Centenary Celebrated Sharnbasveshwar Vidya Vardhak Sangha's

Sharnbasva University

ಶರಣಬಸವ ವಿಶ್ವವಿದ್ಯಾಲಯ

A Private University enacted by Govt. of Karnataka as "Sharnbasva University Act 2012" Karnataka Act No. 17 of 2013. Notification No. ED 144 URC 2016 dated 29/07/2017

www.sharnbasvauniversity.edu.in Kalaburagi-585 103 - Karnataka - India Email : sharnbasvauniversity@gmail.com

Faculty of Engineering and Technology (Exclusively for Women)
Department of Computer Science and Engineering
B. Tech 3rd year (IV and V Semester)
Scheme of Teaching and Examination

Outcome Based Education (OBE) & Choice Based Credit System (CBCS)
(Effective from the academic year 2018-19)

1. PROGRAM EDUCATIONAL OBJECTIVES (PEOs) of the Department

PEO1 PEO 1: Graduates will possess a strong foundation in Computer Science and Engineering that are required for problem solving to excel and succeed in their profession.

PEO2 PEO 2: Graduates will have scientific and engineering breadth to comprehend, analyze, design and solve real life problems using the acquired skills and life long learning

PEO3 PEO 3: Graduates will have exposure to emerging cutting edge technologies and adequate training with opportunity to work on multidisciplinary projects.

PEO4 PEO 4: Graduates will be professional with ethical attitude,effective communication skills, teamwork capability and relate engineering issues to braoder social context

PROGRAMME OUTCOMES (POs) COMPUTER SCIENCE AND ENGINEERING (UG)

PO1 - Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2 - Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3 - Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4 - Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5 - Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6 - The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7 - Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8 - Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9 - Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10 - Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11 - Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments

PO12 - Life-long learning: Recognize the need for, and have the preparation and ability to engage independent and life-long learning in the broadest context of technological change.

2. PROGRAMME SPECIFIC OUTCOMES (PSOs)

On completion of the B. Tech. (Computer Science & Engineering) degree the graduates will be able to

PSO1: Apply principles of basic sciences and Engineering fundamentals in the field of Computer Science and Engineering

PSO2: Apply computational, algorithmic, and programming skills to implement solutions for real-life problems in diverse domain adapting to emerging technologies through lifelong learning

PSO3: Develop practical abilities, ethical understanding, effective communication and leadership skills for successful careers in industry or academia

Sharnbasva University, Kalaburagi **Scheme of Teaching and Examination 2018-19** **Outcome Based Education (OBE) and Choice Based Credit System (CBCS)** **(Effective from the academic year 2018-19)**					
Program: B. Tech Computer Science & Engineering					
V SEMESTER B. Tech.					
			Teaching Dept. & Paper Setting Board	**Teaching Hours/ week**	**Exa**

Sl. No	Course Code	Course Title			L	T	Duration in hours
1.	PCC	18CS51	Database Management System	CSE	3		3
2.	PCC	18CS52	Computer Networks	CSE	3		3
3.	PEC	18CS53X	Professional Elective – I	CSE	3		3
4.	OEC	18CS54X	Open Elective – I	CSE	3		3
5.	PCC	18CSL55	Web Programming Lab	CSE		2	3
6.	PCC	18CSL56	Database Management System lab	CSE		2	3
7.	PEC	18CSL57	Computer Networks lab	CSE		2	3
8.	PRJ	18CSP58	Project-V	CSE		2	3
9.	HSMC	18HSM59	Soft Skills	Humanities	1		3
Total					**08**		**27**

PCC-Professional Core, PEC- Professional Elective, OEC- Open Elective, HSMC-Humanity and Social Science,

Professional Elective – I			Open Elective – I		
Sl. No.	Sub. Code	Sub. Name	Sl. No.	Sub. Code	Sub. Name
1.	18CS531	Automata Theory and Computability	1.	18CS541	Discrete Mathematical Structures and Graph Theory
2.	18CS532	Cloud Computing	2.	18CS542	Microcontroller and Embedded Systems
3.	18CS533	UNIX System Programming	3.	18CS543	Mobile Application Development
4.	18CS534	Social Network Analysis	4.	18CS544	Green Technology

Sharnbasva University, Kalaburagi

Scheme of Teaching and Examination 2018-19

Outcome Based Education (OBE) and Choice Based Credit System (CBCS)

(Effective from the academic year 2018-19))

Program: B. Tech Computer Science & Engineering

VI SEMESTER B. Tech.

Sl.No	Course Code		Course Title	Teaching Dept. & Paper Setting Board	Examination						Credits
					T	P	Duration in hours	CIE Marks	SEE Marks	Total Marks	
1.	PCC	18CS61	System Software and Complier Design	3	1		3	50	50	100	04
2.	PEC	18CS62X	Professional Elective-II	3			3	50	50	100	03

3.	PEC	18CS63X	Professional Elective-III	3			3	50	50	100	03
4.	OEC	18XX64X	Open Elective –II	3			3	50	50	100	03
5.	PCC	18CSL65	System Software and Compiler Design Lab			2	3	50	50	100	01
6.	PEC	18CSL66	Operating System and UNIX system Programming Lab			2	3	50	50	100	01
7.	PEC	18CSL67	Python Lab			2	3	50	50	100	01
8.	PRJ	18CSP68	Project-6			2	3	50	50	100	01
9.	HSMC	18HSM69	Professional Ethics	1			3	50	50	100	01
Total	**13**				**08**		**450**		**450**	**900**	**18**
	PCC-Professional Core, PEC- Professional Elective, OEC- Open Elective, HSMC-Humanity and Social Science, PR-Project										

Professional Electiv	**Professional Elective - III**	**Open Elective – II**

e – II					
Sl. No.	**Sub.Name**	**Sub. Code**	**Sub. Name**	**Sub. Code**	**Sub. Name**
1.	Operating System	18CS631	Rapid Programming Application using Python	18CS641	Software Engineering
2.	Software Testing	18CS632	Sensors and Applications	18CS642	Multi Core Architecture
3.	Cryptography and Network Security	18CS633	Computer Vision	18CS643	Network Programming
4.	Computer Graphics and Visualization	18CS634	Probability Statistics andQueuing Theory	18CS644	Mobile Computing

DATABASE MANAGEMENT SYSTEM

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – V

Subject Code	**18CS51**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**48**	**Exam Hours**	03

CREDITS – 04

Course Objectives: This course will enable students

- To learn the fundamentals of data models and to represent a database system using ER diagrams.

To study SQL and relational database design.

- To understand the internal storage structures using different file and indexing techniques which will help in physical DB design.
- To understand the fundamental concepts of transaction processing- concurrency control techniques and recovery procedures.

To have an introductory knowledge about the Storage and Query processing Techniques

Modules	**Hours**
Module I	
Introduction to Databases: Introduction, Characteristics of database approach, Advantages of using the DBMS approach. History of database applications. **Overview of Database Languages and Architectures**: Data Models, Schemas, and Instances. Three schema architecture and data independence, database languages, and interfaces. **Conceptual Data Modeling using Entities and Relationships:** Entity types, Entity sets, attributes, roles, and structural constraints, Weak entity types, ER diagrams, examples, Specialization and Generalization.	10
Module II	
Relational Model: Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations. **Mapping Conceptual Design into a Logical Design**: Relational Database Design using ER-to-Relational mapping. SQL: SQL data definition and data types, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL. **SQL: Advances Queries**: More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL, Schema change statements in SQL.	10
Module III	
Database Application Development: Accessing databases from applications, An introduction to JDBC, JDBC classes and interfaces, SQLJ, Stored procedures, Case study: The internet Bookshop. **Functional Dependencies and Normalization for Relational Databases**: Informal Design Guidelines for Relation Schemas, Functional Dependencies, And Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Code Normal Form. And Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Code Normal Form.	08
Module IV	
Database Design Algorithms: Properties of Relational Decompositions, Algorithms for RelationalDatabase Schema Design, Nulls, Dangling tuples, Further discussion of Multivalued dependencies and 4NF, Other dependencies and Normal Forms **Transaction Management** – Introduction to Transaction processing, Transaction and system	10

concepts, Desirable properties of Transactions, characterizing schedules based on recoverability and Serializability.	
Module-V	
Concurrency Control in Databases: Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Mult version Concurrency control techniques, Validation Concurrency control techniques, Granularity of Data items and Multiple Granularity Locking. **Database Recovery Techniques**: Recovery Concepts, Recovery techniques based on Deferred update, Recovery techniques based on immediate update, Shadow paging, Database backup and recovery from catastrophic failures.	10

Course Outcomes (COs):

CO1	Design conceptual entity relationship diagrams for the real world applications.
CO2	Use Structured Query Language (SQL) for database manipulation and also demonstrate the basic of query evaluation.
CO3	Develop an optimized database using design guidelines and normalization technique.
CO4	Design and build simple database systems and relate the concept of transaction processing.
CO5	Develop application to interact with databases.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	-	1	-	-	-	-	1	1	2	3	2	-
CO2	3	3	2	-	2	2	-	-	1	-	-	1	2	2	-
CO3	2	3	1	-	1	--	-	-	-	-	2	2	2	3	-
CO4	3	2	2	-	1	-	-	-	-	-	-	1	1	2	-
CO5	3	3	2	-	2	1	-	-	-	-	-	1	1	2	-

Question paper pattern:

The question paper will have ten questions.

There will be 2 questions from each module.

Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

1. Fundamentals of Database Systems, RamezElmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill

Reference Books:

1. SilberschatzKorth and Sudharshan, Database System Concepts, 6th Edition, McGrawHill, 2013.

Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementation and Management, Cengage Learning 2012.

COMPUTER NETWORKS

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – V

Subject Code	**18CS52**	**CIE Marks**	**50**
Number Lecture Hour/Week	**03**	**SEE Marks**	**50**
Number of Lecture Hours	**48**	**Exam Hours**	**03**
CREDITS-04			

Course Objectives: Demonstration of application layer protocols. Discuss transport layer services and understand UDP and TCP protocols. Explain routers, IP and Routing Algorithms in network layer. Demonstration of application layer protocols • Discuss transport layer services and understand UDP and TCP	
Modules	**Hours**
Module I	
Introduction - Hardware and software, Data communication, Networking, Protocols and Protocol architecture, standards. Data transmission concepts. Analog and digital transmission. Transmission impairments. Layered Architecture of Computer Networks, OSI and TCP/IP architectures	10
Module II	
Physical Layer and Data link Layer – 1 - Guided transmission media and wireless transmission, Multiplexing, Spread spectrum. Switching: Introduction, Circuit- Switched networks, packet switching. Data link layer: Introduction, Link-layer addressing. error detection and Correction: Introduction, Block Coding, Cyclic codes, checksum, forward error correction	10
Module III	
Data **Link Layer-2:** DLC services, Data Link layer protocols, HDLC, PPP, Random access, Controlled access, Channelization, Ethernet protocol, Standard Ethernet, Fast Ethernet	10
Module IV	
Network and Transport Layer: Network layer services, Packet switching, Network layer performance, IPV4 addresses, Forwarding of IP packets, IP, ICMPv4,Mobile IP. Unicast routing: Introduction, Routing algorithms, Unicast routing protocols	10
Module V	
Transport Layer and Application Layer: IPv6 addressing, IPv6 protocol, transport layer protocols: Introduction, UDP, TCP, Standard Client-server protocols: WWW and HTTP, FTP, Electronic mail, DNS	08

Course Outcomes (COs):

CO1	Analyze the basic principles of Computer Networks and enumerate the functions of OSI and TCP/IP architectures
CO2	Analyze the transmission medias and apply the link layer properties for error and flow control
CO3	Conceptualize the protocols of data link layer
CO4	Evaluate the performance of network and analyze routing algorithms
CO5	Analyze transport layer services, protocols and principles of application layers

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	-	1	1	-	1	-	-	-	-	-	1	1	1	3	-
CO2	2	2	1	-	2	-	-	-	-	-	2	1	1	2	-
CO3	2	1	1	-	1	1	-	-	-	-	1	1	1	3	-
CO4	2	3	1	-	2	-	-	-	-	-	3	1	1	2	-
CO5	1	1	2	-	2	2	-	-	-	-	3	1	1	2	-

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

Data communication & Networks , by Behrouz A. Forouzan, Tata McGraw Hill. 2002 .

Reference Books:

1. Data Communications, Computer networking on OSI, by Fred Halsall, Addison Wesley Publishing Co. 1998.
2. Computer Networking -A Top-Down Approach Featuring the Internet, James F. Kurose and Keith W. Ross , Addison Wesley Publishing Co. 2004
3. Computer Networks: Protocols standards and interfaces, by Uyless Black, Prentice Hall 2002.
4. Computer Networks, by Andrew S. Tanenbaum, PHI. (2010) Data and Computer Communications, by Walliam Stallings, PHI. (2002).

AUTOMATA THEORY AND COMPUTABILITY

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – V

Subject Code	**18CS531**	**CIE Marks**	**50**
Number Lecture Hour/Week	**03**	**SEE Marks**	**50**
Number of Lecture Hours	**40**	**Exam Hours**	**03**
CREDITS-03			

Course Objectives:

This course will enable students to:

Introduce core concepts in Automata and Theory of Computation

Identify different Formal language Classes and their Relationships

Design Grammars and Recognizers for different formal languages

Prove or disprove theorems in automata theory using their properties

Determine the decidability and intractability of Computational problems

Modules	Hours
Module I	
Why study the Theory of Computation, Languages and Strings: Strings, Languages. A Language Hierarchy, Computation, **Finite State Machines (FSM)**: Deterministic FSM, Regular languages, Designing FSM, Nondeterministic FSMs, From FSMs to Operational Systems, Simulators for FSMs, Minimizing FSMs.	8
Module II	
Regular Expressions (RE): what is a RE?, Kleene's theorem, Applications of REs, Manipulating and Simplifying REs. Regular Grammars: Definition, Regular Grammars and Regular languages. Regular Languages (RL) and Non-regular Languages: How many RLs, To show that a language is regular, Closure properties of RLs, to show some languages are not RLs.	8
Module III	
Context-Free Grammars (CFG): Introduction to Rewrite Systems and Grammars, CFGs and languages, designing CFGs, simplifying CFGs, proving that a Grammar is correct, Derivation and Parse trees, Ambiguity, Normal Forms. Pushdown Automata (PDA): Definition of non- deterministic PDA, Deterministic and Non-deterministic PDAs, Nondeterminism and Halting, alternative equivalent definitions of a PDA, alternatives that are not equivalent to PDA.	8
Module IV	
Algorithms and Decision Procedures for CFLs: Decidable questions, Un-decidable questions. **Turing** by TM, design of TM, Techniques for TM construction. Variants of Turing Machines (TM), The model of Linear Bounded automata **Machine**: Turing machine model, Representation, Language acceptability	8
Module V	
Decidability: Definition of an algorithm, decidability, decidable languages, Undecidable languages, halting problem of TM, Post correspondence problem. Complexity: Growth rate of functions, the classes of P and NP, Quantum Computation: quantum computers, Church- Turing thesis. **Applications:** G.1 Defining syntax of programming language, Appendix J: Security	8

Course Outcomes (COs):

CO1	Design a computational model Finite state machine with conversion between different types of FA and minimize the given FA for any regular language.
CO2	Develop regular expressions, Languages and apply it for designing compilers.
CO3	Develop context free grammar & pushdown automata for the given language and conversion between PDA & CFG.
CO4	Analyze CFL and Design a computational model of Turing machine.
CO5	Analyze and understand decidability and undecidability of various problems with their complexity analysis.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	1	3	-	-	-	-	-	-	-	-	-	2	3	-
CO2	1	1	3	-	-	-	-	-	-	-	-	-	2	3	-
CO3	1	1	3	-	-	-	-	-	-	-	-	-	2	3	-
CO4	1	2	3	-	-	-	-	-	-	-	-	-	2	3	-
CO5	1	2	1	-	-	-	-	-	-	-	-	-	2	3	-

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module

Textbooks:

1. Elaine Rich, Automata, Computability and Complexity, 1st Edition, Pearson education,2012/2013.
2. K L P Mishra, N Chandrasekaran , 3rd Edition, Theory of Computer Science, PhI, 2012.

Reference Books:

1. John E Hopcroft, Rajeev Motwani, Jeffery D Ullman, Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Pearson Education, 2013
2. Michael Sipser : Introduction to the Theory of Computation, 3rd edition, Cengage learning,2013
3. John C Martin, Introduction to Languages and The Theory of Computation, 3rd Edition, Tata McGraw –Hill Publishing Company Limited, 2013
4. Peter Linz, “An Introduction to Formal Languages and Automata”, 3rd Edition, Narosa Publishers, 1998
5. Basavaraj S. Anami, Karibasappa K G, Formal Languages and Automata theory, Wiley India, 2012 C K Nagpal, Formal Languages and Automata Theory, Oxford University press, 2012.

CLOUD COMPUTING

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2020-2021)

SEMESTER – V

Subject Code	18CS532	**CIE Marks**	**50**
Number Lecture Hour/Week	**03**	**SEE Marks**	**50**
Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS-03

Course Objectives:

- Explain the fundamentals of cloud computing
- Illustrate the cloud application programming and aneka platform
- Contrast different cloud platforms used in industry

Modules	**Hours**
Module I	
Introduction ,Cloud Computing at a Glance, The Vision of Cloud Computing, Defining a Cloud, A Closer Look, Cloud Computing Reference Model, Characteristics and Benefits, Challenges Ahead, Historical Developments, Distributed Systems, Virtualization, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing, Building Cloud Computing Environments, Application Development, Infrastructure and System Development, Computing Platforms and Technologies, Amazon Web Services (AWS), Google App Engine, Microsoft Azure, Hadoop, Force.com and Salesforce.com, Manjrasoft Aneka Virtualization, Introduction, Characteristics of Virtualized, Environments Taxonomy of Virtualization Techniques, Execution Virtualization, Other Types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples Xen: Paravirtualization, VMware: Full Virtualization, Microsoft Hyper-V	8
Module II	
Cloud Computing Architecture, Introduction, Cloud Reference Model, Architecture, Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds, Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges, Cloud Definition, Cloud Interoperability and Standards Scalability and Fault Tolerance Security, Trust, and Privacy Organizational Aspects Aneka: Cloud Application Platform, Framework Overview, Anatomy of the Aneka Container, From the Ground Up: Platform Abstraction Layer, Fabric Services, foundation Services, Application Services, Building Aneka Clouds, Infrastructure Organization, Logical Organization, Private Cloud Deployment Mode, Public Cloud Deployment Mode, Hybrid Cloud Deployment Mode, Cloud Programming and Management, Aneka SDK, Management Tools	8
Module III	
Concurrent Computing: Thread Programming, Introducing Parallelism for Single Machine Computation, Programming Applications with Threads, What is a Thread?, Thread APIs, Techniques for Parallel Computation with Threads, Multithreading with Aneka, Introducing the Thread Programming Model, Aneka Thread vs. Common Threads, Programming Applications with Aneka Threads, Aneka Threads Application Model, Domain Decomposition: Matrix Multiplication, Functional Decomposition: Sine, Cosine, and Tangent. High-Throughput Computing: Task Programming, Task Computing, Characterizing a Task, Computing Categories, Frameworks for Task Computing, Task-based Application Models, Embarrassingly Parallel Applications, Parameter Sweep Applications, MPI Applications, Computing Categories, Frameworks for Task Computing, Task-based Application Models, Embarrassingly Parallel Applications, Parameter Sweep Applications, MPI Applications, Workflow Applications with Task Dependencies, Aneka Task-Based Programming, Task Programming Model, Developing Applications with the Task Model, Developing Parameter Sweep Application, Managing Workflows.	8

Module IV	
Data Intensive Computing: Map-Reduce Programming, what is Data-Intensive Computing? Characterizing Data-Intensive Computations, Challenges Ahead, Historical Perspective, Technologies for Data-Intensive Computing, Storage Systems, Programming Platforms, Aneka MapReduce Programming, Introducing the MapReduce Programming Model, Example Application	8
Module V	
Cloud Platforms in Industry, Amazon Web Services, Compute Services, Storage Services, Communication Services, Additional Services, Google App Engine, Architecture and Core Concepts, Application Life-Cycle, Cost Model, Observations, Microsoft Azure, Azure Core Concepts, SQL Azure, Windows Azure Platform Appliance. Cloud Applications Scientific Applications, Healthcare: ECG Analysis in the Cloud, Biology: Protein Structure Prediction, Biology: Gene Expression Data Analysis for Cancer Diagnosis, Geoscience: Satellite Image Processing, Business and Consumer Applications, CRM and ERP, Productivity, Social Networking, Media Applications, Multiplayer Online Gaming.	8

Course Outcomes (COs):

CO1	Students will be able to define cloud computing, describe its benefits and challenges, and explain how virtualization underpins cloud infrastructure
CO2	Students will be able to identify and compare various cloud service models and deployment strategies, and assess their suitability for different use cases.
CO3	Students will develop and implement parallel and task-based applications, leveraging appropriate programming models for cloud platforms.
CO4	Students will be able to design and execute data-intensive computations using the MapReduce framework and identify suitable technologies for specific data-driven tasks .
CO5	Students will evaluate and compare cloud services, assess their benefits in various domains, and gain insights into deploying cloud applications for industry-specific needs.

CO-PO-PSO mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	1	2	1	1	1	-	1	-	2	3	3	2
CO2	3	3	2	1	2	1	1	-	-	1	1	2	2	3	3
CO3	3	3	2	2	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	2	3	3	-	1	-	1	2	2	3	3	3	3
CO5	3	2	3	2	3	2	2	1	2	3	3	3	3	3	2

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module

Textbooks:

1. RajkumarBuyya, Christian Vecchiola, and ThamaraiSelvi Mastering Cloud. Computing McGraw Hill Education

Reference Book:

1.Dan C. Marinescu, Cloud Computing Theory and Practice, Morgan Kaufmann, Elsevier 2013.

UNIX SYSTEM PROGRAMMING

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2020-2021)

SEMESTER – V

Subject Code	**18CS533**	**CIE Marks**	**50**
Number Lecture Hour/Week	**03**	**SEE Marks**	**50**
Number of Lecture Hours	**40**	**Exam Hours**	**03**
CREDITS-03			

Course Objectives: This course will enable students to:

Interpret the features of UNIX and basic commands.

Demonstrate different UNIX files and permissions

Implement shell programs.

Explain UNIX process, IPC and signals.

Modules	Hours
Module I	
Introduction: Unix Components/Architecture. Features of Unix. The UNIX Environment and UNIX Structure, Posix and Single Unix specification. General features of Unix commands/ command structure. Command arguments and options. Basic Unix commands such as echo, printf, ls, who, date,passwd, cal, Combining commands. Meaning of Internal and external commands. The type command: knowing the type of a command and locating it. The root login. Becoming the super user: su command. **Unix files:** Naming files. Basic file types/categories. Organization of files. Hidden files. Standard directories. Parent child relationship. The home directory and the HOME variable. Reaching required files- the PATH variable, manipulating the PATH, Relative and absolute pathnames. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations to represent present and parent directories and their usage in relative path names. File related commands – cat, mv, rm, cp, wc and od commands.	8
Module II	
File attributes and permissions: The ls command with options. Changing file permissions: the relative and absolute permissions changing methods. Recursively changing file permissions. Directory permissions. The shells interpretive cycle: Wild cards. Removing the special meanings of wild cards. Three standard files and redirection. Connecting commands: Pipe. Basic and Extended regular expressions. The grep, egrep. Typical examples involving different regular expressions. Shell programming: Ordinary and environment variables. The .profile. Read and readonly commands. Command line arguments. exit and exit status of a command. Logical operators for conditional execution. The test command and its shortcut. The if, while, for and case control statements. The set and shift commands and handling positional parameters. The here (<<) document and trap command. Simple shell program examples.	8
Module III	
UNIX File APIs: General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs. **UNIX Processes and Process Control:** **The Environment of a UNIX Process:** Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, getrlimit, setrlimit Functions, UNIX Kernel Support for Processes. **Process Control:** Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3, wait4 Functions, Race Conditions, exec Function	8
Module-IV	
Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times, I/O Redirection. **Overview of IPC Methods**, Pipes, popen, pclose Functions, Coprocesses, FIFOs, System V IPC, Message Queues, Semaphores. **Shared Memory**, Client-Server Properties, Stream Pipes, Passing File Descriptors, An Open Server-Version 1, Client-Server Connection Functions.	8
Module-V	

Signals and Daemon Processes: Signals: The UNIX Kernel Support for Signals, signal, Signal Mask, sigaction, The SIGCHLD Signal and the waitpid Function, The sigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX.lb Timers. Daemon Processes: Introduction, Daemon Characteristics, Coding Rules, Error Logging, Client-Server Model.	8

Course Outcomes (COs):

CO1	Identify UNIX architecture, commands, and file system for navigation and file management.
CO2	Identify and demonstrate UNIX permissions, shell commands, and scripting for task automation
CO3	Apply UNIX file and process control APIs for system-level programming.
CO4	Implement user management and IPC for client-server applications.
CO5	Utilize signal handling and daemon processes for reliable client-server programs.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	-	-	2	-	-	-	-	-	-	-	2	2	3
CO2	2	2	-	-	2	-	-	-	-	-	-	-	1	2	2
CO3	2	-	2	-	2	-	-	-	-	-	-	-	1	2	2

CO4	-	-	2	-	3	-	-	-	3	-	-	-	1	2	-
CO5	-	-	2	-	3	3	-	-	-	-	-	-	1	2	-

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module

Textbooks:

1. RajkumarBuyya, Christian Vecchiola, and ThamaraiSelvi Mastering Cloud. Computing McGraw Hill Education

Reference Book:

1.Dan C. Marinescu, Cloud Computing Theory and Practice, Morgan Kaufmann, Elsevier 2013.

SOCIAL NETWORK ANALYSIS

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – V

Subject Code	**18CS534**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**40**	**Exam Hours**	03

CREDITS – 03

Course objectives: This course will enable students to

Discuss essential knowledge of network analysis applicable to real world data, with examples from today's most popular social networks.

Modules	**Hours**
Module I	
Introduction to social network analysis and Descriptive network analysis: Introduction to new science of networks. Networks examples. Graph theory basics. Statistical network properties. Degree distribution, clustering coefficient. Frequent patterns. Network motifs. Cliques and k-cores	8
Module II	
Network structure, Node centralities and ranking on network: Nodes and edges, network diameter and average path length. Node centrality metrics: degree, closeness and betweenness centrality. Eigenvector centrality and PageRank. Algorithm HITS.	8
Module III	
Network communities and Affiliation networks: Networks communities. Graph partitioning and cut metrics. Edge betweenness. Modularity clustering. Affiliation network and bipartite graphs.1-mode projections.Recommendation systems.	8
Module IV	
Information and influence propagation on networks and Network visualization: Social Diffusion. Basic cascade model. Influence maximization.Most influential nodes in network. Network visualization and graph layouts.Graph sampling. Low -dimensional projections	8
Module V	
Social media mining and SNA in real world: FB/VK and Twitter analysis: Natural language processing and sentiment mining. Properties of large social networks: friends, connections, likes, re-tweets.	8

Course Outcomes (COs):

CO1	Analyze social media networks and their characteristics.
CO2	Apply centrality metrics and ranking algorithms like PageRank and HITS to assess node significance in network analysis.
CO3	Identify key properties of large-scale social networks.
CO4	Evaluate user interactions such as likes, retweets, and connections.
CO5	Use real-world datasets for social network analysis.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	2	-	-	1	-	-	-	-	-	-	-	1	1	1
CO2	1	2	2	-	2	-	-	-	-	-	-	-	1	1	2
CO3	-	1	-	1	-	-	1	-	-	-	-	-	1	1	1
CO4	-	-	1	-	-	1	-	-	-	1	-	-	1	1	1
CO5	-	-	-	1	1	-	-	-	-	1	-	-	1	1	1

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books and Reference Books:

David Easley and John Kleinberg. "Networks, Crowds, and Markets: Reasoning About a Highly Connected World." Cambridge University Press 2010.

Eric Kolaczyk, Gabor Csardi. “Statistical Analysis of Network Data with R (UseR!)”.Springer, 2014.

Stanley Wasserman and Katherine Faust. "Social Network Analysis. Methods and Applications." Cambridge University Press, 1994.

DISCRETE MATHEMATICAL STRUCTURES AND GRAPH THEORY

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – V

Subject Code	**18CS541**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**40**	**Exam Hours**	03

CREDITS – 03

Course Objectives: This course will enable students to

- Provide theoretical foundations of computer science to perceive other courses in the Programme.
- Illustrate applications of discrete structures: logic, relations, functions, set theory and counting.
- Describe different mathematical proof techniques
- Illustrate the importance of graph theory in computer science

Modules	**Hours**
Module 1	
Fundamentals of Logic: Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic, Logical Implication – Rules of Inference. Fundamentals of Logic contd.: The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems.	8
Module II	
Properties of the Integers: The Well Ordering Principle – Mathematical Induction, Fundamental Principles of Counting: The Rules of Sum and Product, Permutations, **Combinations –** The Binomial Theorem, Combinations with Repetition.	8
Module III	
Relations and Functions: Cartesian Products and Relations, Functions – Plain and One- to-One, Onto Functions. The Pigeon-hole Principle, Function Composition and Inverse Functions. **Relations:** Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions.	8
Module IV	
The Principle of Inclusion and Exclusion: The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials.**Recurrence Relations:** First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients.	8
Module V	
Introduction to Graph Theory: Definitions and Examples, Sub graphs, Complements, and Graph Isomorphism, **Trees**: Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes **The Principle of Inclusion and Exclusion**: The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials.**Recurrence Relations:** First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients.	8

Course Outcomes (COs):

CO1	Analyze and use propositional and predicate logic in knowledge representation and truth verification.
CO2	Demonstrate the ability to solve problems using Counting Techniques & combinatory in the context of discrete proability.
CO3	Solve problems using recurrence relations and generating functions.
CO4	Interpret problems involving relations and principles of counting
CO5	Demonstrate the fundamental concepts in graph theory and trees

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	-	-	-	-	-	-	-	-	-	2	1	-
CO2	3	3	3	-	-	-	-	-	-	-	-	-	2	1	-
CO3	3	3	3	-	-	-	-	-	-	-	-	-	2	1	-
CO4	3	3	3	-	-	-	-	-	-	-	-	-	2	1	-
CO5	3	3	2	-	-	-	-	-	-	-	-	-	2	1	-

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks:

1. Ralph P. Grimaldi: Discreteand Combinatorial Mathematics, 5thEdition,Pearson Education.2004.

Reference Books:

1. Basavaraj S Anami and Venakanna S Madalli: Discrete Mathematics – A Concept based approach, Universities Press, 2016

2.Kenneth H. Rosen: Discrete Mathematics and its Applications, 6th Edition, McGraw Hill, 2007.

3.Jayant Ganguly: A Treatise on Discrete Mathematical Structures, Sanguine-Pearson, 2010.

4.D.S. Malik and M.K. Sen: Discrete Mathematical Structures: Theory and Applications, Thomson, 2004.

5.Thomas Koshy: Discrete Mathematics with Applications, Elsevier, 2005, Reprint 2008.

MOBILE APPLICATION DEVELOPMENT

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2020-2021)

SEMESTER – V

Subject Code	**18CS543**	**CIE Marks**	**50**

Number Lecture Hour/Week	**03**	**SEE Marks**	**50**
Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS-03

Course Objectives:

This course will enable students to:

- Learn to setup Android application development environment
- Illustrate user interfaces for interacting with apps and triggering actions
- Interpret tasks used in handling multiple activities
- Identify options to save persistent application dat
- Appraise the role of security and performance in Android applications• Create, test and debug Android application by setting up Android development environment

Modules	**Hours**
Module I	
Get started, Build your first app, Activities, Testing, debugging and using support libraries	8
Module II	
User Interaction, Delightful user experience, Testing your UI	8
Module III	
Background Tasks, Triggering, scheduling and optimizing background tasks	8
Module IV	
All about data, Preferences and Settings, Storing data using SQLite, Sharing data with content providers, Loading data using Loaders	8
Module V	
Permissions, Performance and Security, Firebase and AdMob, Publish	8

Course Outcomes (COs):

CO1	Build a foundational Android app, leveraging activities and support libraries for efficient development and debugging
CO2	Design and implement user interfaces that enhance interaction and optimize the overall user experience.
CO3	Develop and optimize background tasks for efficient resource management and performance.
CO4	Manage data in Android apps using preferences, SQLite, content providers, and Loaders for efficient data handling.
CO5	Manage app permissions, optimize performance, integrate Firebase for backend operations, and implement AdMob for monetization and app publishing.

CO-PO-PSO mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	-	2	1	1	1	2	2	1	1	3	2	1
CO2	2	3	2	-	2	2	1	1	2	2	-	1	3	2	1
CO3	2	2	2	-	2	1	2	-	2	-	1	1	2	2	-
CO4	2	2	2	-	3	1	1	1	1	-	-	1	2	2	-
CO5	2	2	2	-	2	2	1	3	1	2	-	1	2	2	3

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks:

1.. Google Developer Training, "Android Developer Fundamentals Course – Concept Reference", Google Developer Training Team, 2017. https://www.gitbook.com/book/googledeveloper-training/android-developer-fundamentals-course-concepts/details (Download pdf file from the above link)

Reference Books:

1. Erik Hellman, "Android Programming – Pushing the Limits", 1st Edition, Wiley India Pvt Ltd, 2014.
2. Dawn Griffiths and David Griffiths, "Head First Android Development", 1st Edition, O"Reilly SPD Publishers, 2015.
3. J F DiMarzio, "Beginning Android Programming with Android Studio", 4th Edition, Wiley India Pvt Ltd, 2016. ISBN-13: 978-8126565580
4. Anubhav Pradhan, Anil V Deshpande, " Composing Mobile Apps" using Android, Wiley 2014, ISBN: 978-81-265-4660-2

MICROCONTROLLER AND EMBEDDED SYSTEM
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2020-2021)
SEMESTER – V

Subject Code	**18CS542**	**CIE Marks**	50

Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**40**	**Exam Hours**	03
CREDITS – 03			
Course Objectives: This course will enable students to			
• To provide students with a foundational understanding of embedded systems, including hardware, software, and programming, enabling them to design and develop embedded solutions for various applications.			

Modules	**Hours**
Module 1	
Microprocessors versus Microcontrollers, ARM Embedded Systems: The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software. ARM Processor Fundamentals: Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and the Vector Table , Core Extensions	8
Module II	
Introduction to the ARM Instruction Set : Data Processing Instructions , Programme Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, Loading Constants **ARM programming using Assembly language:**Writing Assembly code, Profiling and cycle counting, instruction scheduling, Register Allocation, Conditional Execution, Looping Constructs	8
Module III	
Embedded System Components: Embedded Vs General computing system, History of embedded systems, Classification of Embedded systems, Major applications areas of embedded systems, purpose of embedded systems Core of an Embedded System including all types of processor/controller, Memory, Sensors, Actuators, LED, 7 segment LED display, stepper motor, Keyboard, Push button switch, Communication Interface (onboard and external types), Embedded firmware, Other system components.	8
Module IV	
Embedded System Design Concepts: Characteristics and Quality Attributes of Embedded Systems, Operational quality attributes ,non-operational quality attributes, Embedded Systems-Application and Domain specific, Hardware Software Co-Design and Program Modelling, embedded firmware design and development	8
Module V	
RTOS and IDE for Embedded System Design: Operating System basics, Types of operating systems, Task, process and threads (Only POSIX Threads with an example program), Thread preemption, Multiprocessing and Multitasking, Task Communication (without any program), Task synchronization issues – Racing and Deadlock, Concept of Binary and counting semaphores (Mutex example without any program), How to choose an RTOS, Integration and testing of Embedded hardware and firmware, Embedded system Development Environment – Block diagram (excluding Keil), Disassembler/decompiler, simulator, emulator and debugging techniques, target hardware debugging, boundary scan.	8

Course Outcomes (COs):

CO1	Describe the architectural features and instructions of ARM microcontroller
CO2	Understand the ARM processor assemble level programming
CO3	Interpret the basic hardware components and their selection method based on the characteristics and attributes of embedded system

CO4	Develop the hardware /software co-design and firmware design approaches.
CO5	Demonsrate the need of real time operating system for embedded applications

CO-PO-PSO mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3											3		
CO2	3	2	1										3	1	
CO3	3	3	1		1								3	1	
CO4	3	2											3		
CO5	3	2	1		1								3	1	

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks:

1. Andrew N Sloss, Dominic Symes and Chris Wright, ARM system developers guide, Elsevier, Morgan Kaufman publishers, 2008.

2. Shibu K V, "Introduction to Embedded Systems", Tata McGraw Hill Education, Private Limited, 2 nd Edition

Reference Books:

1.Raghunandan..G.H, Microcontroller (ARM) and Embedded System, Cengage learning Publication,2019

2. Steve Furber, ARM System-on-Chip Architecture, Second Edition, Pearson, 2015.

3. Raj Kamal, Embedded System, Tata McGraw-Hill Publishers, 2nd Edition, 2008.

Green Technology
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2020-2021)
SEMESTER – V

Subject Code	**18CS544**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**40**	**Exam Hours**	03

CREDITS – 03

Course Objectives: This course will enable students to

- Comprehensive understanding of grid computing architecture, standards, and tools.
- Able to apply concepts such as service-oriented architecture, OGSA, and grid services to real-world scenarios.

Modules	**Hours**
Module I	
Introduction, Grid Computing Organizations and Their Roles: Early Grid Activities, Current Grid Activities, An Overview of Grid Business Areas, Grid Applications, Grid Infrastructure. Organizations Developing Grid Standards and Best Practice Guidelines, Organizations Developing Grid Computing Toolkits and the Framework, Organizations Building and Using Grid-Based Solutions to Solve Computing, Data and Network Requirements, Commercial Organizations Building and Using Grid-Based Solutions.	8
Module II	
The Grid Computing Anatomy, Road Map: The Grid Problem. Anatomy Computing, Business on Demand and Infrastructure Virtualization, Service-Oriented Architecture and Grid, Semantic Grids**.**	8
Module III	
Architectures : Service-Oriented Architecture, Web Services Architecture, XML, Related Technologies and Their Relevance to Web Services, XML Messages and Enveloping, Service Message Description Mechanisms. Relationship between Web Service and Grid Service, Web Service Interoperability and the Role of the WS-I Organization, OGSA Architecture and Goals, Commercial Data Center (CDC), National Fusion Collaboratory (NFS), Online Media and Entrainment.	8
Module IV	
The OGSA Platform Components, OGSI: Native Platform Services and Transport Mechanisms, OGSA Hosting Environment, Core Networking Services Transport and Security, OGSA Infrastructure, OGSA Basic Services. Grid Services, A High-Level Introduction to OGSI (Open Grid Services Infrastructure). Technical Details of OGSI Specification, Introduction to Service Data Concepts, Grid Service: Naming and Change Management Recommendations.	8
Module V	
OGSA Basic Services and Toolkit: Common Management Model (CMM), Service Domains, Policy Architecture, Security Architecture, Metering and Accounting. Common Distributed Logging, Distributed Data Access and Replication. GLOBUS GT3 Toolkit Architecture	8

Course Outcomes (COs):

CO1	Understand the fundamental concepts of grid computing, including its history, key organizations, and applications in various business and scientific areas.
CO2	Analyze the architecture of grid computing, including the concepts of service-oriented architecture, grid problems, infrastructure virtualization, and semantic grids.
CO3	Explore the relationship between Web Services and Grid Services, understanding XML-related technologies and the roles they play in grid computing architectures.
CO4	Examine the components of the OGSA (Open Grid Services Architecture) platform, including OGSI (Open Grid Services Infrastructure) and core networking services for secure and efficient grid operations.
CO5	Utilize and implement grid computing tools and services, focusing on management, security, and accounting within the OGSA framework, including practical use of the GLOBUS GT3 toolkit.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	-	-	-	-	-	-	-	-	-	3	3	2
CO2	3	3	2	1	-	-	-	-	-	-	-	-	3	3	3
CO3	2	3	2	1	3	-	-	-	-	2	-	-	3	2	3
CO4	3	3	2	3	2	-	-	-	-	2	-	-	3	3	3
CO5	3	2	3	2	3	-	2	2	1	3	-	-	3	3	2

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module

Text Books and Reference Books:

1. Joshy Joseph, Craig Fellenstein: Grid Computing, IBM Press, 2007.
2. Prabhu: Grid and Cluster Computing, Prentice-Hall of India, 2007

WEB PROGRAMMING LABORATORY

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – V

Subject Code	**18CSL55**	**CIE Marks**	50
Number of Lecture Hours/Week	**02**	**SEE Marks**	50
Total Number of Lecture Hours	**30**	**Exam Hours**	03

CREDITS – 01

1. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient.

2. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.

3. Write a JavaScript code that displays text “TEXT-GROWING” with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays “TEXT-SHRINKING” in BLUE color. Then the font size decreases to 5pt.

4. Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems:

Parameter: A string

Output: The position in the string of the left-most vowel

Parameter: A number

Output: The number with its digits in the reverse order

5.Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, and Name of the College, Branch, Year of Joining, and email id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.

6.Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.

7.Write a PHP program to display a digital clock which displays the current time of the server.

8.Write the PHP programs to do the following:

a.Implement simple calculator operations.

b.Find the transpose of a matrix.

c.Multiplication of two matrices.

d.Addition of two matrices.

9.Write a PHP program named states.py that declares a variable states with value "Mississippi Alabama Texas Massachusetts Kansas". write a PHP program that does the following:

a. Search for a word in variable states that ends in xas. Store this word in element 0 of a list named states

b.Search for a word in states that begins with k and ends in s. Perform a caseinsensitive comparison. [Note: Passing re.Ias a second parameter to method compile performs a case- insensitive comparison.] Store this word in element1 of statesList.

c. Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list.

d. Search for a word in states that ends in a. Store this word in element 3 of the list.

10.Write a PHP program to sort the student records which are stored in the database using selection sort.

Course Outcomes (COs):

CO1	Illustrate fundamental principles of HTML, JavaScript, PHP, and Perl through a series of practical experiments.
CO2	Develop a functional web application utilizing Apache server, PHP, XAMPP, and Perl.
CO3	Debug and troubleshoot software issues effectively
CO4	Ability to evaluate various aspects of web pages and interpret the results
CO5	Prepare a well-organized laboratory report detailing experimental procedures, results

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	-	-	-	-	-	-	-	-	-	3	3	3
CO2	3	2	3	-	-	-	-	-	-	-	-	-	3	3	3
CO3	2	3	2	-	-	-	-	-	-	-	-	-	2	3	2
CO4	2	2	2	-	-	-	-	-	-	-	-	-	2	3	2
CO5	2	1	1	-	-	-	-	-	-	-	-	-	2	2	1

Conduct of Practical Examination:

- Experiment distribution

- For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
- For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

- Marks Distribution (Courseed to change in accoradance with university regulations)
- For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+30+5 = 50 Marks
- For laboratories having PART A and PART B

Part A – Procedure + Execution + Viva = 7 + 20 + 3 = 30 Marks

iv. Part B – Procedure + Execution + Viva = 6 + 12 + 2= 20 Marks

DATABASE MANAGEMENT LAB
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2020-2021)
SEMESTER – V

Subject Code	**18CSL56**	**CIE Marks**	50
Number of Lecture Hours/Week	**02**	**SEE Marks**	50
Total Number of Lecture Hours	**30**	**Exam Hours**	03

CREDITS – 01

Course Objectives:

Able to understand the Knowledge about basic SQL fundamentals and table operations.
Able to understand the implementation of Various SQL commands
Understand the working of Commit and Rollback
Understand the implementation of nested queries and transactions.

1. Write the queries for Data Definition and Data Manipulation language.
2. Write SQL queries using Logical operators (=,,etc.).

3. Write SQL queries using SQL operators (Between…. AND, IN(List), Like, IS NULL and also with negating expressions).

4. Write SQL query using character, number, date and group functions.
5. Write SQL queries for Relational Algebra (UNION, INTERSECT, and MINUS, etc.).

6. Write SQL queries for extracting data from more than one table (Equi-Join, Non-EquiJoin , Outer Join)

7. Write SQL queries for sub queries, nested queries.
8. Write programs by the use of PL/SQL.
9. Concepts for ROLL BACK, COMMIT & CHECK POINTS
10. Create VIEWS, CURSORS, and TRIGGRS & write ASSERTIONS.
11. Create FORMS and REPORTS.

12. Creation, altering and droping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.

Course Outcomes (COs):

CO1	Demonstrate database concepts through series of queries.
CO2	Develop a program using MySql.
CO3	Effectively debug and troubleshoot issues in DBMS programs, ensuring stable and performant database operations.
CO4	Examine data and query outputs.
CO5	Prepare a well organized laboratory report.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	-	-	-	-	-	-	-	-	1	1	2	2	2
CO2	2	3	3	-	2	-	-	-	1	-	2	2	2	2	2
CO3	1	3	-	-	1	-	-	-	-	-	-	-	1	1	3
CO4	1	2	2	2	-	2	-	-	-	-	-	-	1	-	3
CO5	1	-	-	-	-	-	-	-	-	2	-	-	1	-	3

Conduct of Practical Examination:

- Experiment distribution

- For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
- For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

- Marks Distribution (Courseed to change in accoradance with university regulations)
- For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+30+5 = 50 Marks
- For laboratories having PART A and PART B

Part A – Procedure + Execution + Viva = 7 + 20 + 3 = 30 Marks

vi. Part B – Procedure + Execution + Viva = 6 + 12 + 2= 20 Marks

Computer Networks Lab

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – V

Subject Code	**18CSL57**	**CIE Marks**	50
Number of Lecture Hours/Week	**02**	**SEE Marks**	50
Total Number of Lecture Hours	**30**	**Exam Hours**	03

CREDITS – 01

PART-A

Objectives:

To understand the functionalities of various layers of OSI model
To understand the operating system functionalities

Implement the following using C/C++ or equivalent with LINUX/ Windows environment:

1. Using TCP/IP Socket programming, implement a program to transfer the contents of a requested file from server to the client using TCP/IP Sockets.
2. Implement the data link layer farming methods such as character, character stuffing and bit stuffing.
3. Implement on a data set of characters the three CRC polynomials - CRC 12, CRC 16 and CRC CCIP.
4. Write a program for frame sorting technique used in buffers.
5. Write a program for Hamming Code generation for error detection and correction.
6. Take an example subnet graph with weights indicating delay between nodes. Now obtain Routing table at each node using distance vector routing algorithm.
7. Using Leaky Bucket Algorithm, design a program to achieve Traffic management at Flow level by implementing Closed Loop Control technique.
8. Using RSA algorithm encrypt a text data and decrypt the same.

Part B

Simulation Programs using any network simulator or any other equivalent software.
Note: (i) Analyze the network behavior by collecting the statistics on network performance and draw the conclusion.

1. Simulate a 3node point to point network with duplex links between them. Set the Queue size and vary the bandwidth and find the number of packets dropped.
2. Simulate a four-node point-to-point network, and connect the links as follows: n0->n2, n1- >n2 and n2-n3. Apply TCP agent changing the parameters and determine the number of packets sent/received by TCP/UDP.
3. Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.
4. Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.
5. Implement and study the performance of GSM on NS2/NS3 (Using MAC layer) or equivalent environment.
6. Implement and study the performance of CDMA on NS2/NS3 (Using stack called Call net) or equivalent environment.

Course Outcomes (COs):

CO1	Demonstrate theoretical concepts of different layers of OSI model, through series of experiments
CO2	Develop a program using C and simulate and analyze network systems using NS2 tool
CO3	Debug and troubleshoot issues effectively.
CO4	Analyze the data and interpret the results.
CO5	Prepare a well-organized laboratory report.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	-	1	-	-	-
CO2	1	2	2	-	3	-	-	-
CO3	-	3	-	-	1	-	-	-
CO4	1	2	2	2	-	2	-	-
CO5	1	-	-	-	-	-	-	-

Conduct of Practical Examination:

- Experiment distribution
- For laboratories having only one part: Students are allowed to pick one experiment from the lot with
- For laboratories having PART A and PART B: Students are allowed to pick one experiment from PA
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the chang
- Marks Distribution (Courseed to change in accoradance with university regulations)
- For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+30+5 = 50 Marks
- For laboratories having PART A and PART B

Part A – Procedure + Execution + Viva = 7 + 20 + 3 = 30 Marks

viii. Part B – Procedure + Execution + Viva = 6 + 12 + 2= 20 Marks

PROJECT-V
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2020-2021)
SEMESTER – V

Subject Code	**18PRJ58**	**CIE Marks**	50
Number of Lecture Hours/Week	**02**	**SEE Marks**	50
Total Number of Lecture Hours	**48**	**Exam Hours**	03

CREDITS – 01

Course objectives: This course will enable students

- Identify real-world problems across programming, databases, and networking domains and understand their business and technical implications.
- Apply systematic methodologies to design, implement, and optimize solutions.
- Resolve technical challenges through debugging, research, and collaboration.
- Take responsibility for specific roles in a team and collaborate effectively to achieve project goals.
- Present project progress and findings clearly and confidently to both technical and non-technical audiences.
- Document the entire project in a structured, professional laboratory report.

Course Outcomes (COs):

CO1	Identify the topic from various domains (example programming databases, networking) to real world problems.
CO2	Develop methodology for the problem.

CO3	Resolve issues that arise during the project .
CO4	Learn to assign and accept roles and responsibilities within a team and write a good technical reports.
CO5	Exhibit skills in presenting their project findings & progress orally

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	2	2	-	1	1	1	2	2	1	2	2	3	2
CO2	2	2	2	2	2	1	2	1	2	2	1	1	2	2	2
CO3	2	2	2	2	3	1	1	1	2	2	1	1	2	3	2
C04	-	-	-	-	-	1	-	2	2	3	1	1	1	1	1
CO5	-	-	-	-	1	1	-	2	2	3	1	1	1	1	1

SYSTEM SOFTWARE AND COMPILER DESIGN

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Subject Code	**18CS61**	**CIE Marks**	**50**
Number Lecture Hour/Week	**04**	**SEE Marks**	**50**
Number of Lecture Hours	**48**	**Exam Hours**	**03**

CREDITS-04

Course Objectives: This course will enable students to:

- Define System Software.
- Familiarize with source file, object file and executable file structures and libraries
- Describe the front-end and back-end phases of compiler and their importance to students

Modules	**Hou**
Module I	

Introduction to System Software, Machine Architecture of SIC and SIC/XE. **Assemblers:** Basic assembler functions, machine dependent assembler features, machine independent assembler features, assembler design options. Basic Loader Function	
Module II	
Introduction: Language Processors, The structure of a compiler, The evaluation of programming languages, The science of building compiler, Applications of compiler technology. **Lexical Analysis:** The role of lexical analyzer, Input buffering, Specifications of token, recognition of tokens.	
Module III	
Syntax Analysis: Introduction, Context Free Grammars, Writing a grammar, Top Down Parsers, Bottom-Up Parsers	
Module IV	
Lex and Yacc –The Simplest Lex Program, Grammars, Parser-Lexer Communication, A YACC Parser, The Rules Section, Running LEX and YACC, LEX and Hand- Written Lexers, Using LEX - Regular Expression, Examples of Regular Expressions, A Word Counting Program,Using YACC – Grammars, Recursive Rules, Shift/Reduce Parsing, What YACC Cannot Parse, A YACC Parser - The Definition Section, The Rules Section, The LEXER, Compiling and Running a Simple Parser, Arithmetic Expressions and Ambiguity.	
Module V	
Syntax Directed Translation, Intermediate code generation, Code generation	

Course Outcomes (COs):

CO1	Understand and apply the fundamental concepts of system software, and the core functions of assemblers and loaders.
CO2	Analyze, design, and implement lexical analyzers, evaluate programming languages, and understand the structure and functioning of compilers.
CO3	Develop Top Down and Bottom Up Parser
CO4	Design and implement lexical analyzers and parsers using Lex and YACC
CO5	Apply syntax-directed translation methods, generate intermediate code, and implement code generation techniques

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	1	-	-	1	-	-	-	-	1	-	1	1	3	-
CO2	1	1	1	-	1	-	-	-	-	-	-	1	1	3	-
CO3	2	2	2	-	-	-	-	-	-	-	-	1	1	3	-
C04	1	1	1	-	-	-	-	-	1	1	-	1	1	3	-
CO5	1	1	1	-	1	-	-	-	-	-	-	1	2	3	-

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books

1. System Software by Leland. L. Beck, D Manjula, 3rd edition, 2012
2. Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers-Principles, Techniques and Tools, Pearson, 2nd edition, 2007
3. Doug Brown, John Levine, Tony Mason, lex &yacc, O'Reilly Media, October 2012.

Reference Books:

1. Systems programming – Srimanta Pal , Oxford university press, 2016
2. System programming and Compiler Design, K C Louden, Cengage Learning
3. System software and operating system by D. M. Dhamdhere TMG

Compiler Design, K Muneeswaran, Oxford University Press 2013

OPERATING SYSTEM

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Subject Code	**18CS621**	**CIE Marks**	**50**
Number Lecture Hour/Week	**03**	**SEE Marks**	**50**
Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS-03

Course Objectives:

- Introduce concepts and terminology used in OS
- Explain threading and multithreaded systems
- Illustrate process synchronization and concept of Deadlock
- Introduce Memory and Virtual memory management,
- File system and storage techniques

Modules	**H**
Module I	

Introduction to operating systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments. Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot.**Process Management** Process concept; Process scheduling; Operations on processes; Inter processCommunication	
Module II	
Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling. **Process Synchronization:** Synchronization: The critical section problem; Peterson's solution;Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.	

Module III	
Deadlocks: Deadlocks; System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.**Memory Management:** Memory management strategies: Background; Swapping; Contiguousmemory allocation; Paging; Structure of page table; Segmentation.	08
Module IV	
Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing .File SystemImplementation of File System: File system: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection: Implementing File system: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.	08
Module V	
Secondary Storage Structures, Protection: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability- Based systems. **Case Study: The Linux Operating System:** Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory Management; File systems, Input and output; Inter-process communication	08

Course Outcomes (COs):

CO1	Demonstrate various concepts, features of OS and need for Operating systems.
CO2	Discuss about the threading and multithreaded systems.
CO3	Illustrate processor, memory, storage scheduling criteria's used.
CO4	Illustrate storage and file system commands.
CO5	Explain the different concepts of OS in platform of usage through case studies.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	**3**	**2**	**1**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**3**	**2**
CO2	**2**	**3**	**1**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**3**	**2**
CO3	**3**	**3**	**1**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**3**	**2**
CO4	**2**	**-**	**1**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**3**	**2**
CO5	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**3**	**3**

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books :

1.Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 7thedition, Wiley-India, 2006.

Reference Books:

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition

2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw-Hill, 2013.

3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.

4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition,Pearson

SOFTWARE TESTING

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Subject Code	**18CS622**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**40**	**Exam Hours**	03

CREDITS – 03

Course Objectives: This course will enable students

- Discuss test cases for any given problem
- Compare the different testing techniques
- Illustrate the problem into suitable testing model
- Understand the appropriate technique for the design of flow graph.
- Design and develop appropriate document for the software artefact.

Modules	**Hours**
Module I	
Basics of Software Testing: Basic definitions, Software Quality, Requirements, Behavior and Correctness, Correctness versus Reliability, Testing and Debugging, Test cases, Insights from a Venn diagram, Identifying test cases, Test-generation Strategies, Test Metrics, Error and fault taxonomies, Levels of testing, Testing and Verification, Static Testing.**Problem Statements:** Generalized pseudocode, the triangle problem, the Next Date function, the commission problem, the SATM (Simple Automatic Teller Machine) problem, the currency converter, Saturn windshield wiper.	8
Module II	
Functional Testing: Boundary value analysis, Robustness testing, Worst-case testing, Robust Worst testing for triangle problem, Next date problem and commission problem, Equivalence classes, Equivalence test cases for the triangle problem, Next Date function, and the commission problem, Guidelines and observations, Decision tables, Test cases for the triangle problem, Next Date function, and the commission problem, Guidelines and observations. **Fault Based Testing:** Overview, Assumptions in fault-based testing, Mutation analysis, Fault-based adequacy criteria, Variations on mutation analysis.	8
Module III	
Structural Testing: Overview, Statement testing, Branch testing, Condition testing, Path testing: DD paths, Test coverage metrics, Basis path testing, guidelines and observations, Data –Flow testing: Definition-Use testing, Slice based testing, Guidelines and observations.**Test Execution:** Overview of test execution, from test case specification to test cases, Scaffolding, Generic versus specific scaffolding, Test oracles, Self-checks as oracles, Capture and replay	8

Module IV	
Process Framework: Basic principles: Sensitivity, redundancy, restriction, partition, visibility, Feedback, the quality process, Planning and monitoring, Quality goals, Dependability properties, Analysis Testing, Improving the process, Organizational factors. **Planning and Monitoring the Process:** Quality and process, Test and analysis strategies and plans, Risk planning, monitoring the process, Improving the process, the quality team Documenting **Analysis and Test:** Organizing documents, Test strategy document, Analysis and test plan, Test design specifications documents, Test and analysis reports.	8
Module V	
Integration and Component-Based Software Testing: Overview, Integration testing strategies, Testing components and assemblies. System, Acceptance and Regression Testing: Overview, System testing, Acceptance testing, Usability, Regression testing, Regression test selection techniques, Test case prioritization and selective execution. **Levels of Testing, Integration Testing:** Traditional view of testing levels, Alternative life-cycle models, The SATM system, separating integration and system testing, A closer look at the SATM system,Decomposition-based, call graph-based, Path-based integrations**.**	8

Course Outcomes (COs):

CO1	Discuss test cases for any given problem.
CO2	Compare the different testing techniques.
CO3	Illustrate the problem into suitable testing model.
CO4	Understand the appropriate technique for the design of flow graph.
CO5	Design and develop appropriate document for the software artifact.

CO-PO-PSO mapping:

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Auerbach Publications, 2008. (Listed topics only from Chapters 1, 2, 5, 6, 7, 9, 10, 12, 13)
2. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, Wiley India, 2009. (Listed topics only from Chapters 3, 4, 16, 17, 20,21,22,24)
3. Aditya P Mathur: Foundations of Software Testing, Pearson Education, 2008. (Listed topics only from Section 1.2, 1.3, 1.4 ,1.5, 1.8,1.12,6. 2.1,6. 2.4)

Reference Books:

1. Software testing Principles and Practices – Gopalaswamy Ramesh, SrinivasanDesikan, 2nd Edition, Pearson, 2007
2. Software Testing – Ron Patton, 2nd edition, Pearson Education, 2004.
3. The Craft of Software Testing – Brian Marrick, Pearson Education, 1995
4. AnirbanBasu, Software Quality Assurance, Testing and Metrics, PHI, 2015..

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	-	-	-	-	-	-	-	2	3	2	-	-
CO2	2	3	-	-	-	-	-	-	-	-	-	3	2	-	-
CO3	2	2	-	-	-	-	-	--	-	-	3	3	-	2	-

C04	-	3	-	-	-	-	-	-	-	-	-	3	-	-	3
CO5	3	3	-	-	-	-	-	-	-	-	-	3	-	-	3

Cryptography and Network Security
[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Subject Code	**18CS623**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**40**	**Exam Hours**	03

CREDITS – 03

Course objectives: This course will enable students to

- To learn the concepts of cryptography
- Illustrate key management issues and solutions.
- Familiarize with cryptography and very essential algorithms.
- Introduce cyber law and ethics to be followed

Modules	**Hours**
Module I	
INTRODUCTION TO NETWORK SECURITY: OSI security architecture, security attacks, security services, Security Mechanisms, a model of Network Security. **SYMMETRIC CIPHERS:** Classical Encryption Techniques, Block Ciphers and the Data Encryption Standard, Introduction to Finite Fields, Confidentiality using Symmetric Encryption.	08
Module II	
PUBLIC - KEY ENCRYPTION AND HASH FUNCTIONS: Introduction to Number The Ory, Public-Key Cryptography and RSA, Key Management: Diffie-Hellman Key Exchange, Message Authentication and Hash Functions, secure Hash Algorithm, Digital Signatures and Authentication Protocols.	08
Module III	

NETWORK SECURITY PRACTICE: Authentication Applications: Kerberos, X.509 Authentication Service, Electronic mail Security: Pretty Good Privacy, S/MIME, IP Security: Overview, Architecture, Authentication header, ESP, Key management.	08
Module IV	
SYSTEM SECURITY: Malicious Software: Viruses and Related Threats, Viruses Countermeasures. Distributed Denial of Service Attacks, Firewalls: Firewall Design Principles, Trusted Systems	08
Module V	
IT act aim and objectives, Scope of the act, Major Concepts, Important provisions, Attribution, acknowledgement, and dispatch of electronic records, Secure electronic records and secure digita lsignatures, Regulation of certifying authorities: Appointment of Controller and Other officers, Digital Signature certificates, Duties of Subscribers, Penalties and adjudication, The cyber regulations appellate tribunal, Offences, Network service providers not to be liable in certain cases, Miscellaneous Provisions	08

Course Outcomes (COs):

CO1	Understand network security concepts and cryptography basics, algorithms.
CO2	Analyze the mathematical foundations of public-key encryption, authentication protocols.
CO3	Understand Authentication Applications and apply IPsec protocols.
CO4	Identify and analyze types of malicious software and firewalls.
CO5	Understand cyber security and need of cyberlaws.

CO-PO-PSO mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	1	-	-	-	-	-	-	-	-		1	2	**-**
CO2	2	2	1	1	-	-	-	-	-	--	-	1	1	2	**-**
CO3	2	2	1	1	-	-	-	-	-	-	-	1	1	2	**-**
CO4	2	1	1	1	-	-	-	-	-	-	-	1	1	2	-
C05	2	1	1	-	-	1	-	-	-	-	-	1	1	2	-

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1.Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Auerbach Publications, 2008. (Listed topics only from Chapters 1, 2, 5, 6, 7, 9, 10, 12, 13)

2.Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, Wiley India, 2009. (Listed topics only from Chapters 3, 4, 16, 17, 20,21,22,24)

3.Aditya P Mathur: Foundations of Software Testing, Pearson Education, 2008. (Listed topics only from Section 1.2, 1.3, 1.4 ,1.5, 1.8,1.12,6. 2.1,6. 2.4)

Reference Books:

1.Software testing Principles and Practices – Gopalaswamy Ramesh, SrinivasanDesikan, 2nd Edition, Pearson, 2007

2.Software Testing – Ron Patton, 2nd edition, Pearson Education, 2004.

3.The Craft of Software Testing – Brian Marrick, Pearson Education, 1995

4.AnirbanBasu, Software Quality Assurance, Testing and Metrics, PHI, 2015..

COMPUTER GRAPHICS AND VISUALIZATION

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Course Code	**18CS624**	**CIE Marks**	50
Number of Contact Hours/Week	**03**	**SEE Marks**	50
Total Number of Contact Hours	**40**	**Exam Hours**	03

CREDITS – 03

Course Objectives: This course will enable students to:

- Explain hardware, software and OpenGL Graphics Primitives.
- Illustrate interactive computer graphic using the OpenGL.
- Design and implementation of algorithms for 2D graphics Primitives and attributes.
- Demonstrate Geometric transformations, viewing on both 2D and 3D objects.
- Infer the representation of curves, surfaces, Color and Illumination models

Modules	**Hours**
Module I	
Overview: Computer Graphics and OpenGL: Computer Graphics: Basics of computer graphics, Application of Computer Graphics, Video Display Devices: Random Scan and Raster Scan displays, graphics software. OpenGL: Introduction to OpenGL, coordinate reference frames, specifying two-dimensional world coordinate reference frames in OpenGL, OpenGL point functions, OpenGL line functions, point attributes, line attributes, curve attributes, OpenGL point attribute functions, OpenGL line attribute functions, Line drawing algorithms (DDA,Bresenham’s), circle generation algorithms (Bresenham’s).	8
Module II	

Fill area Primitives, 2D Geometric Transformations and 2D viewing: Fill area Primitives: Polygon fill-areas, OpenGL polygon fill area functions, fill area attributes, general scan line polygon fill algorithm, OpenGL fill-area attribute functions. 2DGeometric Transformations: Basic 2D Geometric Transformations, matrix representations and homogeneous coordinates. Inverse transformations, 2DComposite transformations, other 2D transformations, raster methods for geometric transformations, OpenGL raster transformations, OpenGL geometric transformation's function, 2D viewing: 2D viewing pipeline, OpenGL 2D viewing functions.	8
Module III	
Transformations, Color and Illumination Models: Clipping: clipping window, normalization and viewport transformations, clipping algorithms,2D point clipping, 2D line clipping algorithms: Cohen Sutherland line clipping only -polygon fill area clipping: Sutherland-Hodgeman polygon clipping algorithm only.3DGeometric Transformations: 3D translation, rotation, scaling, composite 3D transformations, other 3D transformations, affine transformations, OpenGL geometric transformations functions. Color Models: Properties of light, color models, RGB and CMY color models. Illumination Models: Light sources, basic illumination models-Ambient light, diffuse reflection, specular and phong model, Corresponding OpenGL functions.	8
Module IV	
3D Viewing and Visible Surface Detection: 3DViewing:3D viewing concepts, 3D viewing pipeline, 3D viewing coordinate parameters, Transformation from world to viewing coordinates, Projection transformation, orthogonal projections, perspective projections, The viewport transformation and 3D screen coordinates. OpenGL 3D viewing functions. Visible Surface Detection Methods: Classification of visible surface Detection algorithms, depth buffer method only and OpenGL visibility detection functions.	8
Module V	
Input& interaction, Curves and Computer Animation: Input and Interaction: Input devices, clients and servers, Display Lists, Display Lists and Modeling, Programming Event Driven Input, Menus Picking, Building Interactive Models, Animating Interactive programs, Design of Interactive programs, Logic operations. Curved surfaces, quadric surfaces, OpenGL Quadric- Surface and Cubic-Surface Functions, Bezier Spline Curves, Bezier surfaces, OpenGL curve functions. Corresponding OpenGL functions.	8

Course Outcomes (COs):

CO1	Apply line and circle drawing algorithms (DDA, Bresenham‟s) using OpenGL to create basic graphics Programs
CO2	Implement and apply 2D geometric transformations and the 2D viewing pipeline in OpenGLapplications
CO3	Analyze and implement 3D geometric transformations, clipping algorithms, and illumination models using OpenGL
CO4	Design and evaluate 3D viewing pipelines and apply visible surface detection techniques using the depth buffer method in OpenGL
CO5	Use interactive computer 3d modelling techniques on curve surfaces.

CO-PO-PSO mapping:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	1	-	-	-	-	-	-	-	-	3	1	2	-
CO2	3	3	2	-	-	-	-	-	-	-	3	3	2	2	1
CO3	3	2	2	-	-	-	-	-	-	-	-	3	2	2	1
C04	3	3	3	-	-	-	-	-	-	-	-	3	2	2	-
CO5	3	3	1	-	-	-	-	-	-	-	-	3	2	2	1

Question Paper Pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module

Textbooks:

1. Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version,3rd / 4th Edition, Pearson Education,2011
2. Edward Angel: Interactive Computer Graphics- A TopDown approach with OpenGL, 5th edition. Pearson Education, 2008

Reference Books:

1. James D Foley, Andries Van Dam, Steven K Feiner, John F Huges Computer graphics with OpenGL: pearson education
2. Xiang, Plastock : Computer Graphics , sham's outline series, 2nd edition, TMG.
3. Kelvin Sung, Peter Shirley, steven Baer : Interactive Computer Graphics, concepts and applications, Cengage Learning
4. M M Raikar & Shreedhara K S Computer Graphics using OpenGL, Cengage publication

Rapid Programming Application using Python

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Subject Code	**18CS631**	**CIE Marks**	**50**
Number Lecture Hour/Week	**03**	**SEE Marks**	**50**
Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS-03

Course Objectives:

To equip students with fundamental python programming skills, enabling them to design, implement, and debug programs

To understand core concepts like control flow and object-oriented programming

Illustrate the process of structuring the data using lists, tuples and dictionaries. Demonstrate the use of built-in functions to navigate the file system.

Determine the need for scraping websites and working with CSV, JSON.

Modules	**Hours**
Module I	
Python Basics: Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data	

Types, String Concatenation and Replication, Storing Values in Variables, Your First Program, Dissecting Your Program **Flow control:**Boolean Values, Comparison Operators, Boolean Operators,Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution, Flow Control Statements, Importing Modules,Ending a Program Early with sys.exit() **Functions:**def Statements with Parameters, Return Values and return Statements,The None Value, Keyword Arguments and print(), Local and Global Scope, The global Statement, Exception Handling, A Short Program: Guess the Number	8
Module II	
Lists: The List Data Type, Working with Lists, Augmented Assignment Operators, Methods, Example Program: Magic 8 Ball with a List, List-like Types: Strings and Tuples,References, **Dictionaries and Structuring Data:**The Dictionary Data Type, Pretty Printing, Using Data Structures to Model Real-World Things, **Manipulating Strings:**Working with Strings, Useful String Methods, Project: Password Locker, Project: Adding Bullets to Wiki Markup	8
Module III	
Pattern Matching with Regular Expressions: Finding Patterns of Text Without Regular Expressions, Finding Patterns of Text with Regular Expressions,More Pattern Matching with Regular Expressions, Greedy and Nongreedy Matching, The findall() Method, Character Classes, Making Your Own Character Classes, The Caret and Dollar Sign Characters, The Wildcard Character, Review of Regex Symbols, Case-Insensitive Matching, Substituting Strings with the sub() Method, Managing Complex Regexes, Combining re .IGNORECASE, re .DOTALL, and re .VERBOSE, Project: Phone Number and Email Address Extractor, **Reading and Writing Files:**Files and File Paths, The os.path Module, The File Reading/Writing Process, Saving Variables with the shelve Module,Saving Variables with the pprint.pformat() Function, Project: Generating Random Quiz Files, Project: Multiclipboard, **Organizing Files:**The shutil Module, Walking a Directory Tree, Compressing Files with the zipfile Module, Project: Renaming Files with American-Style Dates to European-Style Dates,Project: Backing Up a Folder into a ZIP File, D**ebugging:**Raising Exceptions, Getting the Traceback as a String, Assertions, Logging, IDLE's Debugger.	8
Module IV	
Classes and objects: Programmer-defined types, Attributes, Rectangles, Instances as return values, Objects are mutable, Copying, **Classes and functions:**Time, Pure functions, Modifiers, Prototyping versus planning, **Classes and methods:**Object-oriented features, Printing objects, Another example, A more complicated example,The init method, The __str__ method, Operator overloading, Type-based dispatch, Polymorphism, Interface and implementation, **Inheritance:**Card objects, Class attributes, Comparing cards, Decks, Printing the deck, Add, remove, shuffle and sort, Inheritance, Class diagrams, Data encapsulation	8
Module V	
Web Scraping: Project: MAPIT.PY with the webbrowser Module, Downloading Files from the Web with the requests Module, Saving Downloaded Files to the Hard Drive, HTML, Parsing HTML with the BeautifulSoup Module, Project: "I'm Feeling Lucky" Google Search,Project: Downloading All XKCD Comics, Controlling the Browser with the selenium Module, **Working with Excel Spreadsheets:**Excel Documents, Installing the openpyxl Module, Reading Excel Documents, Project: Reading Data from a Spreadsheet, Writing Excel Documents, Project: Updating a Spreadsheet, Setting the Font Style of Cells, Font Objects, Formulas, Adjusting Rows and Columns, Charts, **Working with PDF and Word Documents:**PDF Documents, Project: Combining Select Pages from Many PDFs, Word Documents **Working with CSV files and JSON data:**The csv Module, Project: Removing the Header from CSV Files, JSON and APIs, The json Module, Project: Fetching Current Weather Data	8

Course Outcomes (COs):

CO1	Create basic programs using variables, conditionals, loops, and functions.
CO2	Use lists, tuples, and dictionaries in Python programs.
CO3	Use Python for regex pattern matching, file manipulation, efficient organization, and debugging to solve computational tasks.
CO4	Utilize the concepts of Object-Oriented Programming in Python.
CO5	Develop python programs to perform web scraping, manipulate excel, word, pdf,CSV,json files for data processing and automation

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	2	-	1	-	-	-	1	-	-	1	1	3	1
CO2	3	2	2	-	1	-	-	-	1	-	-	1	1	3	1
CO3	3	2	2	-	1	-	-	-	1	-	-	1	1	3	1
C04	3	2	2	-	1	-	-	-	1	-	-	1	1	3	1
CO5	3	2	2	-	1	-	-	-	1	-	-	1	1	3	1

Question Paper Pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

1. Charles R. Severance, Python for Everybody: Exploring Data Using Python 3, 1st Edition, CreateSpace Independent Publishing Platform, 2016. (http://do1.dr-chuck.com/pythonlearn/ENus/pythonlearn.pdf) (Chapters 1 – 13, 15)

2. Allen B. Downey, “Think Python: How to Think Like a Computer Scientist, 2ndEdition, Green Tea Press, 2015. (http://greenteapress.com/thinkpython2/thinkpython2.pdf) (Chapters 15, 16, 17) (Downloadpdf files from the above links.

Reference Books:

1 .Charles Dierbach, “Introduction to Computer Science Using Python”, 1st Edition, Wiley India Pvt Ltd. ISBN-13: 978-812655601

2 Mark Lutz, Programming Python, 4th Edition, O’Reilly Media, 2011.ISBN-13: 978-9350232873

3 Wesley J Chun, Core Python Applications Programming, 3rdEdition, Pearson Education India, 2015. ISBN-13: 978-9332555365

4 Roberto Tamassia, Michael H Goldwasser, Michael T Goodrich, Data Structures and Algorithms in Python,1stEdition, Wiley India Pvt Ltd, 2016. ISBN-13: 9788126562176

5 ReemaThareja, Python Programming using problem solving approach, Oxford university press, 2017

SENSORS AND APPLICATIONS

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Subject Code	**18CS632**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**40**	**Exam Hours**	03

CREDITS – 03

Course Objectives: This course will enable students

- To learn the fundamentals of wireless sensor networks and applications
- To study factors and design of wireless sensor networks.
- To medium access control protocols and network layer.
- To study Transport Layer and application layer Protocols.
- To study synchronization and localization of wireless sensor networks

Modules	Hours
Module I	
Introduction: Sensor Mote Platforms, WSN Architecture and Protocol Stack **WSN Applications**: Military Applications, Environmental Applications, Health Applications,Home Applications, Industrial Applications	8
Module II	
Factors Influencing WSN Design: Hardware Constraints Fault Tolerance Scalability Production Costs WSN Topology, Transmission Media, Power Consumption **Physical Layer**: Physical Layer Technologies, Overview of RF Wireless Communication, Channel Coding (Error Control Coding), Modulation, Wireless Channel Effects, PHY Layer Standards	8
Module III	
Medium Access Control: Challenges for MAC, CSMA Mechanism, Contention-Based Medium Access, Reservation-Based Medium Access, Hybrid Medium Access **Network Layer**: Challenges for Routing, Data-centric and Flat Architecture Protocols, Hierarchical Protocols, Geographical Routing Protocols	8
Module IV	

Transport Layer: Challenges for Transport Layer, Reliable Multisegmented Transport (RMST) Protocol, Pump Slowly, Fetch Quickly (PSFQ) Protocol, Congestion Detection and Avoidance (CODA) Protocol, Event-to-Sink Reliable Transport (ESRT) Protocol, GARUDA). **Application Layer**: Source Coding (Data Compression), Query Processing, Network Management	8
Module V	
Time Synchronization: Challenges for Time Synchronization, Network Time Protocol, Timing-Sync Protocol for Sensor Networks (TPSN), Reference- Broadcast Synchronization (RBS), Adaptive Clock Synchronization (ACS)**Localization**; Challenges in Localization, Ranging Techniques, Range-Based Localization Protocols, Range-Free Localization Protocols.	8

Course Outcomes (COs):

CO1	Acquire knowledge of characteristics of mobile/wireless communication channels
CO2	Apply statistical models of multipath fading
CO3	Understand the multiple radio access techniques, radio standards and communication protocols to be

CO4	used for wireless sensor
CO5	Design wireless sensor network system for different applications under consideration.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	-	-	1	-	-	-	-	-	-	-	-	1	3	-
CO2	-	2	1	-		-	-	-	-	-	-	-	1	3	-
CO3	3	-	-	-	1	-	-	-	-	-	-	-	1	3	-
C04	-	-	1	1	-	-	-	-	-	-	-	-	1	3	-
CO5	2	1	-	-	-	-	-	-	-	-	-	-	1	3	-

Question Paper Pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

1. Wireless Sensor Networks’, Ian F. Akyildiz and Mehmet Can Vuran, John Wiley & Sons Ltd. ISBN 978- 0-470-03601-3 (H/B), 2010

2. ‘Wireless Sensor Networks:Signal Processing and Communications Perspectives’, Ananthram Swami, et. al., John Wiley & Sons Ltd., ISBN 978-0470-03557-3, 2007

Computer Vision

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)
SEMESTER – VI

Subject Code	**18CS633**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**40**	**Exam Hours**	03

CREDITS – 03

Course Objectives: This course will enable students to

- To understand the fundamentals of image formation.
- To understand major ideas, methods and techniques of computer vision.
- To develop an appreciation for various issues in the design of computer vision and object recognition systems
- To provide programming experience from implementing computer vision and object recognition applications.

Modules	**Hours**
Module I	
Image Formation Models: Monocular imaging system, Orthographic& Perspective Projection, Camera model and Camera calibration, B inocular imaging systems.	8
Module II	
Image Processing and Feature Extraction: Image representations (continuous and discrete), Edge Detection	8
Module III	
Motion Estimation: Regularization theory, Optical computation, Stereo Vision, Motion estimation, Structure from motion	8
Module IV	
Shape Representation and Segmentation: Deformable curves and surfaces, Snakes and active contours, Level set representations, Fourier and wavelet descriptors, Medial representations, Multiresolution Analysis	8
Module V	
Object recognition: Hough transforms and other simple object recognition methods, Shapecorrespondence and shape matching, Principal Component analysis, Shape priors for recognition.	8

Course Outcomes (COs):

CO1	Identify basic concepts, terminology, theories, models and methods in the field of computer vision, ·
CO2	Perform feature extraction and edge detection for given images.
CO3	Estimate 3D structures from 2D image sequences.
CO4	Apply different segmentation and medial representations for the images.
CO5	Recognize object in a given image using different techniques.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	P O9	PO10	PO11	PO12	PSO1	PSO2	PSO 3
CO1	2	2	-	-	-	-	-	-	-	-	1	1	1	3	-
CO2	2	1	1	-	-	-	-	-	-	-	1	2	1	3	-
CO3	2	1	-	-	1	-	-	-	-	-	1	2	1	3	-
C04	2	2	1	-	-	-	-	-	-	-	1	2	1	3	-
CO5	2	1	-	-	-	-	-	-	-	-	1	2	1	3	-

Question Paper Pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module

Text Books: 1. Computer Vision - A modern approach, by D. Forsyth and J. Ponce, Prentice Hall Robot Vision, by B. K. P. Horn, McGraw-Hill.
Reference Books: 1. Richard Szeliksy “Computer Vision: Algorithms and Applications” (http://szeliski.org/Book/) 2. Haralick& Shapiro, “Computer and Robot Vision”, Vol II 3. G_erardMedioni and Sing Bing Kang “Emerging topics in computer vision” 4. Emanuele Trucco and AllessandroVerri “Introductory Techniques for 3-D Computer Vision”, Prentice Hall, 1998. 5. Olivier Faugeras, “Three-Dimensional Computer Vision”, The MIT Press, 1993.

PROBABILITY STATISTICS AND QUEUING THEORY

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Subject Code	**18CS634**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**40**	**Exam Hours**	03

CREDITS – 03

Course Objectives: This course will enable students

- Develop analytical capability and impart knowledge of Statistics and queuing probability.
- Apply above concepts in Engineering and Technology.
- Acquire knowledge of Hypothesis testing and Queuing methods and their applications so as to enable them to apply them for solving real world problems

Modules	**Hours**
Module I	
Axioms of probability, Conditional probability, Total probability, Baye’s theorem, Discrete Random variable, Probability mass function, Continuous Random variable. Probability density function, Cumulative Distribution Function, and its properties, Two-dimensional Random variables, Joint pdf / cdf and their properties	8
Module II	

Probability Distributions / Discrete distributions: Binomial, Poisson Geometric and hyper- geometric distribution and their properties. continuous distribution. Uniform, Normal, exponential distributions and their properties	8
Module III	
Random Processes: Classification, methods of description, special classes, Average value of random processes, analytical representation of random processes, Autocorrelation	8
Module IV	
Function, cross-correlation function and their properties, Ergodicity, Poisson process, Markov Process, Markov chain	
Testing Hypothesis: Testing of Hypothesis: Formulation of Null Hypothesis, Critical region, level of significance, errors in testing, Tests of significance for Large and Small Samples, t-distribution, its properties and uses, F-distribution, its properties and uses, Chi- square distribution, its properties and uses, $\chi 2$ – test for goodness of fit, $\chi 2$ test for Independence	8
Module V	
Symbolic Representation of a Queuing Model, Poisson Queue system, Little Law, Types of Stochastic Processes, Birth-Death Process, The M/M/1 Queuing System, The M/M/s Queuing System, The M/M/s Queuing with Finite buffers.	8

Course Outcomes (COs):

CO1	Demonstrate use of probability and characterize probability models using probability mass (density) functions & cumulative distribution functions
CO2	Explain the techniques of developing discrete & continuous probability distributions and its applications.
CO3	Describe a random process in terms of its mean and correlation functions .
CO4	Outline methods of Hypothesis testing for goodness of fit.
CO5	Define the terminology &nomenclature appropriate queuing theory and also distinguish various queuing models.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	-	3	-	-	-	-	-	2	-	3	2	1	-
CO2	3	3	-	3	-	-	-	-	-	2	-	3	2	1	-
CO3	3	3	-	3	-	-	-	-	-	2	-	3	2	1	-
C04	3	3	-	3	-	-	-	-	-	2	-	3	2	1	-
CO5	3	3	-	-	-	-	-	-	-	1	-	3	2	1	-

Question Paper Pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

1. Probability, Statistics and Queuing Theory, V. Sundarapandian, Eastern Economy Edition,PHI Learning Pvt. Ltd, 2009

Reference Books:

1. Probability & Statistics with Reliability, Queuing and Computer Applications, 2nd Edition by Kishor. S. Trivedi , Prentice Hall of India ,2004.

2. Probability, Statistics and Random Processes, 1st Edition by P Kausalya, Pearson Education,2013.

SOFTWARE ENGINEERING

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020 -2021)

SEMESTER – VI

Subject Code	**18CS641**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**40**	**Exam Hours**	03

CREDITS – 03

Modules	**Hours**
Module II	
Introduction: Need for Software Engineering, Professional Software Development, Software Engineering Ethics. Case Studies.**Software Processes**: Models: Waterfall Model, Incremental Model and Spiral Model, Process activities.**Requirements Engineering**: Requirements Engineering Processes, Functional and non-functional requirements, The software Requirements Document Requirements Specification, Requirements validation, Requirements Management	8
Module II	
System Models: Context models, Interaction models, Structural models, Behavioral models, Model-driven engineering. **Architectural Design**: Architectural design decisions, Architectural patterns. **Design and Implementation**: Object-oriented design using the UML, Design patterns, Implementation issues.	8
Module III	
Software Testing: Development testing, Test-driven development, Release testing, User testing. **Software Evolution**: Evolution processes, Program evolution dynamics, Software maintenance, Legacy system management.	8
Module IV	
Project Planning: Software pricing, Plan-driven development. Project scheduling:Estimation techniques. **Quality management**: Software quality, Reviews and inspections Software measurement and metrics, Software standards	8
Module V	
Managing People: Selecting staff; Motivating people; Managing people; The People Capability Maturity Model. Software Cost Estimation: Productivity; Estimation techniques; Algorithmic cost modeling, Project duration and staffing.**Agile Software Development:** The Agile Manifesto: Values and Principles. Agile methods:SCRUM (Ref "The SCRUM Primer, Ver 2.0") and Extreme Programming, Plan-driven and agile development.	8

Course Outcomes (COs):

CO1	Apply software engineering principles & methodology to design, develop, test and maintain software system.
CO2	Analyse software requirements and create effective implementations using system models and architectural design.
CO3	Design and implement software architectures that meet performance security and scalability requirements.
CO4	Apply cost estimation and time scheduling for quality project activities.
CO5	Understand individual and team dynamics to manage conflicts, performance issues and embrace diversity and inclusivity in team management and apply agile principles and values to software development project.

CO-PO-PSO mapping:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**P O9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO 3**
CO1	3	-	-	-	1	-	-	-	1	1	-	1	1	3	-
CO2	3	2	-	-	1	-	-	-	1	1	-	1	1	3	-
CO3	3	2	3	-	1	-	-	-	1	1	-	1	1	3	-
C04	3	2	1	-	1	-	-	-	2	1	3	1	1	3	-
CO5	3	2	1	-	1	-	-	-	3	1	-	1	1	3	-

Question Paper Pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

1.Ian Sommerville: Software Engineering, 9th Edition, Pearson Education, 2012.The SCRUM Primer, Ver2.0,http://www.goodagile.com/scrumprimer/scrumprimer20.pdf

Reference Books:

1.Roger S. Pressman: Software Engineering-A Practitioners approach, 7th Edition, Tata McGrawHill.
2.Pankaj Jalote: An Integrated Approach to Software Engineering, WileyIndia

Multi Core Architecture

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020 -2021)

SEMESTER – VI

Subject Code	**18CS642**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**40**	**Exam Hours**	03

CREDITS – 03

Course Objectives: This course will enable students

- To Become familiar with the concepts of computer networks
- To understand technologies of multicore architecture and performance measures Demonstrate problems related to multiprocessing
- Illustrate windows threading, posix threads, openmp programming Analyze the common problems in parallel programming

Modules	Hours
Module I	
Introduction to Multi-core Architecture: Motivation for Concurrency in software, Parallel Computing Platforms, Parallel Computing in Microprocessors, Differentiating Multi-core Architectures from Hyper-Threading Technology, Multi-threading on Single-Core versus Multi- Core Platforms Understanding Performance, Amdahl's Law, Growing Returns: Gustafson's Law. **System Overview of Threading:** Defining Threads, System View of Threads, Threading above the Operating System, Threads inside the OS, Threads inside the Hardware, What Happens When a Thread Is Created, Application Programming Models and Threading, Virtual Environment: VMsand Platforms, Runtime Virtualization, System Virtualization.	8
Module II	
Fundamental Concepts of Parallel Programming: Designing for Threads, Task Decomposition, Data Decomposition, Data Flow Decomposition, Implications of Different Decompositions, Challenges You'll Face, Parallel Programming Patterns, A Motivating Problem: Error Diffusion, Analysis of the Error Diffusion **Algorithm, An Alternate Approach:** Parallel Error Diffusion, Other Alternatives. Threading and Parallel Programming Constructs: Synchronization, Critical Sections, Deadlock, Synchronization Primitives, Semaphores, Locks, Condition Variables, Messages, Flow Control- based Concepts, Fence, Barrier, Implementation-dependent Threading Features	8

Module III	
Threading APIs: Threading Apls for Microsoft Windows, Win32/MFC Thread APls, Threading APls for Microsoft. NET Framework, Creating Threads,Managing Threads, Thread Pools, Thread Synchronization, POSIX Threads, Creating Threads,Managing Threads, Thread Synchronization, Signaling, Compilation and Linking.	8
Module IV	
OpenMP: A Portable Solution for Threading: Challenges in Threading a Loop, Loop-carried Dependence, Data-race Conditions, Managing Shared and Private Data, Loop Scheduling and Portioning, Effective Use of Reductions, Minimizing Threading Overhead, Work-sharing Sections, Performance-oriented Programming, Using Barrier and No wait, Interleaving Single-thread and Multithread Execution, Data Copy-in and Copy-out, Protecting Updates of Shared Variables, Intel Task queuing Extension to OpenMP, OpenMP LibraryFunctions, OpenMP Environment Variables, Compilation, Debugging, performance	8
Module V	
Solutions to Common Parallel Programming Problems: Too Many Threads, Data Races, Deadlocks, and Live Locks, Deadlock, Heavily Contended Locks, Priority Inversion, Solutions for Heavily Contended Locks, Non-blockingAlgorithms, ABA Problem, Cache Line Ping-ponging, Memory Reclamation Problem, Recommendations, Thread-safe Functions and Libraries, Memory Issues, Bandwidth, Working in the Cache, Memory Contention, Cache-relatedIssues, False Sharing, Memory Consistency, Current IA-32 Architecture, Itanium Architecture, High-level Languages, Avoiding Pipeline Stalls on IA-32, Data Organization for High Performance.	8

Course Outcomes (COs):

CO1	Identify the issues involved in multicore architectures.
CO2	Explain fundamental concepts of parallel programming and its design issues.
CO3	Solve the issues related to multiprocessing and suggest solutions.
CO4	Discuss salient features of different multicore architectures and how they exploit.
CO5	Implement Parallelism.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	1	-	-	-	-	-	-	-	-	-	1	3	-
CO2	-	2	-	1	-	-	-	-	-	-	-	-	1	3	-
CO3	-	-	3	-	2	-	-	-	-	-	-	-	1	3	-
CO4	2	-	-	1	-	-	-	-	-	-	-	-	1	3	-
CO5	-	-	-	-	2	-	-	-	-	-	-	1	1	3	-

Question Paper Pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module

Text Book:

1.Multicore Programming, Increased Performance through Software Multi-threading by Shameem Akhter and Jason Roberts, Intel Press, 2006

Reference Books:

1.Roger S. Pressman: Software Engineering-A Practitioners approach, 7th Edition, Tata McGrawHill**.**
2.Pankaj Jalote: An Integrated Approach to Software Engineering, WileyIndia

Network Programming

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)
SEMESTER – VI

Subject Code	**18CS643**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**40**	**Exam Hours**	03

CREDITS – 03

Course Objectives: This course will enable students

- Define Network Programming.
- Demonstrate programming with TCP and SCTP.
- Explain key management and routing sockets.
- Evaluate advanced Socket Programming APIs.

Modules	**Hours**
Module I	
Introduction to network application: client/server communication, OSI Model, BSD Networking history, Test Networks and Hosts, Unix Standards, 64-bit architectures, Transport Layer: TCP, UDP and SCTP.	8
Module II	
Sockets Introduction – socket address structures, value-result arguments, byte ordering and manipulation functions, address conversion functions, Elementary TCP Sockets – socket,connect, bind, listen, accept , fork and concurrent server design, getcsockname and getpeername functions and TCP Client/Server Example- client/server programming through TCP sockets, Normal startup, termination, POSIX signal handling, Signal handling in server,Crashing, rebooting of server host, shutdown.	8
Module III	
I/O Multiplexing and Socket Options, Elementary SCTP Sockets- Interface Models,STCP_X functions, shutdown function, Notifications, SCTP Client/Server Examples – One to-Many, Head–of-Line Blocking, Controlling number of streams and Termination, IPv4 and IPv6 Interoperability–different interoperability scenarios.	8
Module IV	
Daemon Processes, syslog, demonizing functions and the intend super server, Advanced I/O functions-ready, write, send messages and receive message, Ancillary data, Advanced polling, Unix domain protocols- socket address structure, functions and communication scenarios, Nonblocking I/O – connect and accept examples	8
Module V	

IOCTL operations- socket, file, interface configuration information, ARP cache and routing table operations, Routing sockets- data link socket address structure, reading and writing, systole operations, interface name and index functions, Key Management functions –reading, writing, SADB, SA, Dynamically Maintaining SA's, Out-of-Band data, Threads-basic thread functions, TCP echo server using threads, Mutexes and Conditional variables.	8

Course Outcomes (COs):

CO1	Understand the principles of the application layer protocols HTTP, FTP, SMTP and DNS.
CO2	Understand the transport layer services, TCP and UDP protocols.
CO3	Understand the router architecture, IP and routing algorithms.
CO4	Understand the concepts of Network security and cryptography protocols.
CO5	Understand the multimedia network applications, audio, video streaming.

CO-PO-PSO mapping:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	3	-	1	-	-	-	-	-	-	-	-	-	1	3	-
CO2	-	2	-	1	-	-	-	-	-	-	-	-	1	3	-
CO3	-	-	3	-	2	-	-	-	-	-	-	-	1	3	-
C04	2	-	-	1	-	-	-	-	-	-	-	-	1	3	-
CO5	-	-	-	-	2	-	-	-	-	-	-	1	1	3	-

Question Paper Pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module

Text Book:

1.W. Richard Stevens, Bill Fenner, Andrew M. Rudoff: "UNIX Network Programming". Volume 1, Third Edition, Pearson 2004

Reference Books:

1. Barry Nance: “Network Programming in C”, PHI 2002 3. Bob Quinn, Dave Shute: “Windows Socket Network Programming”, Pearson 2003.

2. Richard Stevens: “UNIX Network Programming”. Volume 2, Second Edition

MOBILE COMPUTING

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)
SEMESTER – VI

Subject Code	**18CS644**	**CIE Marks**	**50**
Number Lecture Hour/Week	**03**	**SEE Marks**	**50**
Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS-03

Course Objectives: This course will enable students to:

- Define concepts of wireless communication.
- Compare and contrast propagation methods, Channel models, capacity calculations multiple antennas and multiple user techniques used in the mobile communication.
- Explain CDMA, GSM. Mobile IP, WiMAX and Different Mobile OS
- Illustrate various Markup Languages CDC, CLDC, MIDP; Programming for CLDC, MIDlet model and security concerns

Modules	**Hours**
Module I	
Mobile Computing Architecture: Architecture for Mobile Computing, 3-tier Architecture, Design Considerations for Mobile Computing. Emerging Technologies: Wireless broadband(WiMAX), Mobile IP: Introduction, discovery, Registration, Tunneling, Cellular IP, MobileIP with IPv6. Wireless Networks : Global Systems for Mobile Communication (GSM): GSMArchitecture, Entities, Call routing in GSM, PLMN Interface, GSM Addresses and Identities,Network Aspects in GSM, Mobility Management, GSM Frequency allocation. Short ServiceMessages (SMS): Introduction to SMS, SMS Architecture, SMMT, SMMO, SMS asInformation bearer, applications	08
Module II	
GPRS and Packet Data Network, GPRS Network Architecture, GPRS Network Operations, Data Services in GPRS, Applications for GPRS, Billing and Charging in GPRS. Spread Spectrum technology, IS-95, CDMA versus GSM, Wireless Data, Third Generation Networks, Applications on 3G, Mobile Client: Moving beyond desktop, Mobile handset overview, Mobile phones and their features, PDA, Design Constraints in applications for handheld devices.	8
Module III	

Mobile OS and Computing Environment: Smart Client Architecture, The Client: User Interface, Data Storage, Performance, Data Synchronization, Messaging. The Server: Data Synchronization, Enterprise Data Source, Messaging. Mobile Operating Systems: Wince, Palm OS, Symbian OS, Linux, Proprietary OS Client Development: The development process, Need analysis phase, Design phase, Implementation and Testing phase, Deployment phase, Development Tools, Device Emulators	8
Module IV	
Building Wireless Internet Applications: Thin client overview: Architecture, the client, Middleware, messaging Servers, Processing a Wireless request, Wireless Applications Protocol (WAP) Overview, Wireless Languages: Markup Languages, HDML, WML, HTML, cHTML, XHTML, Voice XML.	8
Module V	
J2ME: Introduction, CDC, CLDC, MIDP; Programming for CLDC, MIDlet model, Provisioning, MIDlet life-cycle, Creating new application, MIDlet event handling, GUI in MIDP, Low level GUI Components, Multimedia APIs; Communication in MIDP, Security Considerations in MIDP.	8

Course Outcomes (COs):

CO1	Explain state of art techniques in wireless communication.
CO2	Discuss CDMA, GSM. Mobile IP, WImax.
CO3	Design, develop, and deploy mobile applications across various operating systems.
CO4	Design and build wireless internet applications using thin client architecture, various wireless languages, and protocols.
CO5	Demonstrate program for CLDC, MIDP let model and security concerns.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	1	1	1	2	1	-	-	-	1	1	3	-
CO2	3	3	2	1	2	1	1	1	-	-	-	1	1	3	-
CO3	2	2	3	2	3	1	2	1	-	-	-	1	1	3	1
C04	2	2	3	2	3	1	1	1	-	-	-	1	1	3	1
CO5	1	2	3	1	2	3	2	1	-	-	-	1	1	3	1

Question Paper Pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

1. Ashok Talukder, RoopaYavagal, Hasan Ahmed: Mobile Computing, Technology, Applications and Service Creation, 2nd Edition, Tata McGraw Hill, 2010

2. Martyn Mallik: Mobile and Wireless Design Essentials, Wiley India, 2003

Reference Books:

1. Raj kamal: Mobile Computing, Oxford University Press, 2007ItiSahaMisra: Wireless Communications
2. and Networks, 3G and Beyond, Tata McGraw Hill,2009

System Software and Compiler Design Lab

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Subject Code	**18CSL65**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**30**	**Exam Hours**	03

CREDITS – 01

Course Objectives: This course will enable students

- To make students familiar with Lexical Analysis and Syntax Analysis phases of Compiler Design and implement programs on these phases using LEX & YACC tools and/or C/C++/Java
- To enable students to learn different types of CPU scheduling algorithms used in operating system.
- To make students able to implement memory management - page replacement and deadlock handling algorithms

PART A

Execute the following programs using LEX:

1.

a. Program to count the number of characters, words, spaces and lines in a given input file.

b. Program to count the numbers of comment lines in a given C program. Also eliminate them and copy the resulting program into separate file.

2.

a. Program to recognize a valid arithmetic expression and to recognize the identifiers and operators present. Print them separately.

b. Program to recognize whether a given sentence is simple or compound.

3. Program to recognize and count the number of identifiers in a given input file.

Execute the following programs using YACC:

4.

a. Program to recognize a valid arithmetic expression that uses operators +, -, * and /.

b. Program to recognize a valid variable, which starts with a letter, followed by any number of letters or digits.

5.

a. Program to evaluate an arithmetic expression involving operators +, -, * and /.

b. Program to recognize strings ‘aaab’, ‘abbb’, ‘ab’ and ‘a’ using the grammar (anbn, n>= 0).

6. Program to recognize the grammar (anb, n>= 10).

PART B

7. Design, develop and implement program to construct Predictive / LL(1)Parsing Table for the grammar rules: □□→ □□□,□□□→ □□|ε . Use this table to parsethe sentence: abba$

8. Design, develop and implement program to demonstrate Shift Reduce Parsing technique for the grammar rules: □□→ □□+ □|□□□→ □□∗ □|□,F→ ( ) □□|□ and parse the sentence: id + id * id.

9. Design, develop and implement syntax-directed definition of “if E then S1” and “if E then S1 else S2”

10. Write a yacc program that accepts a regular expression as input and produce its parse tree as output.

11. Design, develop and implement a program to generate the machine code usingTriples for the statement A = -B * (C +D) whose intermediate code in three-addressform:

T1 = -B

T2 = C + D

T3 = T1 + T2

A = T3

Course Outcomes (COs):

CO1	Demonstrate theoretical concept of System Software and Compiler Design through series of experiment
CO2	Develop a program using software tools.

CO3	Debug and troubleshoot issues effectively.
CO4	Analyze the data and interpret the results.
CO5	Prepare a well-organized laboratory report.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	-	-	-	-	-	-	-	-	2	1	3	3
CO2	1	1	3	-	3	-	-	-	-	-	-	2	1	3	3
CO3	2	3	-	-	1	-	-	-	-	-	-	-	1	2	3
C04	1	2	2	-	-	-	-	-	-	-	-	-	1	1	3
CO5	1	-	-	-	-	-	-	-	-	3	-	-	1	-	3

Conduct of Practical Examination:

- Experiment distribution

- For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
- For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

- Marks Distribution (Courseed to change in accoradance with university regulations)
- For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+30+5 = 50 Marks
- For laboratories having PART A and PART B

Part A – Procedure + Execution + Viva = 7 + 20 + 3 = 30 Marks

x. Part B – Procedure + Execution + Viva = 6 + 12 + 2= 20 Marks

Operating System and UNIX Programming Lab

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Subject Code	**18CSL66**	**CIE Marks**	50
Number of Lecture Hours/Week	**03**	**SEE Marks**	50
Total Number of Lecture Hours	**30**	**Exam Hours**	03

CREDITS – 01

Course Objectives: This course will enable students

To learn the fundamentals of Operating Systems.

To learn the mechanisms of OS to handle processes and threads and their communication

To learn the mechanisms involved in memory management in contemporary OS

- To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols

To know the components and management aspects of concurrency management

To learn programmatically to implement simple OS mechanisms

PART A

1. Implementation of CPU Scheduling Algorithms.
2. Implementation of Semaphores.
3. Implementation of Shared memory and IPC.
4. Implementation of Bankers Algorithm for Deadlock Avoidance.
5. Implementation of Deadlock Detection Algorithm.
6. Implementation of Threading and Synchronization Applications.
7. Implementation of the following Memory Allocation Methods for fixed partition.
8. Implementation of Paging Technique of Memory Management.
9. Implementation of the various File Organization Techniques.
10. Implementation of the following Page Replacement Algorithms.

PART B

1. Design a program that creates a zombie and then calls system to execute the PS command to verify that the process is zombie.
2. Design a program which demonstrates interposes communication between a reader process and a writer process. Use mkfifo, open, read, write and close APIs in your program.
3. Design a program to illustrate the race condition.
4. Design a program that creates a zombie and then calls system to execute the ps command to verify that the process is zombie.
5. Design a program to avoid zombie process by forking twice.
6. Design a program to implement the system function.
7. Design a program to set up a real-time clock interval timer using the alarm API

Course Outcomes (COs):

CO1	Demonstrate various operating system concepts through a series of experiments for managing system resources.
CO2	Develop operating system programs using software tools.
CO3	Debug and troubleshoot software issues effectively.
CO4	Analyze the data and interpret the results.
CO5	Prepare a well organized laboratory report.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	-	-	-	-	-	-	-	-	-	3	3	2
CO2	2	1	3	-	-	-	-	-	-	-	-	-	2	3	2
CO3	3	3	2	-	-	-	-	-	-	-	-	-	2	3	2
C04	3	3	2	-	-	-	-	-	-	-	-	-	2	3	2
CO5	1	1	1	-	-	-	-	-	-	-	-	-	1	3	2

Conduct of Practical Examination:

- Experiment distribution

- For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
- For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

- Marks Distribution (Courseed to change in accoradance with university regulations)

- For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+30+5 = 50 Marks

- For laboratories having PART A and PART B

Part A – Procedure + Execution + Viva = 7 + 20 + 3 = 30 Marks

xii. Part B – Procedure + Execution + Viva = 6 + 12 + 2= 20 Marks

PYTHON LAB

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Subject Code	**18CSL67**	**CIE Marks**	50
Number of Lecture Hours/Week	**02**	**SEE Marks**	50
Total Number of Lecture Hours	**30**	**Exam Hours**	03

CREDITS – 01

Course Objectives:

- To be able to introduce core programming basics and program design with functions using Python programming language.
- To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.
- To understand the high-performance programs designed to strengthen the practical expertise.

PROGRAMS

1. Write a program to demonstrate different number data types in Python. and perform different Arithmetic Operations on numbers in Python.
2. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
3. Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST 2017”
4. Write a program to create, append, and remove lists in python.
5. Write a program to demonstrate working with tuples in python.
6. Write a program to demonstrate working with dictionaries in python.
7. Write a python program to define a module and import a specific function in that module to another program.
8. Using Regular expressions, develop a Python program to
   - Identify a word with a sequence of one upper case letter followed by lower case letters.
   - Find all the patterns of “1(0+)1” in a given string.
9. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
10. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.

Course Outcomes (COs):

CO1	Demonstrate theoretical concepts of Python strings, lists, tuples, functions and file manipulation through series of programs.
CO2	Design and develop solutions to given problems using Python.
CO3	Debug syntactical errors, and troubleshoot programming issues effectively.
CO4	Analyze the programs and interpret the results
CO5	Prepare a well-organized Python programming laboratory report

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	1	2	-	2	-	-	-	1	3	-	1	1	3	3
CO2	2	2	2	-	3	-	-	-	1	3	-	1	1	3	3
CO3	2	2	1	-	2	-	-	-	1	-	-	1	1	3	3
C04	2	3	1	-	2	-	-	-	1	-	-	1	1	3	3
CO5	-	-	-	-	-	-	-	-	-	3	-	-	-	-	-

Conduct of Practical Examination:

- Experiment distribution

- For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
- For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

- Marks Distribution (Courseed to change in accoradance with university regulations)
- For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+30+5 = 50 Marks
- For laboratories having PART A and PART B

Part A – Procedure + Execution + Viva = 7 + 20 + 3 = 30 Marks

xiv. Part B – Procedure + Execution + Viva = 6 + 12 + 2= 20 Marks

PROJECT -6

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Subject Code	18CSL68	**CIE Marks**	50
Number of Lecture Hours/Week	02	**SEE Marks**	50
Total Number of Lecture Hours	30	**Exam Hours**	03
CREDITS – 01			

Course Objectives: This course will enable students

- Identify real-world problems across programming, databases, and networking domains and understand their business and technical implications.
- Apply systematic methodologies to design, implement, and optimize solutions.
- Resolve technical challenges through debugging, research, and collaboration.
- Take responsibility for specific roles in a team and collaborate effectively to achieve project goals.
- Present project progress and findings clearly and confidently to both technical and non-technical audiences.
- Document the entire project in a structured, professional laboratory report.

Project Guidelines:

- Project work shall preferably be batch wise.
- Evaluation is based on concept clarity, system design, implementation, testing, presentation, and documentation quality, with a focus on proper coding standards, teamwork, and effective communication.
- Viva-voce examination in project work shall be conducted batch-wise.
- Minimum requirement of CIE marks for Project work shall be 50% of the maximum marks.
- Students failing to secure a minimum of 50% of the CIE marks in Project work shall not be eligible for the SEE Project examination.
- For a pass in a Project/Viva-voce examination, a student shall secure a minimum of 40% of the maximum marks prescribed.

Course Outcomes (COs):

CO1	Identify the topic from various domains (example programming databases, networking) to real world problems.
CO2	Develop methodology for the problem.
CO3	Resolve issues that arise during the project .
CO4	Learn to assign and accept roles and responsibilities within a team and write a good technical reports.
CO5	Exhibit skills in presenting their project findings & progress orally

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	2	2	-	1	1	1	2	2	1	2	2	3	2
CO2	2	2	2	2	2	1	2	1	2	2	1	1	2	2	2
CO3	2	2	2	2	3	1	1	1	2	2	1	1	2	3	2
C04	-	-	-	-	-	1	-	2	2	3	1	1	1	1	1
CO5	-	-	-	-	1	1	-	2	2	3	1	1	1	1	1

AWS CLOUD FOUNDATION

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2020-2021)

SEMESTER – VI

Course Code	**18CSE69**	**CIE Marks**	50
Number of Contact Hours/Week	**03**	**SEE Marks**	50
Total Number of Contact Hours	**40**	**Exam Hours**	03

CREDITS –01

Course Objectives: This course will enable students to:

- Describe the security and compliance measures of the AWS Cloud, including AWS Identity and Access Management (IAM)
- Create a virtual private cloud (VPC) by using Amazon Virtual Private Cloud (Amazon VPC)
- Demonstrate when to use Amazon Elastic Compute Cloud (Amazon EC2), AWS Lambda, and AWS Elastic Beanstalk
- Differentiate between Amazon Simple Storage Service (Amazon S3), Amazon Elastic Block Store (Amazon EBS), Amazon Elastic File System (Amazon EFS), and Amazon Simple Storage Service Glacier (Amazon S3 Glacier)
- Demonstrate when to use AWS database services, including Amazon Relational Database Service (Amazon RDS), Amazon DynamoDB, Amazon Redshift, and Amazon Aurora
- Explain the architectural principles of the AWS Cloud
- Explore key concepts related to Elastic Load Balancing, Amazon CloudWatch, and Amazon EC2Auto Scaling

Modules	**Hours**
Module I	
Cloud Concepts Overview: Introduction to Cloud Computing, Advantages of the Cloud, Introduction to AWS, Moving to the AWS Cloud, **Cloud Economics and Billing**: Introduction Fundamentals of Pricing, Total Cost of Ownership, Simple Monthly Calculator, Delaware North Case Study, AWS Organizations, AWS Billing and Cost Management, Billing Dashboard	**8**
Module II	

AWS Global Infrastructure Overview: Introduction, AWS Global Infrastructure, AWS Global Infrastructure, AWS Services and Service Categories, AWS Management Console Clickthrough. **Cloud Security:** Introduction, AWS Shared Responsibility Model, AWS Shared Responsibility Model, AWS IAM, AWS IAM Console Demonstration, Securing a New AWS Account, Introduction to AWS IAM, Securing Accounts, Securing Data, Working to Ensure Compliance	**8**
Module III	
Networking and Content Delivery: Introduction, Networking Basics, Amazon VPC, VPC Networking, Label This Diagram, Amazon VPC Console Demonstration, VPC Security, Design a VPC Build a VPC and Launch a Web Server, Route 53, CloudFront. **Compute:** Introduction, Compute Services Overview, Amazon EC2 Part 1, Amazon EC2 Part 2, Amazon EC2 Part 3, Introduction to Amazon EC2, Amazon EC2 versus Managed Services, Amazon EC2 Part Console Demonstration, Amazon EC2 Cost Optimization, Container Services, Introduction to AWS Lambda, AWS Lambda, Introduction to AWS Elastic Beanstalk, AWS Elastic Beanstalk	**8**
Module IV	
Storage: Introduction, AWS EBS, Amazon Elastic Block Store Console, Demonstration, Working with EBS, AWS S3, AWS S3 Console Demonstration, AWS EFS, AWS EFS Console Demonstration, AWS S3 Glacier, AWS S3 Glacier Console Demonstration, Storage Technology Selection**. Databases:** Introduction, Amazon RDS, Amazon RDS Console Demonstration, Build a Database Server, Amazon DynamoDB, Amazon DynamoDB Demonstration, Amazon Redshift, Amazon Aurora, Database Case Study	**8**
Module V	
Cloud Architecture: Introduction, AWS Well-Architected Framework Design, Principles, AWS Well-Architected Framework Design, Principles, Operational Excellence, Security, Reliability Performance Efficiency, Cost Optimization, Reliability & High Availability, AWS Trusted Advisor, Interpret AWS Trusted Advisor Recommendations. **Automatic Scaling and Monitoring:** Introduction,Elastic Load Balancing, Elastic Load Balancing, Amazon CloudWatch, Amazon CloudWatch, Amazon EC2 Auto Scaling, Scale & Load Balance your Architecture	**8**

Course Outcomes (COs):

CO1	Analyze the fundamental concepts of cloud computing and its benefits.
CO2	Adapt knowledge of AWS global infrastructure and key AWS services.
CO3	Implement basic networking and content delivery concepts in AWS.
CO4	Collaborate various AWS storage options and manage databases in AWS
CO5	Design well-architected cloud solutions and implement automatic scaling and monitoring using AWS.

CO-PO-PSO mapping:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**P O9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO 3**
CO1	2	3	1	-	2	1	-	-	-	-	1	1	3	1	2
CO2	2	3	1	-	3	2	-	-	-	-	1	1	3	1	2
CO3	-	-	1	-	3	-	-	2	-	-	1	1	3	1	-
C04	3	-	1	-	3	-	-	-	-	-	1	1	3	1	3
CO5	-	-	1	-	3	-	-	-	-	-	1	1	3	1	-

Question Paper Pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module

Text Book:

1. RajkumarBuyya, Christian Vecchiola, and ThamaraiSelvi Mastering Cloud. Computing McGraw Hill Education

2.AWS Academy Cloud Foundations Course ,Introduction Student Guide Version 2.0.6

Reference Books:

1.Mark Wilkins, "Learning Amazon Web Services (AWS): A Hands-On Guide to the Fundamentals of AWS Cloud", Publisher(s): Addison-Wesley Professional, O'Reilly Media 2019
2."Mastering AWS Cost Optimization: Real-world technical and operational cost-saving best practices (Second Edition)", by Eli Mansoor and Yair Green 2020