traditional Business Intelligence; Key roles in the Big Data ecosystem; Examples of Big Data Analytics; Data Analytics Lifecycle; Big Data Myths; Challenges with Big Data; Understanding Big Data Storage; Overview of High-Performance Architecture; Introduction to Hadoop; Why Hadoop? Limitations of RDBMS; RDBMS vs. Hadoop; History of Hadoop; Hadoop overview; Use cases of Hadoop; Hadoop Distributed File System (HDFS); Processing data with Hadoop; Managing resources and applications with Hadoop YARN (Yet Another Resource Negotiator).	15

II	MapReduce Programming Model: Introduction, Mapper, Reducer, Combiner, Partitioner, Searching, Sorting, Compression, Real-time applications using MapReduce, Data serialization and working with common serialization formats; Introduction to NoSQL Databases; Types of NoSQL Databases (Key-Value, Document, Column-Family, Graph); Advantages and disadvantages of NoSQL; NewSQL; SQL vs. NoSQL vs. NewSQL; Cassandra: Introduction, Features, Data types, CQLSH, Keyspaces, CRUD operations, Collections, Counter, TTL, Alter commands, Import and Export, Querying System tables.	15
III	Hive: Introduction to Hive, Hive architecture, Hive data types, Hive file format, Hive Query Language (HQL), User-Defined Functions (UDF) in Hive; Pig: The anatomy of Pig, Pig on Hadoop, Pig philosophy, Use case for Pig, ETL processing with Pig, Pig Latin overview, Data types in Pig, Running Pig, Execution modes of Pig, HDFS commands, Relational operators, Eval Functions, Complex data types, Piggy Bank, Word count example using Pig; Introduction to Apache Spark; Spark's advantages over MapReduce; Spark architecture; Resilient Distributed Datasets (RDDs); Spark Core, Spark SQL, Spark Streaming, MLlib, GraphX; Hands-on with PySpark/Scala for basic data processing.	15
IV	Analytical Theory and Methods: Clustering and associated algorithms, Association Rules, Apriori Algorithm, Candidate Rules, Applications of Association Rules, Validation and Testing, Diagnostics; Regression: Linear Regression, Logistic Regression, Additional Regression Models; Classification: Decision Trees, Naïve Bayes, Diagnostics of Classifiers, Additional Classification Methods; Time Series Analysis: Box-Jenkins methodology, ARIMA Model, Additional methods; Text Analysis: Steps, Text Analysis Example, Collecting Raw Text, Representing Text, Term Frequency-Inverse Document Frequency (TF-IDF); Introduction to machine learning for Big Data; Real-world case studies and applications of Big Data Analytics in various domains (e.g., healthcare, finance, social media, e-commerce).	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) "Big Data Analytics: A Practitioner's Approach" by V. Bhuvaneswari and T. Devi.
2) "Mining of Massive Datasets" by Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman.
3) "Big Data and Analytics" by Seema Acharya and Subhashini Chellappan.
4) "Hadoop: The Definitive Guide" by Tom White.
5) "Spark: The Definitive Guide: Big Data Processing Made Simple" by Bill Chambers and Matei Zaharia.
6) "Data Science from Scratch" by Joel Grus.
7) "Designing Data-Intensive Applications" by Martin Kleppmann.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M. Sc. Computer Science (Software)		
Semester	4th		
Name of the Course	Compiler Design		
Course Code	M24-CSE-405		
Course Type	DEC-4		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course objectives	The objective of this course is to provide students with a comprehensive understanding of the design and implementation of compilers, emphasizing the translation of high-level programming languages into machine code. The course covers all key phases of a compiler including lexical analysis, parsing, syntax-directed translation, intermediate code generation, symbol table management, error handling, and code optimization. Students will learn various parsing techniques, attribute grammars, and optimization strategies, equipping them with the theoretical foundation and practical skills necessary for compiler construction and language processing tools.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1** Acquire knowledge of different phases and passes of the compiler. **CLO-2** Understand the parser and its types i.e. Top-Down and Bottom-up parsers and construction of LL, SLR, CLR, and LALR parsing table. **CLO-3** Implement the compiler using syntax-directed translation method and get knowledge about the synthesized and inherited attributes. **CLO-4** Understand the target machine's run time environment, its instruction set for code generation and techniques used for code optimization		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Overview of the Translation Process**: Introduction to translation, Difference between interpreter, assembler and compiler, Types of Compilers, Analysis of the Source Program, Phases of a Compiler. **Lexical Analysis:** Role of Lexical Analyzer, Tokens, Patterns, Lexemes, Lexical Errors, Input Buffering, Specifications of Tokens, Recognition of Tokens, **Parsing**: Parsing, role of parser, context free grammar, derivations, parse trees, ambiguity, elimination of left recursion, left factoring.	15
II	**Parsing Techniques:** Top-down parsing - backtracking, recursive descent parsing, predictive parsers, LL (1) grammars. **Bottom-Up Parsing:** Definition of bottom-up parsing, handles, handle pruning, stack implementation of shift-reduce parsing, conflicts during shift-reduce parsing, Operator Precedence Parsing, LR grammars, LR parsers-simple LR, canonical LR(CLR) and Look Ahead LR (LALR) parsers, error recovery in parsing.	15
III	**Syntax Directed Translation**: Syntax directed definitions (SDD), Inherited and Synthesized attributes, Evaluation orders for SDD, Applications of syntax directed translation. **Intermediate Code Generation:** DAG for expressions, three address codes - quadruples and triples, types and declarations, translation of expressions, array references, type checking and conversions, translation of Boolean expressions and control flow statements.	15

IV	**Symbol Table:** Structure and features of symbol tables, symbol attributes and scopes. **Error Detection and Recovery**: Errors, Lexical-Phase Errors, Syntactic Phase Errors, Semantic Errors **Code Optimization:** Potential cases of Code Optimization, Optimization of basic blocks, Local and Global optimizations, Common optimization techniques: Folding, Copy propagation, Common Sub expression eliminations, Code motion, Frequency reduction, Strength reduction	15
	Total Contact Hours	60

Suggested Evaluation Methods			
Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Text Books:

1) Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman (2007), Compilers Principles, Techniques and Tools, 2nd edition, Pearson Education, New Delhi, India.

Reference Books:

1) Alfred V. Aho, Jeffrey D. Ullman (2001), Principles of compiler design, Indian student edition, Pearson Education, New Delhi, India.
2) Kenneth C. Louden (1997), Compiler Construction– Principles and Practice, 1st edition, PWS Publishing.
3) D.M.Dhamdhere, "Compiler Construction – Principles & Practice", Macmillan India
4) Andrew W. Appel (2004), Modern Compiler Implementation C, Cambridge University Press, UK.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M. Sc. Computer Science (Software)		
Semester	4th		
Name of the Course	Biometric Security		
Course Code	M24-CSE-406		
Course Type	DEC-4		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)			
Course Objectives	The objective of this course is to introduce students to the principles, technologies, and applications of biometric systems. It provides a detailed understanding of various biometric traits, including both physiological and behavioral characteristics, and the processes of enrollment, identification, and verification. The course covers performance evaluation, error analysis, and the design of unimodal and multimodal biometric systems along with sensor technologies. It also addresses biometric system security, ethical considerations, and real-world applications in government and commercial domains, preparing students to assess and implement secure biometric solutions.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1.** Understand the basics of biometric and its functionalities **CLO-2** Understand and analyze biometric systems with different traits. **CLO-3** Analyze the performance of biometrics by combining multiple traits and biometric sensors. **CLO-4** Understand various Biometric security issues & Applications		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction to Biometric; Biometric technologies, Biometric Enrollment, Identification & Verification process , Biometric advantages, Different Biometric traits, characteristics of ideal biometric, Comparison of different biometrics, Accuracy in biometric systems; Error rates, Biometric deformations.	15
II	Physiological and Behavioral Biometric; Different Physiological and Behavioral Biometrics, Fingerprint, Face, Iris Introduction, Recognition, Feature extraction and matching, IRIS segmentation- normalization, Gait feature extraction, Hand geometry; Behavioral Biometrics: Introduction, Features, classification and properties of behavioral biometrics; Signature, Keystroke dynamics, Voice etc., merits and their demerits, soft biometrics.	15
III	Multimodal Biometrics and Biometric Sensors; Multimodal and Multi-biometric systems, Biometric Integration strategies, Architecture, Level of fusion, Training and adaptability, examples of multimodal biometric systems, Performance evaluation. Different types of Biometric sensors and their use; Biometric sensor interoperability.	15
IV	Biometrics Security and Applications; Biometrics Security; Biometric system challenges, Attack on biometric system, Liveness detection, Cancelable biometrics. Biometric standards, Applications of biometrics; Government sector, Commercial sector, Privacy in biometrics, Biometric Ethics and Technology usage, DNA Biometrics.	15
	Total Contact Hours	60

Suggested Evaluation Methods			
Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Anil K. Jain, Arun Ross, and Karthik Nandakumar, “Introduction to Biometrics”, Springer.
2) Anil K Jain, Patrick Flynn and Arun A Ross, “Handbook of Biometrics”, Springer.
3) James Wayman, Anil Jain, Davide Maltoni, Dario Maio, “Biometric Systems, Technology Design and Performance Evaluation”,Springer.
4) S.Y. Kung, S.H. Lin, M.W.Mak, “Biometric Authentication: A Machine Learning Approach” Prentice Hall.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	4th		
Name of the Course	Cloud Computing and IoT		
Course Code	M24-CSE-407		
Course Type	DEC-4		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The objective of this course is to equip students with foundational and advanced knowledge of cloud computing and the Internet of Things (IoT). It covers cloud architecture, deployment and service models, virtualization, and key cloud services, along with their applications and security considerations. The course also introduces the concepts, architecture, and networking technologies of IoT, followed by in-depth study of IoT communication protocols, constrained environments, and device-level implementations using platforms like Arduino and Raspberry Pi, with a focus on associated security challenges.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1** Understand core issues of cloud computing and enabling technologies. **CLO-2** Learn services provided on cloud computing platforms & security challenges. **CLO-3** Understand concepts, architecture, applications & networking technologies of IoT. **CLO-4** Learn IoT-oriented communication protocols and security concerns.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Cloud Computing:** Definition and Characteristics of Cloud Computing, Evolution and Enabling Technologies, Benefits & Challenges, NIST Reference Architecture of Cloud Computing; Deployment Models, Service Models, Service Oriented Architecture (SOA). **Virtualization:** Virtualization and Its Role in Cloud Computing; Benefits & Drawbacks of Virtualization, Types of Virtualization, Server Virtualization, Hypervisor-Based Approaches, Virtualization Of - Operating System, Platform, CPU, Network, Application, Memory and I/O Devices.	15
II	**Cloud Computing Services & Applications:** Cloud Computing Platforms; Compute Services, Storage Services, Database Services, Applications Services, Queuing Services, E-Mail Services, Notification Services, Media Services, Content Delivery Services, Analytics Services, Deployment& Management Services, Identity & Access Management Services and their Case Studies. **Cloud Security and Compliance:** Security Challenges in The Cloud, Data Protection and Privacy, Compliance and Regulatory Issues.	15

III	**Internet of Thing (IoT):** Definition and Characteristics of IoT, Key Components of IoT (Sensors, Actuators, Devices). IoT Ecosystems and Architecture, Conceptual Framework, Common Applications of IoT. Modified OSI Model for IoT/M2M Systems, M2M Vs IoT. **Iot Networking Technologies:** NFC, RFID, Bluetooth BR/EDR and Bluetooth Low Energy, Zigbee, WiFi.	15
IV	**Iot Communication Technologies & Security Issues:** Constrained Nodes, Constrained Networks, Types of Constrained Devices, Low Power and Lossy Networks. Security Issues and Challenges in IoT. **IoT Protocols** – 6LoWPAN, QUIC Protocol; Data Protocols - MQTT, MQTT-SN, CoAP, AMQP. Introduction to Arduino and Raspberry Pi Boards.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Textbook

1) Rajkumar Buyya, James Broberg, Andrzej Goscinski, Cloud Computing – Principles and Paradigms, Wiley India Pvt. Ltd.
2) Sudip Misra, Anandarup Mukherjee, Arijit Roy, Introduction to IOT, Cambridge
3) Arshdeep Bahga, Vijay Madisetti, Cloud Computing – A Hands-on Approach, University Press.
4) Raj Kamal, Internet of Things - Architecture and Design Principles, McGraw Hills

Reference Books

1) Mayur Ramgir, Internet of Things – Architecture, Implementation and Security, Pearson
2) Kai Hwang, Geoffrey C.Fox, and Jack J. Dongarra, Distributed and Cloud Computing, Elsevier India Private Limited
3) Anthony T. Velte, Toby J. Velte and Robert Elsenpeter, Cloud computing: A practical Approach, McGraw Hill
4) Vijay Madisetti and Arshdeep Bahga, Internet of Things (A Hands-on Approach), VPT

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	4th		
Name of the Course	Security in Computing		
Course Code	M24-CSE-409		
Course Type	DEC-5		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The objective of this course is to provide students with a comprehensive understanding of computer security principles, challenges, and techniques used to protect systems, data, and networks. It introduces key concepts such as threats, attacks, cryptography, and secure programming, along with practical knowledge of securing databases, networks, internet communication, operating systems, and physical environments. The course also covers legal, ethical, and policy aspects of cybersecurity to prepare students for addressing real-world security issues.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO 1:** Students will understand the fundamental concepts of computer security, cryptography, and common program threats along with methods to mitigate them. **CLO 2::** Students will gain knowledge of database and network security mechanisms including access control, encryption, firewalls, and intrusion detection systems. **CLO 3:** Students will be able to evaluate and apply internet security protocols and secure various operating systems such as Linux and Windows. **CLO 4:** Students will analyze physical security risks, perform risk assessments, and understand ethical and legal considerations in cybersecurity.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Computer Security Concepts, Threats, Attacks and Assets, Security Functional Requirements, Security Architecture and Scope of Computer Security, Computer Security Trends and Strategies. Cryptography: Terminology and Background, Substitution Ciphers, Transpositions, Cryptanalysis, Program Security: Secure Program, Non-malicious Program Error, Viruses and other Malicious Code, Targeted Malicious Code, Control against Program Threats.	15
II	Database Security: Database Management System, Relational Databases, Database Access Control, Inference, Security Requirements, Reliability and Integrity, Sensitive Data, Database Encryption. Network Security: Threats in Network, Network Security Controls, and Firewall- Need for firewall, Characteristics, Types of firewall, Firewall Basing, Intrusion Detection System- Types, Goals of IDS, IDS strengths and Limitations.	15

III	Internet Security Protocols and Standards: Secure Socket Layer (SSL) and Transport Layer Security (TLS), IPv4 and IPv6 Security, Kerberos 672, X.509, Public Key Infrastructure. Linux Security Model, File System Security, Linux Vulnerability, Linux System Hardening, Application Security. Window Security Architecture, Windows Vulnerability, Windows Security Defense, Browser Defenses.	15
IV	Physical Security Threats, Physical Security Prevention and Mitigation Measures, Recovery form Physical Security Breaches, Security Auditing Architecture, Security Audit Trail, Security Risk assessment, Security Controls or Safeguard, IT Security Plan, Implementation of Controls, Cybercrime and Computer Crime, Intellectual Property, Privacy, Ethical Issues.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Charles. P. Pfleeger & Shari Lawrence Pfleeger, Security in Computing, Pearson Education.
2) William Stalling, Lawrie Brown, Computer Security Principles and Practice, Pearson Education.
3) Atul Kahate, Cryptography and Network Security, Tata McGraw-Hill Education

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M. Sc. Computer Science (Software)		
Semester	4th		
Name of the Course	Software Testing		
Course Code	M24- CSE -410		
Course Type	DEC-5		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	This course provides a foundational understanding of software testing principles, methodologies, and life cycles. It covers test design, execution, and automation using tools like Selenium and JUnit, along with quality assurance standards and performance testing techniques. Students also explore advanced testing approaches and emerging trends such as Agile, DevOps, and AI in testing to ensure effective and reliable software development.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1.** Explain the fundamentals of digital image and its processing **CLO-2.** Perform image enhancement techniques in spatial domain. **CLO-3.** Elucidate the mathematical modelling of image restoration and compression **CLO-4.** Apply the concept of image segmentation and morphological Image Processing.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Fundamentals of Software Testing:** Software testing principles, objectives, and types; Verification and validation; Software development life cycle (SDLC) vs. software testing life cycle (STLC); Testing methodologies: Black-box testing, White-box testing, Grey-box testing; Static testing techniques: Reviews, Walkthroughs, and Inspections; Dynamic testing techniques: Functional and Non-functional testing; Levels of testing: Unit testing, Integration testing, System testing, Acceptance testing.	15
II	**Test Design and Execution:** Test planning, Test strategies, and Test documentation; Test case design techniques: Equivalence partitioning, Boundary value analysis, Decision table-based testing, State transition testing, Use case-based testing, Error guessing; Test execution process; Test result analysis; Test automation concepts and tools; Introduction to Selenium, JUnit, TestNG, and other testing frameworks.	15
III	**Software Quality and Performance Testing:** Software quality assurance (SQA); ISO and CMMI standards for quality; Metrics for software testing; Defect life cycle and defect tracking tools; Performance testing: Load testing, Stress testing, Scalability testing, and Volume testing; Tools for performance testing: JMeter, LoadRunner; Security testing concepts and techniques; Usability testing and its importance.	15
IV	**Advanced Testing Techniques and Emerging Trends:** Mutation testing, Regression testing, Configuration testing, Compatibility testing, Recovery testing; Testing in Agile and DevOps environments; Continuous Integration and Continuous Testing; Cloud-based testing; AI and machine learning in software testing;	15

		Total Contact Hours	60
Suggested Evaluation Methods			
Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Glenford J. Myers, Corey Sandler, Tom Badgett – The Art of Software Testing, Wiley.
2) Paul C. Jorgensen – Software Testing: A Craftsman's Approach, CRC Press.
3) Srinivasan Desikan, Gopalaswamy Ramesh – Software Testing: Principles and Practices, Pearson.
4) William Perry – Effective Methods for Software Testing, Wiley.
5) Aditya P. Mathur – Foundations of Software Testing, Pearson.
6) Ron Patton – Software Testing, Sams Publishing.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M. Sc. Computer Science (Software)		
Semester	4th		
Name of the Course	Digital Image Processing		
Course Code	M24- CSE -411		
Course Type	DEC-5		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	This course aims to introduce students to the fundamental concepts and techniques of digital image processing. It covers key topics such as image formation, sampling, quantization, and pixel relationships. Students will learn methods for enhancing images in the spatial domain, restoring degraded images, and applying Fourier transform techniques. The course also explores color image processing, compression methods, morphological operations, and image segmentation techniques, preparing students to apply image processing in real-world applications.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1.** Explain the fundamentals of digital image and its processing **CLO-2.** Perform image enhancement techniques in spatial domain. **CLO-3.** Elucidate the mathematical modelling of image restoration and compression **CLO-4.** Apply the concept of image segmentation and morphological Image Processing.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Introduction:** Examples of fields that use digital image processing, fundamental steps in digital image processing, components of image processing system. Digital Image Fundamentals: A simple image formation model, image sampling and quantization, basic relationships between pixels**.** **Image enhancement in the spatial domain**: Basic gray-level transformation, histogram processing, arithmetic and logic operators, basic spatial filtering, smoothing, and sharpening spatial filters.	15
II	**Image restoration:** A model of the image degradation/restoration process, noise models, and restoration in the presence of noise–only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms; Introduction to the Fourier transform and the frequency domain, estimating the degradation function.	15
III	**Color Image Processing:** Color fundamentals, color models, pseudo color image processing, basics of full–color image processing, color transforms, smoothing and sharpening, **Image Compression:** Fundamentals, image compression models, error-free compression, lossy predictive coding, some basic compression methods	15
IV	**Morphological Image Processing**: Preliminaries, dilation, erosion, open and closing, basic morphologic algorithms, The Hit-or-Miss Transformation **Image Segmentation:** Detection of discontinuous, edge linking and boundary detection, thresholding, Hough Transform Line Detection and Linking, region–based segmentation.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		End Term Examination: 70	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Text Books:

1) Digital Image Processing, RafealC.Gonzalez, Richard E.Woods, Second Edition, Pearson Education/PHI.

Reference Books

1) Image Processing, Analysis, and Machine Vision, Milan Sonka, Vaclav Hlavac and Roger Boyle, Second Edition, Thomson Learning.
2) Introduction to Digital Image Processing with Matlab, Alasdair McAndrew, Thomson Course Technology
3) Computer Vision and Image Processing, Adrian Low, Second Edition, B.S.Publications
4) Digital Image Processing using Matlab, RafealC.Gonzalez, Richard E.Woods, Steven L. Eddins, Pearson Education.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	M.Sc. Computer Science (Software)		
Semester	4th		
Name of the Course	Research Ethics		
Course Code	M24-CSE-413		
Course Type	EEC		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The objective of this course is to instill a deep understanding of ethical principles and practices in scientific research, particularly in computer science. It aims to familiarize students with the moral dimensions of research conduct, including integrity, plagiarism, authorship, peer review, data management, and ethical issues in computer science such as privacy, intellectual property, and responsible innovation. The course prepares students to identify ethical dilemmas in research environments and to respond thoughtfully and responsibly in accordance with professional and institutional ethical standards.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1**: Students will be able to understand the meaning, scope, and importance of research ethics in scientific and technological advancement. **CLO-2:** Students will be able to identify and evaluate ethical issues related to research misconduct, authorship, and peer review. **CLO-3**: Students will be able to apply ethical practices in data management, privacy, and intellectual property within the context of computer science research. **CLO-4** Students will be able to analyze case studies and institutional frameworks to uphold research integrity and compliance.		
Credits	Theory	Tutorial	Total
	2		2
Teaching Hours per week	2		2
Internal Assessment Marks	15		15
End Term Exam Marks	35		35
Max. Marks	50		50
Examination Time	3		3

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Introduction to Research Ethics:** Definition, need and importance of ethics in research; Scope and objectives of ethical research; Research and society; Scientific temper and research culture; Core values – honesty, trust, fairness, accountability; Professionalism and responsibility in research	15
II	**Misconduct in Research and Ethical Publishing:** Research misconduct – fabrication, falsification, plagiarism; Ethical guidelines for authorship and publication; Responsibilities of authors, reviewers, and editors; Conflict of interest; Retraction and correction of published work; Case studies on research misconduct	15
III	**Ethical Issues in Computer Science Research:** Data privacy and confidentiality; Cybersecurity ethics; Ethical use of AI and machine learning systems; Responsible innovation; Open-source ethics; Software piracy and licensing; Intellectual property rights – copyrights, patents, trademarks in research	15
IV	**Institutional Frameworks and Case Studies:** Institutional ethics committees and review boards; National and international guidelines – IEEE, ACM, UGC, DST, and CSIR ethics codes; Ethical clearance procedures; Whistleblowing and protection; Case studies of ethical dilemmas and best practices in research ethics	15

		Total Contact Hours	60
Suggested Evaluation Methods			
Internal Assessment: 15		**End Term Examination: 35**	
➢ **Theory**	**15**	➢ **Theory:**	**35**
➢ Class Participation:	4	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	4		
➢ Mid-Term Exam:	7		

Part C-Learning Resources

Reference Books

1) Resnik, D. B. (2011). Ethics of Science: An Introduction. Routledge
2) Shamoo, A. E., & Resnik, D. B. (2015). Responsible Conduct of Research (3rd ed.). Oxford University Press
3) Beauchamp, T. L., & Childress, J. F. (2019). Principles of Biomedical Ethics (8th ed.). Oxford University Press
4) Steneck, N. H. (2007). ORI Introduction to the Responsible Conduct of Research. U.S. Government Printing Office
5) Bhatia, R. (2018). Research Methodology and Scientific Writing. Springer

With effect from the Session: 2024-25	
Name of Programme	M.Sc. Computer Science (Software)
Semester	4th
Name of the Course	Dissertation / Project
Course Code	M24-CSE-414
Course Type	Dissertation / Project
Level of the course	500-599
Pre-requisite for the course (if any)	
Course Objectives	The objective of the Dissertation/Project course is to provide the students with an opportunity to undertake an in-depth investigation of a contemporary research problem or to apply their knowledge and skills in a real-world industrial setting. The course is designed to develop independent research aptitude, analytical and problem-solving skills, and the ability to design and implement effective computational solutions. Students may choose to work on a research-oriented topic leading to a dissertation report, or on an industry-based project culminating in a project report. The emphasis is on fostering innovation, critical thinking, technical competence, and professional reporting and presentation skills, thereby preparing students for higher research, entrepreneurial ventures, or responsible professional roles in the IT industry.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO1:** Demonstrate the ability to identify, formulate, and analyze a research problem or industrial requirement in the domain of computer applications. **CLO2:** Apply advanced concepts, tools, and methodologies to design and implement effective computational solutions. **CLO3:** Develop independent research, project management, and technical documentation skills with professional standards. **CLO4:** Present findings and outcomes effectively through a well-structured dissertation/project report and oral defense before experts.
Credits	12
Max. Marks	300

Guidelines for Examiners for Dissertation:

1. Relevance of Topic – Assess whether the chosen problem/project is significant, original, and relevant to the field of Computer Applications.
2. Literature Review & Problem Formulation – Evaluate the depth of background study, clarity in objectives, and problem definition.
3. Methodology & Technical Content – Examine the appropriateness of research methods, tools, algorithms, and technologies employed.
4. Implementation & Results – Assess the quality of system design, implementation, experiments conducted, and accuracy/reliability of results.
5. Report Quality – Assess organization, clarity, formatting, referencing, and overall technical writing of the dissertation/project report.
6. Originality & Innovation – Look for independent thinking, novelty of the solution, and avoidance of plagiarism.
7. Viva-Voce / Presentation – Evaluate student's understanding, ability to defend work, and clarity in oral communication.

Guidelines for Examiners (Industrial Project Evaluation)

1. Problem Identification & Industry Relevance: Assess whether the project addresses a real-world industrial problem and its relevance to current practices/technologies.
2. Project Objectives & Scope: Evaluate clarity of objectives, scope definition, and alignment with industry requirements.
3. System Design & Methodology: Examine the design models, architecture, frameworks, and tools used for problem-solving.
4. Assess quality of coding, integration, testing, and overall execution of the project.
5. Assess clarity, completeness, and professionalism of the project report, including industry-specific standards followed.
6. Viva-Voce / Presentation: Evaluate student's ability to present the project, explain technical decisions, and respond to queries confidently.