



(An Autonomous Institution - AFFILIATED TO ANNA UNIVERSITY, CHENNAI)

S.P.G.Chidambara Nadar - C.Nagammal Campus

S.P.G.C. Nagar, K.Vellakulam – 625 701 (Near VIRUDHUNAGAR).

## **B.E. COMPUTER SCIENCE AND ENGINEERING**

**Regulation - 2020**

**AUTONOMOUS SYLLABUS CHOICE BASED CREDIT SYSTEM (CBCS)**

**CURRICULUM AND SYLLABI**

**(SEM V & VI)**

### **VISION:**

To make the Department of Computer Science and Engineering the unique of its kind in the field of Research and Development activities in this part of world.

### **MISSION:**

To impart highly innovative and technical knowledge to the urban and unreachable rural student folks in Computer Science and Engineering through "Total Quality Education".

### **PROGRAM EDUCATIONAL OBJECTIVES:**

#### **PEO 1:**

Apply the necessary mathematical tools and fundamental knowledge of computer science & engineering to solve variety of engineering problems.

#### **PEO 2:**

Develop software based solutions for real life problems and be leaders in their profession with social and ethical responsibilities.

#### **PEO 3:**

Pursue life-long learning and research in selected fields of computer science & engineering and contribute to the growth of those fields and society at large.

## PROGRAM OUTCOMES:

After going through the four years of study, the Computer Science and Engineering graduates will have the ability to

S.No	Graduate Attribute	Programme Outcome
1	Engineering knowledge	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2	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.
3	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.
4	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
5	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

6	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
7	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.
8	Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9	Individual and team work	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10	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.
11	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.
12	Life-long learning	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## **PROGRAMME SPECIFIC OUTCOMES (PSOs):**

### **PSO1 :**

**Professional Skills:** The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexity.

### **PSO2:**

**Problem - Solving Skills:** The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.

**B.E. COMPUTER SCIENCE AND ENGINEERING**  
**Regulation - 2020 Syllabus**  
**CHOICE BASED CREDIT SYSTEM (CBCS)**  
**SEMESTER V**

Sl. No	COURSE CODE	COURSE TITLE	CATEGORY	L	T	P	Contact Periods	C
<b>THEORY</b>								
1	IT1571	Computer Networks	PC	3	0	0	3	3
2	CS1501	Internet Programming	PC	3	0	0	3	3
3	CS1502	Theory of Computation and Compiler Design	PC	3	0	2	5	4
4	PE1	Professional Elective I	PE	3	0	0	3	3
5	PE2	Professional Elective II	PE	3	0	0	3	3
6	OE1	Open Elective – I	OE	3	0	0	3	3
		Online Course*						
<b>PRATICALS</b>								
7	CS1511	Internet Programming Lab	PC	0	0	4	4	2
8	IT1581	Computer Networks Laboratory	PC	0	0	4	4	2
<b>TOTAL</b>				<b>18</b>	<b>0</b>	<b>10</b>	<b>28</b>	<b>23</b>

**SEMESTER VI**

Sl. No	COURSE CODE	COURSE TITLE	CATEGORY	L	T	P	Contact Periods	C
<b>THEORY</b>								
1	CS1601	Distributed Systems	PC	3	0	0	3	3
2	CS1602	Internet of Things	PC	3	0	0	3	3
3	CS1603	Introduction to Machine learning	PC	3	0	0	3	3
4	PE3	Professional Elective III	PE	3	0	0	3	3
5	PE4	Professional Elective IV	PE	3	0	0	3	3
6	OL1	Online Course – I*	OL	0	0	0	0	3
<b>PRATICALS</b>								
7	HS1521	Professional Communication	EEC	0	0	2	2	1
8	CS1611	Internet of Things laboratory	PC	0	0	4	4	2
9	CS1612	Introduction to Machine Learning laboratory	PC	0	0	2	2	1
10	CS1681	Mobile Application Development laboratory	PC	0	0	4	4	2
<b>TOTAL</b>				<b>15</b>	<b>0</b>	<b>12</b>	<b>27</b>	<b>24</b>

\* Students shall complete online course in this semester. Credits earned will be added in consolidated mark statement.

**Professional Elective Courses (Elective – I, Semester V)**

SL.NO.	COURSE CODE	COURSE NAME	CREDITS				
			L	T	P	Contact Periods	C
1.	CS1531	Computer Graphics	3	0	0	3	3
2.	CS1532	Data Warehousing and Data Mining	3	0	0	3	3
3.	CS1533	Linux Programming and Shell Scripting	2	0	2	4	3
4.	CS1534	Real time systems	3	0	0	3	3
5.	CS1535	Software Testing and Quality Assurance	3	0	0	3	3

**Professional Elective Courses (Elective – II, Semester V)**

SL.NO	COURSE CODE	COURSE NAME	CREDITS				
			L	T	P	Contact Periods	C
1.	CS1536	Agile Development	3	0	0	3	3
2.	CS1537	C# and .NET Fundamentals	2	0	2	4	3
3.	CS1538	Data Science using R	2	0	2	4	3
4.	AD1503	Object Oriented Programming using Java	2	0	2	4	3
5.	CS1539	Multimedia Systems	3	0	0	3	3

**Professional Elective Courses (Elective – III, Semester VI)**

SL.NO	COURSE CODE	COURSE NAME	CREDITS				
			L	T	P	Contact Periods	C
1.	CS1631	Big Data Analytics – Tools and Techniques	2	0	2	4	3
2.	AD1534	Ethical Hacking	3	0	0	3	3
3.	CS1632	Game Design and Development	2	0	2	4	3
4.	CS1633	Multicore Architecture	3	0	0	3	3
5.	CS1634	Software Verification and Validation	3	0	0	3	3

**Professional Elective Courses (Elective – IV, Semester VI)**

SL.NO	COURSE CODE	COURSE NAME	CREDITS				
			L	T	P	Contact Periods	C
1.	CS1635	Augmented Reality and Virtual Reality	3	0	0	3	3
2.	CS1636	Full Stack Development	2	0	2	4	3
3.	CS1637	Mobile Adhoc Networks	3	0	0	3	3
4.	CS1638	Social Media Analytics	3	0	0	3	3
5.	CS1639	Software Design	3	0	0	3	3

**Suggested List of Open Electives**

**Open Elective – I (Semester V )**

SL NO.	COURSE CODE	COURSE NAME	CATEGORY	L	T	P	C
1	OCS151	Programming in Python	OE	3	0	0	3

L	T	P	C
3	0	0	3

**OBJECTIVES:**

To enable the students to

- Infer the importance of computer networks
- Apply error checking and error correction mechanisms in data Link layer
- Make use of various routing protocols
- Distinguish the functionalities of TCP & UDP protocols in the transport layer
- Summarize the functionalities and real time usage of various application layer protocols.

**UNIT I PHYSICAL LAYER****8**

Introduction: OSI – TCP/IP architectures – Networking: Topology – Types – Devices: Hubs, Bridges, Switches, Routers and Gateways – Transmission Media – Performance: jitter – delay – throughput – bandwidth X delay

**UNIT II DATALINK LAYER****10**

Layers: MAC and LLC – HDLC - Frame – Types of Frames – Framing - PPP – Services: Error Detection and Correction – Ethernet, IEEE 802.11, Bluetooth, Token Ring, FDDI, Link Layer Addressing: ARP

**UNIT III NETWORK LAYER****9**

IP address: IPv4 and IPv6 – Subnetting – CIDR - Packet switching – Services: Routing - Intra domain: Distance Vector – RIP, Link State - OSPF, Inter domain - BGP, Multicast, IP, DHCP, ICMP

**UNIT IV TRANSPORT LAYER****10**

Services: Flow Control, Congestion Control – QoS - Port Addressing, Protocols: TCP and UDP, SCTP

**UNIT V APPLICATION LAYER****8**

Protocols: HTTP, DNS, SMTP, MIME, POP, IMAP3, FTP, S-FTP, TELNET, SSH, SNMP

**TOTAL : 45 PERIODS**

**OUTCOMES:**

**Upon successful completion of the course, students will be able to**

- CO1: Infer the importance of computer networks with OSI and TCP/IP architectures.
- CO2: Apply error checking and error correction mechanisms in data Link layer for error free data transmission.
- CO3: Make use of various routing protocols and their strategies in the network.
- CO4: Compare the functionalities of TCP & UDP protocols in the transport layer during data transmission.
- CO5: Summarize the functionalities of various application layer protocols and their real time usage.

**TEXT BOOKS:**

1. Larry L Peterson & Bruce S Davie, 2011, *Computer Networks: A Systems Approach*, 5<sup>th</sup> ed, Morgan Kaufmann Publishers Inc.
2. Behrouz A Forouzan, 2013, *Data Communications and Networking*, 5<sup>th</sup> ed, TMH.

**REFERENCE BOOKS:**

1. William Stallings, 2014, *Data and Computer Communications*, 10<sup>th</sup> ed, Pearson Education.
2. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, 2012, *Computer Networks: An Open Source Approach*, McGraw Hill.



basics - XML: Basic XML- Document Type Definition- XML Schema - XML Parsers and Validation: DOM, SAX parsers– XSLT – Newsfeed (RSS and Atom)

## **UNIT V WEB SERVICES**

**7**

AJAX: Ajax Client Server Architecture - XML Http Request Object Call Back Methods - Web Services: Introduction- Java web services Basics – Describing web services using WSDL – Communication using SOAP – UDDI - Database Driven web service from an application.

**TOTAL: 45 PERIODS**

### **OUTCOMES:**

**Upon successful completion of the course, students will be able to**

- CO1: Build responsive web pages using HTML, CSS.
- CO2: Develop dynamic and interactive web pages by embedding JavaScript in HTML.
- CO3: Construct server side web applications using JSP, Java servlets.
- CO4: Make use of PHP and XML to develop interoperable web Applications.
- CO5: Construct cross platform web applications using Asynchronous Web requests and Web services.

### **TEXT BOOK:**

1. Deitel, Deitel & Nieto, 2012, *Internet and World Wide Web - How to Program*, 5<sup>th</sup> ed, Prentice Hall.

### **REFERENCES:**

1. Stephen Wynkoop & John Burke, 1999, *Running a Perfect Website*, 2<sup>nd</sup> ed, QUE.
2. Chris Bates, 2009, *Web Programming – Building Intranet Applications*, 3<sup>rd</sup> ed, Wiley Publications.
3. Jeffrey C & Jackson, 2011, *Web Technologies A Computer Science Perspective*, Pearson Education.
4. Gopalan, NP & Akilandeswari, J, 2011, *Web Technology*, Prentice Hall of India.
5. Uttam K.Roy, 2011, *Web Technologies*, Oxford University Press.

## CS1502 THEORY OF COMPUTATION AND COMPILER DESIGN

L	T	P	C
3	0	2	4

### OBJECTIVES:

To enable the students to

- Use finite automata in lexical analyzer design.
- Make use of various parsing techniques in designing a parser.
- Understand about syntax-directed translation and intermediate code generation.
- Learn about run-time environment and code generation.
- Implement code optimization techniques.

### UNIT I FORMAL LANGUAGE AND REGULAR EXPRESSIONS 11

Structure of a compiler – Definition: Languages, regular expressions, Finite Automata: DFA, NFA - Conversion of regular expression to NFA, NFA to DFA - Role of Lexical Analyzer – Input Buffering – Specification of Tokens - Recognition of Tokens - Applications of Finite Automata to lexical analysis, Lex tool.

### UNIT II CONTEXT FREE GRAMMARS AND PARSING 11

Context Free Grammars, Derivation, Parse trees, ambiguity, Role of Parser – Error Handling – Top down parsers - Bottom up parsers - Handle pruning, LR Parsers – SLR, CLR, LALR - Error Handling and Recovery in Syntax Analyzer - YACC tool.

### UNIT III SEMANTICS AND INTERMEDIATE CODE GENERATION 9

Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking.

### UNIT IV RUNTIME STORAGE AND CODE GENERATION 7

Storage Organization, Stack Allocation Space, Access to Non-local Data on the Stack, Heap Management - Issues in Code Generation - Design of a simple Code Generator.

## UNIT V CODE OPTIMIZATION

7

Principal Sources of Optimization – Peep-hole optimization - DAG- Optimization of Basic Blocks- Global Data Flow Analysis - Efficient Data Flow Algorithm.

**TOTAL: 45 PERIODS**

### LIST OF EXPERIMENTS

1. Implement a Lexical Analyzer using Lex Tool.
2. Implement an Arithmetic Calculator using LEX and YACC.
3. Generate three address code for a simple program.
4. Implement back-end of the compiler for which three address code is given as input and 8086 assembly language code is produced as output.
5. Implement simple code optimization techniques (Constant folding, Strength reduction and Algebraic transformation).

**PRACTICAL : 30 PERIODS**

### LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS

?

S. No.	Description of Equipment	Quantity Required
1.	Personal Computers (Intel Core i3, 250 GB, 4 GB RAM)	30
2.	Printer	1
3.	Software: C / C++ / Java or equivalent compiler Tools : Lex / YACC	Open source

**TOTAL : 75 PERIODS**

### OUTCOMES:

**Upon successful completion of the course, students will be able to**

- CO1: Develop lexical analyser for a sample language.
- CO2: Construct parser for a simple context free language.
- CO3: Explain the key features of syntax-directed translation and intermediate codes.
- CO4: Summarize the concepts of run-time environment and code generation.
- CO5: Experiment with different code optimization techniques.

**TEXT BOOK:**

1. Aho, Ullman & Ravi Sethi, 2009, *Compilers Principles, Techniques and Tools*, 2<sup>nd</sup> ed, Pearson Education.

**REFERENCE BOOKS:**

1. John E Hopcroft, Rajeev Motwani & Jeffrey D Ullman, 2007, *Introduction to Automata Theory Languages & Computation*, 3<sup>rd</sup> ed, Pearson Education.
2. Tremblay JP, Sorenson GP, 2010, *The Theory & Practice of Compiler writing*, 1<sup>st</sup> ed, BSP publication.
3. Appel W & Andrew GM, 2003, *Modern Compiler Implementation in C*, 1<sup>st</sup> ed, Cambridge University Press.
4. Louden, 2006, *Compiler Construction, Principles & Practice*, 1<sup>st</sup> ed, Thomson Press.

L	T	P	C
0	0	4	2

**OBJECTIVES:**

To enable the students to

- Be familiar with web page design using HTML5, CSS3 and JavaScript.
- Implement web applications using AngularJS framework.
- Gain knowledge about server side scripting languages.
- Build interoperable web applications using XML, PHP.
- Be familiar with the concepts of web service and its related technologies.

**LIST OF EXPERIMENTS:**

1. Create a website using HTML5 tables, lists, control elements, media elements, graphics elements, Geolocation, drag and drop effects, image maps.
2. Create web pages with the following.
  - a. Use different CSS selectors
  - b. External style sheets.
  - c. Embedded style sheets.
  - d. Inline style sheets.
3. Construct and Validate the Registration, user login, user profile and payment by credit card forms using JavaScript.
4. Develop interactive scientific calculator using AngularJS framework.
5. Develop programs in Java using Servlets:
  - a. To invoke servlets from HTML forms
  - b. To track sessions using hidden form fields, URL rewriting techniques
  - c. To track sessions for counting the number of visitors who visited the web page.
6. Create a three-tier web application using servlets for any of the following:
  - a. Online examination System.
  - b. Library management System.
  - c. Ticket reservation system, etc.

Assume that related information is available in a database which has been stored in a database server.

7. Develop simple dynamic web application along with database connectivity using JSP and cookies.

8. Create an XML document which contains a minimum of 10 books information. Do the following on the above created XML document
  - a. Validate the XML document using DTD / Schema.
  - b. Return the details of the particular book whose book ID is given as an input by taking the book information from the XML document using DOM parser.
  - c. Display all book details in the sorted order(ascending) of price using XSLT.
9. Construct an PHP applications for the following:
  - a. Validate the HTML forms using regular expressions
  - b. Send an email
  - c. Store form data into database
  - d. File uploading and downloading
10. Develop a web service for finding the feedback about a particular product.

**TOTAL: 60 PERIODS**

**OUTCOMES:**

**Upon successful completion of the course, students will be able to**

- CO1: Build dynamic web pages using HTML5, CSS3, JavaScript and its related frameworks.  
 CO2: Develop three-tier web applications using server side scripting languages.  
 CO3: Construct simple web applications using PHP.  
 CO4: Develop interoperable web applications using XML and its related technologies.  
 CO5: Build web services for solving real world applications.

**LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS**

Sl. No.	Description of Equipment	Quantity Required
1.	Personal Computers (Intel Core i3, 500 GB, 8 GB RAM)	30
2.	Printer	1
3.	Software: Netbeans IDE 8.2, WAMP/XAMPP, Apache Tomcat server.	30

L	T	P	C
0	0	4	2

**OBJECTIVES:**

To enable the students to

- Learn and use network commands.
- Understand socket programming.
- Familiarize with the implementation and analysis of various network protocols.
- Make use of simulation tools to analyze the performance of various network protocols.

**LIST OF EXPERIMENTS:**

1. Learn to use commands like tcpdump, netstat, ifconfig, nslookup and traceroute. Capture ping and traceroute PDUs using wireshark and examine.
2. Simulate bit stuffing concept
3. Simulate error correction mechanism (like CRC).
4. Simulation of Distance Vector/ Link State Routing algorithm.
5. Performance evaluation of Routing protocols using Simulation tool.
6. Implementation of subnetting concepts
7. Write a HTTP web client program to download a web page using TCP sockets.
8. Applications using TCP sockets like:
  - Echo client and echo server
  - Chat
  - File Transfer
9. Simulation of DNS using UDP sockets.
10. Write suitable codes for simulating ARP /RARP protocols.
11. Simulation of Congestion Control Algorithms using NS.
12. Performance Evaluation of TCP/UDP using Simulation tool.
13. Working with RMI using Java

**TOTAL: 60 PERIODS**

**OUTCOMES:**

**Upon completion of this course, the students will be able to:**

CO1: Make use of basic networking commands for capturing packets in live networks.

CO2: Implement bit stuffing and error correction algorithms in a client/server environment.

CO3: Analyse the performance of network routing protocols and transport layer services through simulation tools.

CO4: Simulate the application protocols using TCP and UDP.

CO5: Build an RMI server/client for a real time application.

**LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS**

<b>S. No.</b>	<b>DESCRIPTION OF EQUIPMENT</b>	<b>QUANTITY REQUIRED</b>
1.	Personal Computers ( Intel Core i3 processor)	30
2.	Printer	1
3.	Server (Intel Core i3, 4 GB RAM) (High Speed Processor)	1
4.	For capturing packets : Wireshark Simulation Tools Preferred : CISCO Packet Tracer Language preferred : C, Java, Python	30
5.	Complier : C, Java, Python	30

L	T	P	C
3	0	0	3

**OBJECTIVES:**

To enable the students to

- Understand the foundations of distributed systems.
- Learn issues related to clock Synchronization and the need for global state in distributed systems.
- Learn distributed mutual exclusion and deadlock detection algorithms
- Understand the significance of agreement, fault tolerance and recovery protocols in Distributed Systems.
- Learn the characteristics of peer-to-peer and distributed shared memory systems

**UNIT I INTRODUCTION****9**

Introduction: Definition –Relation to computer system components –Motivation –Relation to parallel systems – Message-passing systems versus shared memory systems –Primitives for distributed communication –Synchronous versus asynchronous executions –Design issues and challenges. A model of distributed computations: A distributed program –A model of distributed executions –Models of communication networks –Global state– Cuts – Past and future cones of an event –Models of process communications. Logical Time: A framework for a system of logical clocks –Scalar time –Vector time – Physical clock synchronization: NTP.

**UNIT II MESSAGE ORDERING AND SNAPSHOTS****9**

Message ordering and group communication: Message ordering paradigms –Asynchronous execution with synchronous communication –Synchronous program order on an asynchronous system –Group communication – Causal order (CO) - Total order. Global state and snapshot recording algorithms: Introduction –System model and definitions –Snapshot algorithms for FIFO channels

### **UNIT III      DISTRIBUTED MUTEX AND LOCKS**

**9**

Distributed mutual exclusion algorithms: Introduction – Preliminaries – Lamport’s algorithm – Ricart-Agrawala algorithm – Maekawa’s algorithm – Suzuki–Kasami’s broadcast algorithm. Deadlock detection in distributed systems: Introduction – System model – Preliminaries – Models of deadlocks – Knapp’s classification – Algorithms for the single resource model, the AND model and the OR model.

### **UNIT IV      RECOVERY AND CONSENSUS**

**9**

Checkpointing and rollback recovery: Introduction – Background and definitions – Issues in failure recovery – Checkpoint-based recovery – Log-based rollback recovery – Coordinated checkpointing algorithm – Algorithm for asynchronous checkpointing and recovery. Consensus and agreement algorithms: Problem definition – Overview of results – Agreement in a failure – free system – Agreement in synchronous systems with failures.

### **UNIT V      P2P & DISTRIBUTED SHARED MEMORY**

**9**

Peer-to-peer computing and overlay graphs: Introduction – Data indexing and overlays – Chord – Content addressable networks – Tapestry. Distributed shared memory: Abstraction and advantages – Memory consistency models – Shared memory Mutual Exclusion – Introduction to Cloud Computing.

**TOTAL : 45 PERIODS**

#### **OUTCOMES:**

**Upon successful completion of the course, students will be able to**

CO1: Explain the foundations and issues of distributed systems.

CO2: Outline the various synchronization issues and global state for distributed systems.

CO3: Explain the mutual exclusion and deadlock detection algorithms in distributed systems.

CO4: Summarize the agreement protocols and fault tolerance mechanisms in distributed systems.

CO5: Illustrate the features of peer-to-peer and distributed shared memory systems

**TEXT BOOKS:**

1. Ajay D Kshemkalyani & Mukesh Singhal, 2011, *Distributed computing: principles, algorithms, and systems*, Cambridge University Press.
2. George Coulouris, Jean Dollimore & Tim Kindberg, 2012, *Distributed Systems Concepts and Design*, 5<sup>th</sup> ed, Pearson Education.

**REFERENCE BOOKS:**

1. Pradeep K Sinha, 2000, *Distributed Operating Systems: Concepts and Design*, Prentice Hall of India.
2. Mukesh Singhal & Niranjan G Shivaratri, 1994, *Advanced concepts in operating systems*, McGraw-Hill, Inc.
3. Tanenbaum, AS, Van Steen, M, 2007, *Distributed Systems: Principles and Paradigms*, Pearson Education.
4. Liu, ML, 2004, *Distributed Computing, Principles and Applications*, Pearson Education.
5. Nancy A Lynch, 2003, *Distributed Algorithms*, Morgan Kaufman Publishers, USA.

L	T	P	C
3	0	0	3

**OBJECTIVES:**

To enable the students to

- Understand the fundamentals of IoT and Architecture
- Be familiar with various IoT-related protocols
- Build IoT Systems using Arduino and Raspberry Pi
- Summarize the basics of data analytics.
- Develop real time IoT applications.

**UNIT I INTRODUCTION TO IoT****9**

Evolution of Internet of Things - IoT Enabling Technologies - IoT Levels - IoT Architectures - IoT and M2M – Sensors types, principle, requirement and advantages – Classification of Sensors, Actuators, Smart Objects and Connecting Smart Objects

**UNIT II SENSORS FUNDAMENTALS****9**

Amplification Basics of Measurement – Classification of errors – Error analysis – Data Acquisition: Single channel and multi-channel data acquisition. Data logging – applications, Motion Sensors – Potentiometers, Accelerometer, GPS, Bluetooth, Ultrasonic Ranging, Strain Gauge, Load Cell, and Magnetic Sensors.

**UNIT III IoT PROTOCOLS****9**

IoT access technologies: Physical and MAC Layers, Topology – Application transport methods: Supervisory control and data acquisition – Application layer protocols: CoAP and MQTT. Design Methodology - Embedded Computing Logic - Microcontroller, System on Chips - IoT System Building Blocks.

**UNIT IV BUILDING IoT WITH ARDUINO, RASPBERRY PI & JETSON****9**

Arduino - Board Details, IDE Programming - Logical Design using Python, Raspberry Pi - Interfaces and Raspberry Pi with Python Programming, Introduction to Jetson controller and its applications

## UNIT V CASE STUDIES AND REAL-WORLD APPLICATIONS

9

IoT Cloud Storage Models & Communication APIs - Cloud for IoT – Smart Agriculture Power Utility Industry - Smart Grid - Smart and Connected Cities: Smart Lighting, Smart Parking, Smart Traffic Control and Commercial building automation.

**TOTAL : 45 PERIODS**

### OUTCOMES:

**Upon successful completion of the course, students will be able to**

CO1: Outline the IoT architecture and sensor fundamentals.

CO2: Summarize the data acquisition concepts and various sensors operation

CO3: Outline the various protocols used in IoT applications.

CO4: Build IoT Systems using Arduino and Raspberry PI.

CO5: Construct real time smart IoT Application using embedded system.

### TEXT BOOK:

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton & Jerome Henry, 2017, *IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things*, Cisco Press.

### REFERENCE BOOKS:

1. Arshdeep Bahga & Vijay Madisetti, 2015, *Internet of Things - A hands-on approach*, Universities Press.
2. Olivier Hersent, David Boswarthick & Omar Elloumi, 2012, *The Internet of Things - Key Applications and Protocols*, Wiley (for Unit 2).
3. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stamatis Karnouskos, Stefan Avesand & David Boyle, 2014, *From Machine-to-Machine to the Internet of Things - Introduction to a New Age of Intelligence*, Elsevier.
4. Dieter Uckelmann, Mark Harrison & Florian Michahelles, 2011, *Architecting the Internet of Things*, Springer.
5. Michael Margolis, Arduino Cookbook, 2011, *Recipes to Begin, Expand, and Enhance Your Projects*, 2<sup>nd</sup> ed, O'Reilly Media.

L	T	P	C
3	0	0	3

**OBJECTIVES:**

To enable the students to

- Understand the basic concepts of Machine Learning techniques.
- Learn regression algorithms and its applications.
- Learn classify algorithms and its applications
- Make use of ensemble learning techniques
- Learn unsupervised learning techniques and its applications.

**UNIT I FOUNDATIONS OF MACHINE LEARNING****9**

Types of Machine learning: Supervised learning- Unsupervised learning-Reinforcement Learning- Machine Learning Process-Terminologies: Weight Space, Curse of Dimensionality, Overfitting, Training, Testing, Validation Sets- Performance Measures: Confusion Matrix, Accuracy Metrics, Receiver Operator Characteristic (ROC) Curve, Measurement Precision- Model selection-No free lunch theorem- Bias-Variance Tradeoff

**UNIT II SUPERVISED LEARNING - REGRESSION AND TREE BASED MODELS****9**

Linear Regression - Multivariate Regression- Logistic Regression- Principal Component Regression- Decision Trees, Regression Trees

**UNIT III SUPERVISED LEARNING – CLASSIFICATION****9**

Linear Classification- Probability and Bayes learning - Naive Bayes - Bayesian Network - Perceptron - Perceptron Learning - Neural Networks – Back propagation- Support Vector Machines

**UNIT IV ENSEMBLE LEARNING****9**

Ensemble Methods - Bagging, Committee Machines and Stacking, Boosting - Gradient Boosting, Adaptive Boosting, Random Forests-Multi-class Classification

## UNIT V UNSUPERVISED LEARNING

9

Introduction to Clustering- Partitional Clustering - Hierarchical Clustering - Birch Algorithm- CURE Algorithm - Density based Clustering- Expectation Maximization

**TOTAL: 45 PERIODS**

### OUTCOMES:

**Upon successful completion of the course, students will be able to**

- CO1: Explain the fundamental concepts of machine learning.
- CO2: Apply suitable regression algorithms for an application.
- CO3: Apply suitable classification algorithms for an application
- CO4: Apply ensemble learning for the real world applications.
- CO5: Apply clustering algorithms for different types of applications.

### TEXT BOOKS:

1. Ethem Alpaydin, 2015, *Introduction to Machine Learning*, 3<sup>rd</sup> ed, Prentice Hall of India.
2. Stephen Marsland, 2015, *Machine Learning - An Algorithmic Perspective 2nd Edition*, CRC Press.

### REFERENCE BOOKS:

1. Christopher Bishop, 2006, *Pattern Recognition and Machine Learning*, Springer.
2. Kevin P. Murphy, 2012, *Machine Learning: A Probabilistic Perspective*, MIT Press.
3. Stephen Marsland, 2014, *Machine Learning –An Algorithmic Perspective*, 2<sup>nd</sup> ed, CRC Press.
4. Tom Mitchell, 2017, *Machine Learning*, McGraw-Hill.

**HS1521**

**PROFESSIONAL COMMUNICATION**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>2</b>	<b>1</b>

**OBJECTIVES:**

**This course aims to:**

- Enhance the Employability and Career Skills of students
- Orient the students towards grooming as a professional
- Make them Employable Graduates.
- Develop their confidence and help them attend interviews successfully

**UNIT I**

Introduction to Soft Skills – Hard skills & soft skills – employability and career Skills - Grooming as a professional with values - Time Management - General awareness of Current Affairs.

**UNIT II**

Self-Introduction-organizing the material – Introducing oneself to the audience – introducing the topic – answering questions with clarity and appropriate phrases – individual presentation practice - presenting the visuals effectively – 5 minute presentations

**UNIT III**

Introduction to Group Discussion - Participating in group discussions – understanding group dynamics – brainstorming the topic - questioning and clarifying – GD strategies- activities to improve GD skills

**UNIT IV**

Interview etiquette – dress code – body language – attending job interviews– telephone/skype interview - one to one interview & panel interview – FAQs related to job interviews

**UNIT V**

Recognizing differences between groups and teams - managing time-managing stress - networking professionally - respecting social protocols-understanding career management - developing a long-term career plan-making career changes - Error spotting.

**TOTAL: 30 PERIODS**

**OUTCOMES:**

**At the end of the course, learners will be able to:**

CO1: Make effective presentations

CO2: Participate confidently in Group Discussions.

CO3: Attend job interviews and be successful in them.

CO4: Develop adequate Soft Skills required for the workplace.

**REFERENCES:**

1. Jeff Butterfield, 2015, *Soft Skills for Everyone*, Cengage Learning: New Delhi.
2. Suresh Kumar, E, et al., 2015, *Communication for Professional Success*, Orient Blackswan: Hyderabad.
3. *Interact English Lab Manual for Undergraduate Students*, Orient BlackSwan: Hyderabad, 2016.
4. Raman, Meenakshi & Sangeeta Sharma, 2014, *Professional Communication*, Oxford University Press: Oxford.
5. Hariharan, S, et al., 2010, *Soft Skills*, MJP Publishers: Chennai.

**LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS**

<b>S. No.</b>	<b>DESCRIPTION OF EQUIPMENT</b>	<b>QUANTITY REQUIRED</b>
1.	Personal Computers (Intel Core i3, 250 GB, 4 GB RAM)	30
2.	Printer	1
3.	Software: Orell Techno Systems Digital Language Lab Software	30 (Licenses)

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>

**OBJECTIVES:**

To enable the students to

- Demonstrate the operation of different electronic devices and sensors.
- Select the most appropriate sensors for an IoT application.
- Build a small low cost embedded system using Arduino.
- Demonstrate the usage of Raspberry PI processor in developing IoT applications.
- Apply the concept of Internet of Things in real world scenario.

**LIST OF EXPERIMENTS**

**Internet of Things**

List of Experiments:

1. Working with Basic Analog and Digital Sensors
  - a. LED Display
  - b. Intensity Measurements (Dawn to Dusk)
  - c. Human Detection
2. Working with Advanced Analog and Digital Sensors
  - a. Flex Sensor
  - b. Wet Measurement-Soil Moisture Sensor
  - c. Sound Control
  - d. Load Monitoring
3. Implement the following experiments using Arduino like IDE
  - a. Temperature and Humidity measurement
  - b. Signal Variance - Potentiometer
  - c. Fire alarm indication using Buzzer
4. Write Program for monitoring sensor values in real time using Arduino.
  - a. IR Temperature sensor
  - b. Gas leakage detection
  - c. Smart Intrusion detection with SMS alert
5. Study the ESP8266 WIFI module and write program to transfer the data in the cloud.
  - a. Light Control Monitoring

- b. Soil Condition Monitoring
- c. Human detection – PIR Sensor
- 6. Various applications using Raspberry Pi
  - a. Stepper Motor
  - b. Face recognition
  - c. RFID
- 7. Experiments on Industrial IoT
  - a. Smart AC Controller System
  - b. Machine health monitoring
  - c. Energy Meter monitoring for theft detection.
- 8. Develop smart application using Jetson controller
- 9. Mini projects in IoT
  - a. Sensor Fabrication
  - b. AI Thermometer
  - c. Vehicle Density Calculation
  - d. Smart AI pot hole detector
  - e. Open ALPR license
  - f. Fruit Classifier
  - g. Autonomous mine detector
  - h. Water Quality Management
  - i. Defect identification stereo camera
  - j. Home automation
  - k. Smart health monitoring
  - l. Smart agriculture
  - m. Smart Pest Control using Drone
  - n. Field surveillance using Drone

**TOTAL: 60 PERIODS**

**COURSE OUTCOMES:**

**Upon successful completion of the course, students will be able to**

CO1: Construct simple applications using sensors.

CO2: Make use of various sensors to develop IoT applications.

CO3: Build IoT applications using Arduino.

CO4: Make use of Raspberry PI processor for developing IoT applications.

CO5: Develop real time smart IoT Applications

**LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS**

<b>S. No.</b>	<b>DESCRIPTION OF EQUIPMENT</b>	<b>QUANTITY REQUIRED</b>
1.	Personal Computers (Intel Core i3, 500 GB, 4 GB RAM)	30
2.	Printer	1
3.	Software: Arduino IDE, Third Party Cloud API like (Azure / Think speak), Python 3 interpreter	30
<b>Hardware list:</b>		
1.	Sensors and Actuator	60
2.	Arduino Boards	10
3.	Node MCU	10
4.	GSM/GPRS shields	10
5.	Raspberry PI 4	10
6.	Jetson GPU Board	10
7.	Robotic and Drone Kit	3

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>2</b>	<b>1</b>

**OBJECTIVES:**

To enable the students to

- Develop the skills in using recent machine learning software for solving practical problems.
- Interpret the type of machine learning algorithm required for a given problem
- Use suitable classification or regression algorithm for an application
- Use clustering algorithms for different types of applications

**LIST OF EXPERIMENTS****Implement the following problems using Python Packages:**

1. Regression Algorithms – Simple Linear, Multiple Linear and Logistics
2. Classification Algorithms for Tree based models– Decision Tree, Principal component regression
3. Linear Classification Algorithms – Naïve Bayes, Neural Networks, Support Vector Machine
4. Classification Algorithm for multiclass – Gradient boosting, Random forest
5. Clustering Algorithms – Partition clustering, Hierarchical clustering

**TOTAL: 30 PERIODS**

**OUTCOMES:**

Upon successful completion of the course, students will be able to

**CO1:** Solve the regression problems using statistical packages

**CO2:** Build the tree based models using suitable classification algorithms

**CO3:** Apply the supervised machine learning techniques for various classification problems

**CO4:** Make use of ensemble learning techniques to solve problems

**CO5:** Implement various clustering problems using unsupervised machine learning techniques

**LIST OF LAB EQUIPMENTS FOR A BATCH OF 30 STUDENTS**

<b>Sl.No.</b>	<b>DESCRIPTION OF EQUIPMENT</b>	<b>QUANTITY REQUIRED</b>
1	Personal Computers (Intel Core i3, 500GB, 4GB RAM)	30
2	Printer	1
3	Python3 interpreter for Windows/Linux	30 users

L	T	P	C
0	0	4	2

**OBJECTIVES:**

To enable the students to

- Understand the components and structure of mobile application development frameworks.
- Develop mobile applications using Graphical Primitives and Event Driven Concepts
- Familiarize with the file concepts and inbuilt/SQLite database for developing mobile applications
- Construct real time mobile applications to integrate cloud database and IoT

**LIST OF EXPERIMENTS:**

1. Develop an application that uses GUI components, Styles
2. Develop an application that uses Layout Managers and event listeners.
3. Develop an application to draw basic graphical primitives on the screen.
4. Create an Animation using Multithreading
5. Develop an application that makes use of databases. (SQLite)
6. Implement an application that writes data to the SD card.
7. Develop an application that makes use of Notification Manager
8. Develop a native application that uses GPS location information
9. Implement an application that creates an alert upon receiving a message
10. Develop a mobile application to send an email.
11. Develop a mobile application with regional language support
12. Develop a mobile application to record audio
13. Develop a mobile application for sensing temperature using IOT
14. Develop a mobile application to retrieve and process data from cloud storage
15. Develop a Mini Project

**PRACTICAL : 60 PERIODS**

### LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS

S. No.	DESCRIPTION OF EQUIPMENT	QUANTITY REQUIRED
1.	Personal Computers (8 core Apple M1 chip,8 GB)	30
2.	Printer	1
3.	Server (Intel Core i3, 4 GB RAM) (High Speed Processor)	1
4.	Software: Android Studio 4.2	30
5.	Android Emulator	30

#### OUTCOMES:

**Upon successful completion of the course, students will be able to**

CO1: Build simple mobile applications using GUI components.

CO2: Develop mobile applications using Graphical Primitives and Event Driven Concepts.

CO3: Make use of file concepts and inbuilt/SQLite database for developing mobile applications.

CO4: Utilize notification and location tracking concepts in mobile application development.

CO5: Construct real time mobile applications to integrate cloud database and IoT.

#### TEXT BOOK:

1. John Horton, 2018, *Android Programming for Beginners*, 2<sup>nd</sup> ed, Packt Publishing.

#### REFERENCE BOOK:

1. Pradeep Kothari, Kogent Learning Solutions Inc., 2014, *Android Application Development (With Kitkat Support)*, Black Book, 1<sup>st</sup> ed, Kindle Edition.

L	T	P	C
3	0	0	3

**PREREQUISITE:** CS1401 - Analysis of Algorithms

**OBJECTIVES:**

To enable the students to

- Describe the basic concepts of graphics systems.
- Familiarize with various 2D Modelling techniques.
- Understand the different 3D Modelling techniques.
- Gain knowledge about Illumination and different color models.
- Explore the various Animation and Graphics techniques.

**UNIT I INTRODUCTION**

**9**

Historical Evolution - Survey of computer graphics, Overview of graphics systems – Video display devices, Raster scan systems, Random scan systems, Graphics monitors and Workstations, Input devices, Hard copy Devices, Graphics Software; Output primitives – points and lines, line drawing algorithms, loading the frame buffer, line function; circle and ellipse generating algorithms; Pixel addressing and object geometry, filled area primitives.

**UNIT II TWO-DIMENSIONAL GRAPHICS**

**10**

Two dimensional geometric transformations – Matrix representations and homogeneous coordinates, composite transformations; Two-dimensional viewing – viewing pipeline, viewing coordinate reference frame; widow-to-viewport coordinate transformation, two-dimensional viewing functions; clipping operations – point, line, and polygon clipping algorithms

**UNIT III THREE-DIMENSIONAL GRAPHICS**

**10**

Three dimensional concepts; Three-dimensional object representations – Polygon surfaces- Polygon tables- Plane equations – Polygon meshes; Curved Lines and surfaces, Quadratic surfaces; Blobby objects; Spline representations – Bezier curves and surfaces -B-Spline curves and surfaces. Transformation and Viewing: Three dimensional geometric and modelling transformations – Translation, Rotation, Scaling, composite

transformations; Three-dimensional viewing – viewing pipeline, viewing coordinates, Projections, Clipping; Visible surface detection methods.

#### **UNIT IV ILLUMINATION AND COLOUR MODELS 7**

Light sources – basic illumination models – halftone patterns and dithering techniques; Properties of light – Standard primaries and chromaticity diagram; Intuitive colour concepts – RGB colour model – YIQ colour model – CMY colour model – HSV colour model – HLS colour model

#### **UNIT V ANIMATIONS AND REALISM 9**

Animation Graphics: Design of Animation sequences – animation function – raster animation – key frame systems – motion specification – morphing – Computer Graphics Realism: Tiling the plane – Recursively defined curves – Koch curves – C curves – Dragons – space filling curves – fractals – Grammar based models – fractals – turtle graphics – ray tracing

**TOTAL: 45 PERIODS**

#### **OUTCOMES:**

**Upon successful completion of the course, students will be able to**

CO1: Explain the basic concepts of display devices and primitives of Computer Graphics.

CO2: Apply two dimensional transformations and different types of clipping algorithms on 2D graphical objects.

CO3: Explain about different 3D Transformation techniques and Visible surface detection methods.

CO4: Interpret the idea of illumination and color models.

CO5: Illustrate the animation sequences and fractal concepts.

#### **TEXT BOOKS:**

1. John F Hughes, Andries Van Dam, Morgan Mc Guire, David F Sklar, James D Foley, Steven K Feiner & Kurt Akeley, 2013, *Computer Graphics: Principles and Practice*, 3<sup>rd</sup> ed, Addison Wesley Professional.
2. Donald Hearn & Pauline Baker, 2007, *Computer Graphics*, Prentice Hall, New Delhi.

## REFERENCE BOOKS:

1. Donald Hearn, Pauline Baker & Warren Carithers, 2010, *Computer Graphics with Open GL*, 4<sup>th</sup> ed, Pearson Education.
2. Jeffrey McConnell, 2006, *Computer Graphics Theory into Practice*, Jones and Bartlett Publishers.
3. Hill FS Jr, 1990, *Computer Graphics*, Maxwell Macmillan.
4. Peter Shirley, Michael Ashikhmin, Michael Gleicher, Stephen R Marschner, Erik Reinhard, Kelvin Sung & Peters AK, 2010, *Fundamental of Computer Graphics*, CRC Press.
5. William M Newman & Robert F Sproull, 1978, *Principles of Interactive Computer Graphics*, Mc Graw Hill.

**CS1532 DATA WAREHOUSING AND DATA MINING**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**PREREQUISITE:** CS1371 - Database Management Systems

**OBJECTIVES:**

To enable the students to

- Understand data warehouse concepts, architecture, business analysis and OLAP tools
- Understand data pre-processing and data visualization techniques
- Learn algorithms for finding hidden and interesting patterns in data
- Understand and apply various classification and clustering techniques
- Use Weka tool to experiment various data mining algorithms for various datasets.

**UNIT I DATA WAREHOUSING, BUSINESS ANALYSIS AND ON-LINE ANALYTICAL PROCESSING (OLAP) 9**

Basic Concepts - Data Warehousing Components – Building a Data Warehouse – Database Architectures for Parallel Processing – Parallel DBMS Vendors - Multidimensional Data Model – Data Warehouse Schemas for Decision Support Concept Hierarchies - Characteristics of OLAP Systems – Typical OLAP Operations, OLAP and OLTP.

**UNIT II DATA MINING – INTRODUCTION 9**

Introduction to Data Mining Systems – Knowledge Discovery Process – Data Mining Techniques – Issues – applications- Data Objects and attribute types, Statistical description of data, Data Pre-processing – Cleaning, Integration, Reduction, Transformation and discretization, Data Visualization, Data similarity and dissimilarity measures.

**UNIT III DATA MINING - FREQUENT PATTERN ANALYSIS 9**

Mining Frequent Patterns, Associations and Correlations – Mining Methods-Pattern Evaluation Method – Pattern Mining in Multilevel, Multi-Dimensional Space – Constraint Based Frequent Pattern Mining, Classification using Frequent Patterns

## **UNIT IV CLASSIFICATION AND CLUSTERING**

**9**

Decision Tree Induction - Bayesian Classification – Rule Based Classification –Classification by Back Propagation – Support Vector Machines — Lazy Learners – Model Evaluation and Selection- Techniques to improve Classification Accuracy. Clustering Techniques – Cluster analysis - Partitioning Methods - Hierarchical Methods – Density Based Methods - Grid Based Methods – Evaluation of clustering –Clustering high dimensional data-Clustering with constraints, Outlier analysis-outlier detection methods.

## **UNIT V WEKA TOOL**

**9**

Datasets – Introduction, Iris plants database, Breast cancer database, Auto imports database - Introduction to WEKA, The Explorer – Getting started, Exploring the explorer, Learning algorithms, Clustering algorithms, Association– Rule learners.

**TOTAL: 45 PERIODS**

### **OUTCOMES:**

**Upon successful completion of the course, students will be able to**

CO1: Develop a Data warehouse system for a given application.

CO2: Identify suitable pre-processing and visualization techniques for different types of datasets.

CO3: Apply frequent pattern and association rule mining techniques for data analysis.

CO4: Apply appropriate classification and clustering techniques for data analysis.

CO5: Experiment with Weka tool, various data mining algorithms

### **TEXT BOOKS:**

1. Alex Berson & Stephen J Smith, 2016, *Data Warehousing, Data Mining & OLAP*, 35<sup>th</sup> Reprint, Tata McGraw –Hill Edition.
2. Jiawei Han & Micheline Kamber, 2012, *Data Mining Concepts and Techniques*, 3<sup>rd</sup> ed, Elsevier.

## REFERENCE BOOKS:

1. Soman, KP, Shyam Diwakar & Ajay, V, 2006, *Insight into Data Mining Theory and Practice*, Eastern Economy Edition, Prentice Hall of India.
2. Ian H Witten & Eibe Frank, 2016, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann.

## CS1533 LINUX PROGRAMMING AND SHELL SCRIPTING

L	T	P	C
2	0	2	3

**PREREQUISITE:** CS1372 – System Programming and Operating Systems

### OBJECTIVES:

To enable the students to

- Understand the basic concepts of Linux.
- Familiarize fundamentals of Shell Scripts.
- Demonstrate the shell scripting operations.
- Explain Linux networking concepts.
- Familiarize advanced Linux concepts and configuration.

### UNIT I INTRODUCTION TO LINUX

6

A brief history of Linux, architecture of Linux, features of Linux, File System of the Linux, Linux structure, various flavours of Linux. Usage of Linux kernel. Basic and utility commands.

### UNIT II INTRODUCTION TO SHELL SCRIPTS

6

Different types of Editors (like ed, sed, gEdit, emacs, nano, vi, vim), Different modes of the vi editor, Different types of shell, Shell Scripting, write and execute a Shell script file, Scripting Language vs Programming Language, Process, Running jobs in the background and foreground, Scheduling Job (at, batch, cron, nice), Terminate Process Environment Variable, Local and Global Variables, Positional Parameter.

### UNIT III SHELL SCRIPT OPERATION AND PROGRAMMING

6

Batch Script vs Shell Script, Advantages and Disadvantages of Shell Script, Shell Arithmetic Basic Operators, Arrays, Shell Function, Decision Making Statement. - Debugging Shell Scripts, Shell Loops, Loop Control flow, Shell Script Programs.

### UNIT IV LINUX NETWORKING

6

Introduction to Networking in Linux, Network basics & tools, File transfer protocol in Linux, Network file system, Domain Naming Services, Setting Up Local Network Services, Dynamic hosting configuration Protocol & Network information Services.

## UNIT V CASE STUDY

6

Linux System- Basic Concepts; System Administration-Requirements for Linux System Administrator, setting up a LINUX Multifunction Server, Virtualization- Basic Concepts, Setting Up Xen, VMware on Linux Host and Adding Guest OS.

**THEORY : 30 PERIODS**

### LIST OF EXPERIMENTS:

1. Execution of various basic commands, file/directory, filters, pipe commands.
2. Simple shell script for basic arithmetic, logical calculations and various operations on string.
3. Shell scripts to check various attributes of files and directories.
4. Shell scripts to explore system variables such as PATH, HOME etc.
5. Shell scripts to check and list attributes of processes.
6. Execution of various system administrative commands.

**PRACTICAL: 30 PERIODS**

### LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS

S. No.	DESCRIPTION OF EQUIPMENT	QUANTITY REQUIRED
1.	Personal Computers (Intel Core i3, 250 GB, 4 GB RAM)	30
2.	Printer	1
3.	Server (Intel Core i3, 4 GB RAM) (High Speed Processor)	1
4.	Systems with Linux OS and C Compiler	30
5.	Putty software in each terminal to connect to server and use BASH SHELL for SHELL SCRIPTS.	30

**TOTAL : 60 PERIODS**

### OUTCOMES:

**Upon successful completion of the course, students will be able to**

- CO1: Summarize the basic concepts of Linux.  
CO2: Explain the basic concepts of shell script.  
CO3: Explain shell scripting operations.  
CO4: Explain Linux networking and its services.  
CO5: Illustrate the advanced concepts of Linux administration.

**TEXT BOOKS:**

1. Mokhtar Ebrahim & Andrew Mallett, 2018, *Mastering Linux Shell Scripting*, 2<sup>nd</sup> ed, Packt Publications.
2. Tom Adelstein & Bill Lubanovic, 2007, *Linux System Administration*, O'Reilly Media Inc.

**REFERENCE BOOKS:**

1. Ganesh Sanjiv Naik, 2018, *Learning Linux Shell Scripting*, 2<sup>nd</sup> ed, Packt Publications.
2. Richard Stevens, W, 2005, *Advanced Programming in the UNIX Environment*, 3<sup>rd</sup> ed, Pearson Education, New Delhi, India.
3. Robert Love, 2013, *Linux System Programming*, 2<sup>nd</sup> ed, O'Reilly.
4. William Stallings, 2011, *Operating Systems – Internals and Design Principles*, 7<sup>th</sup> ed, Prentice Hall.

L	T	P	C
3	0	0	3

**PREREQUISITE:** CS1372 – System Programming and Operating Systems

**OBJECTIVES:**

To enable the students to:

- Learn the fundamentals of Real Time Systems.
- Understand the software requirements of Real Time Systems.
- Understand intertask communication and memory management in Real Time Systems.
- Learn the concepts of real time databases.
- Understand the evaluation techniques used in Real Time Systems.

**UNIT I REAL TIME SYSTEM AND SCHEDULING**

**9**

Introduction– Structure of a Real Time System –Task classes – Performance Measures for Real Time Systems – Estimating Program Run Times – Issues in Real Time Computing – Task Assignment and Scheduling – Classical uniprocessor scheduling algorithms –Fault Tolerant Scheduling.

**UNIT II SOFTWARE REQUIREMENTS ENGINEERING**

**9**

Requirements engineering process – types of requirements – requirements specification for real time systems – Formal methods in software specification – structured Analysis and Design – object oriented analysis and design and unified modelling language – organizing the requirements document – organizing and writing documents – requirements validation and revision.

**UNIT III INTERTASK COMMUNICATION AND MEMORY MANAGEMENT**

**9**

Buffering data – Time relative Buffering- Ring Buffers – Mailboxes – Queues – Critical regions – Semaphores – other Synchronization mechanisms – deadlock – priority inversion – process stack management – run time ring buffer – maximum stack size – multiple stack arrangement – memory management in task control block - swapping – overlays – Block page management – replacement algorithms – memory locking – working sets – real time garbage collection – contiguous file systems

## **UNIT IV REAL TIME DATABASES**

**9**

Real time Databases – Basic Definition, Real time Vs General Purpose Databases, Main Memory Databases, Transaction priorities, Transaction Aborts, Concurrency control issues, Disk Scheduling Algorithms, Two– phase Approach to improve Predictability – Maintaining Serialization Consistency – Databases for Hard Real Time Systems.

## **UNIT V EVALUATION TECHNIQUES AND CLOCK SYNCHRONIZATION**

**9**

Reliability Evaluation Techniques – Obtaining parameter values, Reliability models for Hardware Redundancy–Software error models. Clock Synchronization–Clock, A Nonfault– Tolerant Synchronization Algorithm – Impact of faults – Fault Tolerant Synchronization in Hardware – Fault Tolerant Synchronization in software.

**TOTAL: 45 PERIODS**

### **OUTCOMES:**

**Upon successful completion of the course, students will be able to**

- CO1: Explain the fundamentals of Real Time Systems.
- CO2: Summarize the software requirements of Real Time Systems.
- CO3: Explain intertask communication and memory management in Real Time Systems.
- CO4: Outline the Real Time databases concepts.
- CO5: Summarize the evaluation techniques and synchronization.

### **TEXT BOOK:**

1. Krishna, CM & Kang G Shin, 1997, *Real-Time Systems*, McGraw-Hill International Editions.

### **REFERENCE BOOKS:**

1. Philip A Laplante, 2004, *Real Time System Design and Analysis*, Prentice Hall of India, 3rd Edition.
2. Rajib Mall, 2009, *Real-time systems: theory and practice*, Pearson Education.
3. Buhur RJA & Bailey DL, 1999, *An Introduction to Real-Time Systems*, Prentice Hall International.
4. Stuart Bennett, 1998, *Real Time Computer Control-An Introduction*, Prentice Hall of India.

5. Allen Burns & Andy Wellings, 2003, *Real Time Systems and Programming Languages*, Pearson Education.

## CS1535 SOFTWARE TESTING AND QUALITY ASSURANCE

L	T	P	C
3	0	0	3

**PREREQUISITE:** CS1402- Software Engineering with UML Design

### OBJECTIVES:

To enable the students to

- Understand the basics and goals of software testing
- Familiarize with various methods of unit and white box testing techniques.
- Understand integration, system and acceptance testing techniques.
- Gain knowledge about automation testing and the software quality standards for establishing quality environment
- Familiarize with the methods and evaluation procedures for improving the quality models

### UNIT I INTRODUCTION 9

Basics Concepts of Software Testing and Quality Assurance - Quality Revolution, Verification and Validation, Failure, Error, Fault, and Defect, Objectives of Testing, Testing Activities, Test Case Selection White-Box and Black ,test Planning and design, Test Tools and Automation, . Power of Test. Test Team Organization and Management-Test Groups, Software Quality Assurance Group, System Test Team Hierarchy, Team Building

### UNIT II UNIT AND WHITE BOX TESTING 9

Unit Testing - Control Flow Testing: Control Flow Graph - Paths in a Control Flow Graph - Path Selection Criteria - Generating Test Input - Examples of Test Data Selection - Containing Infeasible Paths Data Flow Testing: Data Flow Graph - Data Flow Testing Criteria - Comparison of Data Flow Test Selection Criteria - Feasible Paths and Test Selection Criteria - Comparison of Testing Techniques.

### UNIT III INTEGRATION, SYSTEM AND ACCEPTANCE TESTING 9

System Integration Techniques - Software and Hardware Integration - Test Plan for System Integration – Off the shelf Component Integration System Test: System Test Categories, Functional Testing: Functional Testing Concepts of Howden - Complexity of Applying Functional Testing - Pairwise Testing - Equivalence Class Partitioning - Boundary Value Analysis - Decision Tables - Random Testing - Error Guessing - Category Partition Acceptance Testing: Types -

Acceptance Criteria - Selection of Acceptance Criteria - Test Plan - Test Execution - Test Report  
- Acceptance Testing in eXtreme Programming.

**UNIT IV SOFTWARE QUALITY MODELS 9**

Software quality - People 's Quality Expectations, Frameworks and ISO-9126, McCall 's Quality Factors and Criteria – Relationship. Quality Metrics. Quality Characteristics ISO 9000:2000 Software Quality Standard. Maturity models- Test Process Improvement, Testing Maturity Model

**UNIT V EMERGING AREAS IN TESTING AND QUALITY THROUGH CONTINUOUS IMPROVEMENT PROCESS 9**

Software test automation, Role of Statistical Methods in Software Quality - Key to Quality management - Quality Assurance - Root Cause Analysis, modeling, technologies, standards and methodologies for defect prevention. Fault Tolerance and Failure Containment - Safety Assurance and Damage Control, Hazard analysis using fault-trees and event-trees. Comparing Quality Assurance Techniques and Activities. QA Monitoring and Measurement, Risk Identification for Quantifiable Quality Improvement. Case Study: FSM-Based Testing of Web-Based Applications

**TOTAL: 45 PERIODS**

**OUTCOMES:**

**Upon successful completion of the course, students will be able to**

- CO1: Identify the right type of software testing process for any given real-world problem.
- CO2: Make use of unit and white box software testing process.
- CO3: Explain the integration, system and acceptance testing techniques.
- CO4: Infer quality standards for developing quality software.
- CO5: Utilize the quality procedures through continuous improvement process.

**TEXT BOOKS:**

1. Kshirasagar Naik & Priyadarsh Tripathy, 2011, *Software Testing and Quality Assurance - Theory and Practice* 1<sup>st</sup> edition, John Wiley & Sons publication.
2. William E Lewis, 2011, *Software Testing and Continuous Quality Improvement*, 3<sup>rd</sup> ed, Auerbach Publications.

3. Srinivasan Desikan & Gopaldaswamy Ramesh, 2008 *Software testing: Principles and Practices*, 6<sup>th</sup> ed, Pearson Education.

**REFERENCE BOOKS:**

1. Alan C Gillies, 2011, *Software Quality Theory and Management*, 2<sup>nd</sup> ed, Cengage Learning.
2. Ron Patton, 2007, *Software testing*, 2<sup>nd</sup> ed, Pearson Education,
3. Elfriede Dustin, Jeff Rashka & John Paul, 1999, *Automated Software Testing: Introduction, Management and Performance 1<sup>st</sup> edition*, Addison-Wesley.
4. William E Perry, 2006, *Effective Methods for Software Testing*, 3<sup>rd</sup> ed, Wiley India.

**CS1536**

**AGILE DEVELOPMENT**

L	T	P	C
3	0	0	3

**PREREQUISITE:** CS1402- Software Engineering with UML Design

**OBJECTIVES:**

To enable the students to

- Explore the theoretical as well as practical understanding of agile software development practices
- Learn the various agile processes
- Explore agility and knowledge management.
- Understand the impact of agile processes in requirements engineering.
- Learn the various agile metrics.

**UNIT I      AGILE METHODOLOGY**

**9**

Theories for Agile Management – Agile Software Development – Traditional Model vs. Agile Model – Classification of Agile Methods – Agile Manifesto and Principles – Agile Project Management – Agile Team Interactions – Ethics in Agile Teams – Agility in Design, Testing – Agile Documentations – Agile Drivers, Capabilities and Values.

**UNIT II      AGILE PROCESSES**

**9**

Lean Production – SCRUM, Crystal, Feature Driven Development- Adaptive Software Development – Extreme Programming: Method Overview – Lifecycle – Work Products, Roles and Practices.

**UNIT III      AGILITY AND KNOWLEDGE MANAGEMENT**

**9**

Agile Information Systems – Agile Decision Making – Earl\_S Schools of KM – Institutional Knowledge Evolution Cycle – Development, Acquisition, Refinement, Distribution, Deployment , Leveraging – KM in Software Engineering – Managing Software Knowledge – Challenges of Migrating to Agile Methodologies – Agile Knowledge Sharing – Role of Story-Cards – Story-Card Maturity Model (SMM).

## **UNIT IV      AGILITY AND REQUIREMENTS   ENGINEERING**

**9**

Impact of Agile Processes in RE–Current Agile Practices – Variance – Overview of RE Using Agile – Managing Unstable Requirements – Requirements Elicitation – Agile Requirements Abstraction Model – Requirements Management in Agile Environment, Agile Requirements Prioritization – Agile Requirements Modeling and Generation – Concurrency in Agile Requirements Generation.

## **UNIT V      AGILITY AND QUALITY ASSURANCE**

**9**

Agile Product Development – Agile Metrics – Feature Driven Development (FDD) – Financial and Production Metrics in FDD – Agile Approach to Quality Assurance – Test Driven Development – Agile Approach in Global Software Development.

**TOTAL: 45 PERIODS**

### **OUTCOMES:**

**Upon successful completion of the course, students will be able to**

- CO1: Demonstrate agile software development processes.
- CO2: Explain the various agile processes.
- CO3: Explain agility and knowledge management.
- CO4: Illustrate agile processes in Requirements engineering.
- CO5: Demonstrate the various agile metrics for quality assurance.

### **TEXT BOOKS:**

1. David J Anderson & Eli Schragenheim, 2003, *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results 1<sup>st</sup> edition*, Prentice Hall.
2. Hazza & Dubinsky, 2009, *Agile Software Engineering, Series: Undergraduate Topics in Computer Science*, Springer.

### **REFERENCE BOOKS:**

1. Craig Larman, 2004, *Agile and Iterative Development: A Managers Guide*, Addison-Wesley.
2. Kevin C Desouza, 2007, *Agile Information Systems: Conceptualization, Construction, and Management*, Butterworth-Heinemann.

**CS1537**

**C# AND .NET FUNDAMENTALS**

L	T	P	C
2	0	2	3

**PREREQUISITE:** GE1171-Fundamentals of Computing and Programming

**OBJECTIVES:**

To enable the students to

- Understand the .NET architecture and basic features of C#.
- Be familiar with the object oriented programming concepts of C#.
- Gain knowledge about advanced features in C#.
- Build simple application using database connectivity with C#.
- Develop web application using ASP.NET.

**UNIT I INTRODUCTION TO .NET AND C# FUNDAMENTALS**

**6**

Introduction to .NET Architecture - CLS, CTS, BCL, Overview of C# - Data types, Literals and Variables, Boxing and Un Boxing – Operators – Control statements - Enumerations -Parameter passing mechanisms - Arrays, Strings, StringBuilder.

**UNIT II OBJECT ORIENTED CONCEPTS IN C#**

**6**

Classes and Objects, Inheritance: Virtual methods and Overriding–Abstract Class, Sealed Class. Interfaces – Generic classes and methods - Collection

**UNIT III ADVANCED C# FEATURES**

**6**

Exception handling - Delegates and Events - lambda expressions, Multithreading -Introduction to assemblies – Properties - Reflection, and Attributes

**UNIT IV ADO.NET AND DATABASE CONNECTIVITY**

**6**

Connected Model: Data provider classes, Sql Connection, Sql Command and Sql Data Reader. Disconnected Model: Data Adapter, Data Set, Data Table. Parameters and Stored procedures.

ASP.NET based web application –Web Forms Code behind model –Autopostback – concept of state - validating controls - master and content pages – Web services, Introduction to ASP.NET MVC.

**THEORY : 30 PERIODS**

**LIST OF EXPERIMENTS:**

1. Implement a simple C# program using jagged array concept
2. Develop programs with the usage of C# specific keywords (ref, in, out)
3. Implement object oriented concepts using class, objects and constructors
4. Implement multiple inheritance using C#
5. Implement a C# program for the following concepts.
  - a. Generic class
  - b. Exception handling
6. Implement a C# program for the following concepts.
  - a. Delegates
  - b. Event handling
7. Develop a simple database application using ADO.NET connected Model
8. Develop a simple database application using ADO.NET disconnected Model
9. Develop a web application using ASP.NET
10. Create a web service to implement the basic calculator services

**PRACTICAL : 30 PERIODS**

**LABORATORY REQUIREMENTS FOR A BATCH OF 30 STUDENTS:**

S. No.	DESCRIPTION OF EQUIPMENT	QUANTITY REQUIRED
1.	Personal Computers (Intel Core i3, 250 GB, 4GB RAM)	30
2.	Printer	1
3.	Software: Microsoft Visual Studio 2019, SQL Server or equivalent	

**TOTAL: 60 PERIODS**

**OUTCOMES:**

**Upon successful completion of the course, students will be able to**

CO1: Explain the basic concepts of C# and .NET framework.

CO2: Develop C# programs with object oriented programming concepts.

CO3: Build simple applications with exception handling and threading concepts.

CO4: Construct three tier applications using ADO.NET.

CO5: Build simple web application using ASP.NET.

**TEXT BOOKS:**

1. Herbert Schildt, 2017, *C# 4.0: The Complete Reference*, McGraw Hill Education, Indian Edition, Reprint.
2. Andrew Troelsen, Philip Japikse, 2015, *C# 6.0 and the .NET 4.6 Framework*, Apress, 7<sup>th</sup> ed.

**REFERENCE BOOKS:**

1. Mark Michaelis, Eric Lippert, 2015, *Essential C# 6.0*, Pearson.
2. Daniel Solis, 2012, *Illustrated C#*, Apress, 2<sup>nd</sup> ed.
3. Joseph Albahari, Ben Albahari, 2010, *C# 4.0 in a Nutshell: The Definitive Reference*, O'Reilly.
4. Ian Gariffiths, 2012, *Programming C# 5.0*, 4<sup>th</sup> ed, O'Reilly.
5. John Sharp, 2013, *Microsoft Visual C# 2013 Step by Step Developer*, Pearson Education.
6. Thamarai Selvi, S, Murugesan, R, 2012, *A textbook on C#*, Pearson.

L	T	P	C
2	0	2	3

**PREREQUISITES:**

- MA1473 - Probability and Statistics
- CS1532 - Data Warehousing and Data Mining

**OBJECTIVES:**

To enable the students to

- Learn programming using R Software
- Understand the data structures in R
- Familiarize with programming constructs in R
- Be able to manipulate data using R Packages
- Learn how to communicate results using R

**UNIT I INTRODUCTION****6**

Introduction: Overview of R: Essentials of the R Language, Installing R, Running R, Introduction to functions and Important R Data structures, Declarations, Recycling, Common vector operations, Vectorized Operations, NA and NULL Values, Filtering, Testing Vector Equality, Matrices and Arrays: Creating Matrices, General Matrix Operations, Applying functions to Matrix Rows and Columns, Adding and Deleting Matrix Rows and Columns, Avoiding Unintended Dimensional Reduction, Higher Dimensional Arrays

**UNIT II PROGRAMMING BASICS****6**

DATA STRUCTURES : Lists, Data Frames, Factors and Tables, PROGRAMMING STRUCTURES: Control Statements, Arithmetic and Boolean Operators and Values, Default values for Arguments, Return Values, Functions are objects, Environment and Scope issues, Recursion, Replacement Functions, Tools for composing Function Code, Writing own binary operations, Anonymous Functions, INPUT/OUTPUT : Accessing the Keyboard and Monitor, Reading and Writing Files, Accessing the internet.

### **UNIT III DATA MANIPULATION - I**

**6**

R Packages - Data Visualization with ggplot2, Data transformation with dplyr, Exploratory Data Analysis: Introduction, Variation, Missing values, Covariation, Patterns and models, ggplot2 calls, Tibbles with tibble: Creating tibbles, Tibbles Versus Data Frame, Interacting with Older Code, Data Import with readr: Parsing a Vector, Parsing a file, Writing to a File, Types of Data, Tidy data with tidyr: Tidy Data, Spreading and Gathering, Separating and Pull, Missing Values, Case Study, Nontidy Data. Relational Data with dplyr: Keys, Mutating Joins, Filtering Joins, Join Problems, Set Operations, Strings: Basics, Matching patterns with Regular Expressions, Tools, Types of Pattern, Uses of Regular Expressions, Stringi, Factors with forcats, Dates and Times with lubridate

### **UNIT IV DATA MANIPULATION - II**

**6**

Pipes with magrittr, Iteration with purr: Model basics with modelr: Simple model, Visualizing models, Formulas and Model Families, Missing values, Model Building, Many models with purr and broom: Introduction, gapminder, List-Columns, Creating List-Columns, Simplifying List-Columns. Making Tidy Data with broom

### **UNIT V COMMUNICATING RESULTS**

**6**

R Markdown: Basics, Text Formatting with Markdown, Code Chunks, Troubleshooting, YAML Header, Graphics for Communication with ggplot2: Label, Annotations, Scales, Zooming, Themes, Saving Plots R Markdown Formats: Output Options, Documents, Notebooks, Presentations, Dashboards, Interactivity, Websites, R Markdown Workflow

**THEORY : 30 PERIODS**

#### **LIST OF EXPERIMENTS:**

1. R Program to add two vectors
2. R program to find sum, mean and product of a vector
3. R program to find the factorial of a number
4. R program to print Fibonacci series
5. R program to make a simple calculator
6. R program to find sum of natural numbers using recursion
7. Data visualization with R Packages
8. Data manipulation with R packages

## 9. Preparing Documents with R Mark Down

**PRACTICAL: 30 PERIODS**

### **LABORATORY REQUIREMENTS FOR A BATCH OF 30 STUDENTS**

<b>S. No</b>	<b>DESCRIPTION OF EQUIPMENT</b>	<b>QUANTITY REQUIRED</b>
1	Personal Computers (Intel Core i3, 250 GB, 1 GB RAM) or Higher configuration	30
2	Printer	1
3	Software R-4.0.5 or latest version (free, open source)	

**TOTAL : 60 PERIODS**

### **OUTCOMES:**

**Upon successful completion of the course, students will be able to**

CO1: Solve simple problems with R programs.

CO2: Apply appropriate data structure and control structure to solve the given problem

CO3: Apply R for data visualization.

CO4: Make use of Packages in R for Data Manipulation.

CO5: Make use of R Markdown to communicate results

### **TEXT BOOKS:**

1. Hadley Wickham & Garrett Golemund, 2017, *R for Data Science*, O'Reilly Medi Inc.
2. Roger D Peng, 2015, *R Programming for Data Science*, Lean Publishing.

### **REFERENCE BOOKS:**

1. Norman Matloff, 2011, *The Art of R Programming: A tour of Statistical Software Design*, no starch press.
2. Garrett Golemund, 2014, *Hands-On Programming with R*, O'Reilly.
3. Nina Zumel & John Mount, 2014, *Practical Data Science with R*, Manning Publications.

**AD1503**

**OBJECT ORIENTED PROGRAMMING USING JAVA**

L	T	P	C
2	0	2	3

**PREREQUISITES:**

- CS1201 – Object Oriented Programming using Python
- CS1211 - Object Oriented Programming laboratory using Python

**OBJECTIVES:**

To enable the students to

- Understand the basic Object Oriented Programming concepts and Java fundamentals
- Explore the principles of inheritance and interfaces
- Learn the concepts of exception handling and use I/O streams
- Develop a java application using Threads and generics classes
- Familiarize simple Graphical User Interface applications

**UNIT I INTRODUCTION TO OOPS AND JAVA FUNDAMENTALS 6**

Object Oriented Programming Concepts – Fundamental Programming Structures in Java – Data Types, Variables, Operators, Control Flow, Arrays, Defining classes in Java – constructors, methods and Fields - Packages.

**UNIT II INHERITANCE AND INTERFACES 6**

Inheritance – Super classes & Sub classes, types of Inheritance, Abstract classes and methods – Final Classes and methods – Interfaces – Inner classes – String.

**UNIT III EXCEPTION HANDLING AND I/O 6**

Exceptions – Exception hierarchy, Throwing & Catching exceptions, Built-in Exceptions and Creating own exceptions, Input / Output Streams Basics – Byte streams and Character streams – Reading and Writing Console – Reading and Writing Files

**UNIT IV MULTITHREADING AND GENERIC PROGRAMMING 6**

Differences between multi-threading and multitasking, thread life cycle, creating threads, synchronizing threads, Inter-thread communication, Generic Programming – Generic classes – generic methods, Type inference with var and datetime and Collection API.

Introduction to Swing – layout management, Swing Components, Working with 2D shapes – Using color, fonts, and images – Basics of event handling – AWT event hierarchy, adapter classes & mouse events –Solid pattern in Java and JDBC

**THEORY : 30 PERIODS**

**LIST OF EXPERIMENTS:**

1. Develop a Java application to generate Electricity bill.
2. Develop a java application to implement the User Defined Packages.
3. Write a program to perform string operations using ArrayList.
4. Develop a java application with Employee class with Emp\_name, Emp\_id, Address, Mail\_id, Mobile\_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary
5. Design a Java interface for ADT Stack/Queue. Implement this interface using array. Provide necessary exception handling in both the implementations
6. Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
7. Write a Java program that reads the file information and display it in console by Byte/Character Stream Classes.
8. Write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
9. Write a java program to find the maximum value from the given type of elements using a generic function
10. Design a calculator using event-driven programming paradigm of Java

11. Write a java program to connect MySQL/Oracle using JDBC

**PRACTICAL : 30 PERIODS**

**LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS**

<b>S. No.</b>	<b>DESCRIPTION OF EQUIPMENT</b>	<b>QUANTITY REQUIRED</b>
1.	Personal Computers (Intel Core i3, 250GB, 2 GB RAM)	30
2.	Printer	1
3.	Software: Java (JDK Latest Version), NetBeans IDE / Eclipse IDE or equivalent	30 users

**TOTAL : 60 PERIODS**

**OUTCOMES:**

**Upon successful completion of the course, students will be able to**

CO1: Explain the basic Object Oriented Programming concepts and Java Fundamentals.

CO2: Demonstrate Java applications with Inheritance and Interfaces.

CO3: Infer the use of Exceptions and I/O Streams in Java applications.

CO4: Build applications using Threads and Generic Classes.

CO5: Construct interactive Java applications using swings.

**TEXTBOOKS:**

1. Herbert Schildt, 2019, *Java The complete reference*, 11<sup>th</sup> ed, McGraw Hill Education.
2. Cay S Horstmann & Gary Cornell, 2013, *Core Java Volume – I Fundamentals*, 9<sup>th</sup> ed, Prentice Hall.

**REFERENCE BOOKS:**

1. Paul Deitel & Harvey Deitel, 2015, *Java SE 8 for programmers*, 3<sup>rd</sup> ed, Pearson.
2. Steven Holzner, 2011, *Java 2 Black book*, Dreamtech press.
3. Timothy Budd, 2000, *Understanding Object-oriented programming with Java*, Updated Edition, Pearson Education.



Entertainment networks, Collaborative multimedia support, Real-time distributed multimedia networks, Hypertext, Hypermedia.

## **UNIT V SMART MULTIMEDIA APPLICATIONS**

**9**

Blogs & Vlogs, Social Media, Visualization, Virtual, Augmented and Mixed Reality, Interactive E-learning, HCI and UX design, Games and Animation, Real-Time video conferencing.

**TOTAL: 45 PERIODS**

### **OUTCOMES:**

**Upon successful completion of the course, students will be able to**

CO1: Outline the characteristics and formats of multimedia elements.

CO2: Make use of Multimedia Hardware and Software for Editing and Authoring multimedia applications.

CO3: Classify Compression algorithms for appropriate multimedia applications.

CO4: Demonstrate effective strategies to deliver Quality-of-Experience in networked Multimedia applications.

CO5: Extend multimedia applications in prominent domains.

### **TEXTBOOKS:**

1. Ze-Nian Li, Mark S Drew & Jiangchuan Liu, 2014, *Fundamentals of Multimedia*, 2<sup>nd</sup> ed, Springer Nature.
2. Prabhat K Andleigh & Kiran Thakrar, 2015, *Multimedia Systems Design*, 1<sup>st</sup> ed, Pearson Education India.
3. Ralf Steinmetz & KlaraNahrstedt, 2002, *Multimedia computing, communications, and applications*, Pearson India, Pearson.

### **REFERENCE BOOKS:**

1. Fred Halsall, 2002, *Multimedia Communications: Applications, Networks, Protocols and Standards*, Pearson Education.
2. Khalid Sayood, 2012, *Introduction to Data Compression*, 4<sup>th</sup> ed, Morgan Kauffman.
3. Rao, KR, Zoran S Bojkovic & Bojan M Bakmaz, 2017, *Wireless Multimedia Communication systems: Design, Analysis and Implementation*, CRC press.

4. Subrahmanian, VS, 2008, *Principles of Multimedia Database Systems*, Elsevier / Morgan Kauffmann.



## UNIT V DATA ANALYTICS USING HIVE

4

Introduction – Data Types and File Formats – Databases in Hive – HiveQL: Data Definition – Data Manipulation – Queries – Views – Indexes – Schema Design

**THEORY : 30 PERIODS**

### LIST OF EXPERIMENTS

- (i) Perform setting up and Installing Hadoop in its two operating modes:
  - Pseudo distributed,
  - Fully distributed.(ii) Use web based tools to monitor the Hadoop setup.
- (i) Implement the following file management tasks in Hadoop:
  - Adding files and directories
  - Retrieving files
  - Deleting files(ii) Benchmark and stress test an Apache Hadoop cluster
- Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.
  - Find the number of occurrence of each word appearing in the input file(s)
  - Performing a MapReduce Job for word search count (look for specific keywords in a file)
- Install and Run Pig then write Pig Latin scripts to sort, group, join, project, and filter your data.
- Write a Pig Latin scripts for finding TF-IDF value for book dataset (A corpus of eBooks available at: Project Gutenberg)
- Install and Run Hive then use Hive to create, alter, and drop databases, tables, views, functions, and indexes

**PRACTICAL : 30 PERIODS**

### LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS

Sl. No.	DESCRIPTION OF EQUIPMENT (MINIMUM HARDWARE AND SOFTWARE REQUIREMENTS)	QUANTITY REQUIRED
1.	Hadoop Framework	30

2.	Map Reduce Frame work	30
3.	Big data tools(Pig Latin, Hive..)	30
4.	PCs with minimum (8 GB RAM)	30

**TOTAL : 60 PERIODS**

### **OUTCOMES:**

**Upon successful completion of the course, students will be able to**

CO1: Explain the fundamental concepts of Big Data and its tools and techniques.

CO2: Apply the concepts of MapReduce framework.

CO3: Infer appropriate NoSQL database techniques for storing and processing large volumes of structure and unstructured data.

CO4: Write script using Pig latin.

CO5: Perform various HiveQL queries.

### **TEXT BOOK:**

1. Tom White, 2015, *Hadoop: The Definitive Guide*, 4th ed, O'Reilly Publications.

### **REFERENCE BOOKS:**

1. Judith Hurwitz, Alan Nugent, Fern Halper & Marcia Kaufman, 2014, *Big Data*, Wiley Publications.
2. Mark Van Rijmenam, 2014, *Think Bigger: Developing a Successful Big Data Strategy for Your Business*, 1<sup>st</sup> ed, Amazon.
3. Hurwitz JS, Nugent A, Halper F, Kaufman M, 2013, *Big data for dummies*, John Wiley & Sons.
4. Alan Gates & Daniel Dai, 2016, *Pig: Data flow Scripting with Hadoop*, O'Reilly Media.
5. Jason Rutherglen, Dean Wampler & Edward Capriolo, 2012, *Programming Hive*, 1<sup>st</sup> ed, O'Reilly Media.
6. Dayong Du, 2015, *Apache Hive Essentials*, Packet Publishing.
7. Hanish Bansal & Saurabh Chauhan, 2016, *Apache Hive Cookbook*, Packt publishing.

**AD1534**

**ETHICAL HACKING**

L	T	P	C
3	0	0	3

**PREREQUISITE:** IT1571 - Computer Networks

**OBJECTIVES**

To enable the students to

- 
- Understand the information security threats and counter measures
- Gain the knowledge to perform security auditing and testing
- Explain the various issues related to ethical hacking
- Familiarize with various ethical hacking techniques in web applications
- Experiment with the predominant Security Management tools

**UNIT I      ETHICAL HACKING OVERVIEW      9**

Introduction to Hacking – Importance of Security – Elements of Security – Phases of an Attack – Types of Hacker Attacks – Hacktivism – Vulnerability Research – Introduction to Footprinting – Information Gathering Methodology – Footprinting Tools – WHOIS Tools – DNS Information Tools – Locating the Network Range – Meta Search Engines.

**UNIT II      SCANNING AND ENUMERATION      9**

Introduction to Scanning – Objectives – Scanning Methodology – Tools – Introduction to Enumeration – Enumeration Techniques – Enumeration Procedure – Tools

**UNIT III      SYSTEM HACKING      9**

Introduction – Cracking Passwords – Password Cracking Websites – Password Guessing – Password Cracking Tools – Password Cracking Counter measures – Escalating Privileges – Executing Applications – Keyloggers and Spyware.

**UNIT IV      ETHICAL HACKING IN WEB      9**

Social Engineering, Session Hijacking, Hacking Web servers, Hacking Web Applications SQL Injection, Hacking Wireless Networks, Hacking Mobile Platforms.

## UNIT V SECURITY MANAGEMENT TOOLS

9

Manage security assessments with Metasploit, Honey pots,IDS- SolarWinds, Security Event Manager (SEM) Sagan, Kismet

**TOTAL : 45 PERIODS**

### OUTCOMES

**Upon successful completion of the course, students will be able to**

CO1: Outline the information security threats and countermeasures.

CO2: Illustrate the security auditing and testing process.

CO3: Interpret various issues related to ethical hacking.

CO4: Demonstrate real-world hacking techniques in web applications.

CO5: Make use of Security Management Tools.

### TEXT BOOKS:

1. EC-Council, 2016, *Ethical Hacking and Countermeasures: Attack Phases*, 2nd edition, Cengage Learning, Nelson Education.
2. Shimonski, R., 2016, *CEH v9: Certified Ethical Hacker Version 9 Study Guide*, John Wiley & Sons.

### REFERENCE BOOKS:

1. Rajat Khare, 2006, *Network Security and Ethical Hacking*, Luniver Press.
2. Ramachandran V, Back Track, 2011, *Wireless Penetration Testing Beginner's Guide*, 3<sup>rd</sup> ed, Packt Publishing.
3. Thomas Mathew, 2003, *Ethical Hacking*, OSB publishers.
4. Michael T Simpson, 2010, *Hands-on Ethical Hacking & Network Defense*, Course Technology.

L	T	P	C
2	0	2	3

**OBJECTIVES**

To enable the students to

- Understand the fundamental concepts of game design and development
- Expose concept of 2D and 3D Computer Graphics principles and algorithms for game design
- Familiarize with the processes, mechanics, issues in game design
- Understand the architecture of game engines
- Develop and implement simple games using standard APIs

**UNIT I INTRODUCTION****6**

Elements of Game Play – Artificial Intelligence – Getting Input from the Player - Sprite Programming – Sprite Animation - Multithreading – Importance of Game Design – Game Loop.

**UNIT II GRAPHICS FOR GAME PROGRAMMING****6**

Coordinate Systems – Ray Tracing, Modeling in Game Production – Vertex Processing, Rasterization – Fragment Processing and Output Merging – Illumination and Shaders, Parametric Curves and Surfaces.

**UNIT III GAME DESIGN PRINCIPLES****6**

Character Development – Story Telling – Narration – Game Balancing – Core mechanics – Principles of level design – Genres of Games – Collision Detection – Game Logic – Game AI, Path Finding – Case study : Tetris.

**UNIT IV GAMING ENGINE DESIGN****6**

Renderers – Software Rendering – Hardware Rendering and Controller Based Animation – Spatial Sorting – Level of Detail – Collision Detection – Standard Objects and Physics – Case study: The Sims

## UNIT V GAME DEVELOPMENT

6

Developing 2D and 3D Interactive Games Using OpenGL – DirectX – Isometric and Tile Based Games – Puzzle Games – Single Player Games – Multi -Player Games – Case study: Mine craft.

**THEORY : 30 PERIODS**

### LIST OF EXPERIMENTS:

1. Design sprites with necessary animations for a 2D sidescroller game.
2. Add related physics components for a 2D sidescroller game.
3. Develop the game design documents for a 2D sidescroller game.
4. Develop simple 2D puzzle game.
5. Develop simple 3D interactive game.

**PRACTICAL : 30 PERIODS**

### LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS

S. No.	DESCRIPTION OF EQUIPMENT	QUANTITY REQUIRED
1.	Personal Computers (Intel Core i5/i7, 500 GB, 8 GB RAM)	30
2.	Printer	1
3.	Software: Unity game engine	30

**TOTAL: 60 PERIODS**

### OUTCOMES:

**Upon successful completion of the course, students will be able to**

CO1: Understand the essential elements of Game Design and Game Play.

CO2: Demonstrate the concepts of Graphics for Game Design and Development.

CO3: Illustrate the importance of Game Design Principles and process.

CO4: Outline the essential components of game engine for designing and developing games.

CO5: Develop interactive 2D/3D Games

### TEXT BOOKS:

1. David H Eberly, 2010, *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*, 2<sup>nd</sup> ed, Morgan Kaufmann.
2. Jung Hyun Han, 2011, *3D Graphics for Game Programming*, 1<sup>st</sup> ed, Chapman and Hall/CRC.

## REFERENCE BOOKS:

1. Jonathan S Harbour, 2009, *Beginning Game Programming*, Course Technology, 3<sup>rd</sup> ed, PTR.
2. Ernest Adams and Andrew Rollings, 2014, *Fundamentals of Game Design*, 3<sup>rd</sup> ed, Pearson Education.
3. Scott Rogers, 2010, *Level Up: The Guide to Great Video Game Design*, 1<sup>st</sup> ed, Wiley.
4. Jim Thompson, Barna by Berbank-Green & Nic Cusworth, 2008, *Game Design: Principles, Practice, and Techniques - The Ultimate Guide for the Aspiring Game Designer*, 1<sup>st</sup> ed, Wiley.

L	T	P	C
3	0	0	3

**PREREQUISITE:** CS1372 – System Programming and Operating Systems

**OBJECTIVES:**

To enable the students to

- Understand the fundamental concepts of parallelization
- Learn the challenges in parallel and multi-threaded programming
- Gain knowledge about the use of MPI in distributed-memory programming
- Understand the use of OpenMP in shared-memory programming
- Compare MPI, OpenMP and Pthread APIs

**UNIT I INTRODUCTION 9**

Need for parallel computing – Von Neumann Architecture –Processes, multitasking, and threads  
–Parallel hardware –Parallel software –Input and Output –Performance –Parallel Program design  
–Writing and Running Parallel Programs

**UNIT II PARALLEL PROGRAM CHALLENGES 9**

Performance - Scalability Synchronization and data sharing - Data Races - Synchronization Primitives – Deadlocks - Livelocks - Communication between threads and variables - Storing thread private data

**UNIT III DISTRIBUTED-MEMORY PROGRAMMING WITH MPI 9**

Introduction - The trapezoidal rule –Input and Output – Collective Communication – MPI derived datatypes – Performance Evaluation – Parallel sorting algorithm.

**UNIT IV SHARED-MEMORY PROGRAMMING WITH OPENMP 9**

Introduction - The trapezoidal rule – Scope of variables – Reduction Clause – The parallel for directive – Loops – Scheduling loops –Producers and Consumers – Caches, Cache Coherence, and False sharing – Thread-safety

**UNIT V PARALLEL PROGRAM DEVELOPMENT 9**

Two n-Body solvers – Tree Search – Comparison and selection of API

**TOTAL : 45 PERIODS**

**OUTCOMES:**

**Upon successful completion of the course, students will be able to**

- CO1: Explain the fundamentals of multi core processors.
- CO2: Explain the performance issues in a parallel program.
- CO3: Develop parallelized programs using MPI.
- CO4: Develop parallelized programs using OpenMP.
- CO5: Compare serial and parallel programming implementations.

**TEXT BOOKS:**

1. Peter S Pacheco, 2011, *An Introduction to Parallel Programming*, Morgan-Kaufman/Elsevier.
2. Darryl Gove, 2011, *Multicore Application Programming for Windows, Linux, and Oracle Solaris*, Pearson (Unit II).

**REFERENCE BOOKS:**

1. Michael J Quinn, 2003, *Parallel programming in C with MPI and OpenMP*, Tata McGrawHill.
2. Shameem Akhter & Jason Roberts, 2006, *Multi-core Programming*, Intel Press.

L	T	P	C
3	0	0	3

**PREREQUISITE:** CS1402- Software Engineering with UML Design, CS1356 Agile Development

**OBJECTIVES:**

To enable the students to

- Understand the principles of verification and validation.
- Familiarize with different verification and validation techniques.
- Understand various stages of testing.
- Gain knowledge about the use of tools for verification and validation.
- Familiarize with the benefits of using metrics for verification and validation.

**UNIT I INTRODUCTION****9**

Principles of verification and validation – software architecture frameworks – model driven architecture – UML – systems modeling language – verification, validation and accreditation.

**UNIT II METHODS OF SOFTWARE VERIFICATION****9**

Verification and validation life cycle – traceability analysis – interface analysis – design and code verification – test analysis - Reviews – inspections - walkthroughs – audits – tracing – formal proofs – Model based verification and validation - Program verification techniques – formal methods of software verification – clean room methods – Defect analysis and prevention

**UNIT III TESTING****9**

Stages of Testing: Test Planning – Test design – Test case definition – Test procedure – Test reporting – Unit testing: white box, black box and performance testing – system testing: Function, performance, interface, operations, resource, security, portability, reliability, maintainability, safety, regression and stress testing – integration testing – acceptance testing: capability, constraint testing - structured testing – structured integration testing – Alpha and Beta testing

## **UNIT IV TOOLS FOR SOFTWARE VERIFICATION**

**9**

Tools for verification and validation: static analyser – configuration management tools – reverse engineering tools – tracing tools – tools for formal analysis – tools for testing – test case generators – test harnesses – debuggers – coverage analysers – performance analysers – test management tools

## **UNIT V ADVANCED APPROACHES**

**9**

Automatic approach for verification and validation – validating UML behavioral diagrams – probabilistic model checking of activity diagrams in SysML – metrics for verification and validation

**TOTAL: 45 PERIODS**

### **OUTCOMES:**

**Upon successful completion of the course, students will be able to**

- CO1: Identify the different techniques for verification and validation.
- CO2: Make use of traceability analysis tools on sample requirements.
- CO3: Explain the coverage analysers in terms of functionality or features used.
- CO4: Design system test cases.
- CO5: Infer test case generators and test management tools.

### **TEXT BOOKS:**

1. Avner Engel, 2011, *Verification, Validation & Testing of Engineered Systems*, Wiley series in systems Engineering and Management.
2. Mourad Debbabi, Hassaine, F, Jarrya, Y, Soeanu, A & Alawneh, L, 2010, *Verification and Validation in Systems Engineering*, Springer.

### **REFERENCE BOOKS:**

1. Marcus S Fisher, 2007, *Software Verification and Validation: An Engineering and Scientific Approach*, Springer.
2. ESA Board for Software Standardisation and Control (BSSC), 1995, *Guide to software verification and Validation*, European Space Agency ESA PSS-05-10 Issue 1 Revision 1.

**CS1635 AUGMENTED REALITY AND VIRTUAL REALITY**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**PREREQUISITE:** CS1531 – Computer Graphics

**OBJECTIVES:**

To enable the students to

- Understand the basic concepts and framework of virtual reality
- Be familiar with the relevance of virtual reality concept, virtual reality methodology, and virtual reality environment
- Design virtual reality on the web and mobile to demonstrate engineering applications
- Understand augmented reality methodology
- Demonstrate the basic functionalities of augmented reality

**UNIT I INTRODUCTION 9**

The three I's of virtual reality-commercial Virtual Reality technology and the five classic components of a Virtual Reality system - Input Devices: (Trackers, Navigation, and Gesture Interfaces): Three-dimensional position trackers, navigation and manipulation-interfaces and gesture interfaces-Output Devices: Graphics displays-sound displays & haptic feedback.

**UNIT II VIRTUAL REALITY DEVELOPMENT PROCESS AND CONTENT CREATION CONSIDERATIONS 9**

Geometric modelling - kinematics modelling- physical modelling - behaviour modelling - model Management, Methodology and terminology-user performance studies- Virtual Reality health and safety issues-Usability of virtual reality system- cyber sickness -side effects of exposures to virtual reality environment

**UNIT III VIRTUAL REALITY ON THE WEB & MOBILE 9**

JS-pros and cons-building blocks (WebVR, WebGL, Three.js, device orientation events) - frameworks (A-frame, React VR)-Google Virtual Reality for Android-Scripts, mobile device configuration, building to android-cameras and interaction-teleporting-spatial audio-Assessing human parameters-device development and drivers-Design Haptics

## **UNIT IV INTRODUCTION TO AUGMENTED REALITY**

**9**

Taxonomy, technology and features of augmented reality, difference between AR and VR, Challenges with AR, AR systems and functionality, Augmented reality methods, visualization techniques for augmented reality, wireless displays in educational augmented reality applications, mobile projection interfaces, marker-less tracking for augmented reality, enhancing interactivity in AR environments, evaluating AR systems

## **UNIT V AUGMENTED REALITY DEVELOPEMENT PROCESS**

**9**

Connecting Context and Content in AR System, Various types of contents. Associating content to context, Content Rendering in AR, POSE estimation, Obtaining RST (Rotation Scaling Translation) matrix, rendering 2D content - Images and Video, rendering 3D content - 3D models -AR Application Areas- Case study – Unity 3D and Vuforia – Deploying AR on mobiles.

**TOTAL: 45 PERIODS**

### **OUTCOMES:**

**Upon successful completion of the course, students will be able to**

- CO1: Demonstrate the basic concepts and framework of virtual reality.
- CO2: Interpret the relevance of this virtual reality concept, virtual reality methodology, and virtual reality environment.
- CO3: Develop web and mobile applications using virtual reality.
- CO4: Outline the augmented reality methodology.
- CO5: Demonstrate the basic functionalities using augmented reality.

### **TEXT BOOKS:**

1. Burdea & Philippe Coiffet, 2008, *Virtual Reality Technology*, 2<sup>nd</sup> ed, Gregory, John Wiley & Sons, Inc.
2. Alan B Craig, 2013, *Understanding Augmented Reality, Concepts and Applications*, Morgan Kaufmann.

### **REFERENCE BOOKS:**

1. Dieter Schmalstieg & Tobias Hollerer, 2016, *Augmented Reality: Principles and Practice (Usability)*, Pearson Education (US), Addison-Wesley Educational Publishers Inc, New Jersey, United States.

2. Jason Jerald, 2015, *The VR Book: Human-Centred Design for Virtual Reality*, Association for Computing Machinery and Morgan & Claypool, New York, NY, USA.
3. Steve Aukstakalnis, 2016, *Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR (Usability)*, 1<sup>st</sup> ed, Addison-Wesley Professional.
4. Tony Parisi, 2015, *Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile*, 1<sup>st</sup> ed, O'Reilly Media.
5. Tony Parisi, 2014, *Programming 3D Applications with HTML5 and WebGL: 3D Animation and Visualization for Web Pages*, 1<sup>st</sup> ed, O'Reilly Media.
6. Jos Dirksen, 2015, *Learning Three.js: The JavaScript 3D Library for WebGL*, 2<sup>nd</sup> ed, Packt Publishing.
7. Dieter Schmalstieg & Tobias Hollerer, 2016, *Augmented Reality: Principles & Practice*.
8. Cawood, 2008, *Augmented Reality: A Practical Guide*.

L	T	P	C
2	0	2	3

**PREREQUISITE:** CS1501 - Internet Programming

**OBJECTIVES:**

To enable the students to

- Familiarize with the front-end design of web pages using HTML5, CSS3 and Bootstrap4.
- Implement interactive web application design using JavaScript and jQuery.
- Build simple web applications using AngularJS framework.
- Understand the basic concepts of Node.js & Express.js Framework.
- Familiarize with the different back-end design using MySQL, MongoDB.

**UNIT I INTRODUCTION TO FRONTEND DESIGN 6**

Introduction to full stack development – Front-end web UI frameworks fundamentals – Basics of HTML5 & CSS3 – Responsive web design - Overview of Bootstrap4 – Containers – Grid system – Bootstrap4 CSS – Bootstrap4 Layout components – Themes – simple web application design.

**UNIT II JAVASCRIPT AND JQUERY 6**

An Introduction to JavaScript – JavaScript Objects – DOM model – functions – Event handling – jQuery overview – Selectors - Events – jQuery traversing– jQuery Effects - jQuery HTML- jQuery and AJAX

**UNIT III ANGULARJS FRAMEWORK 6**

Overview of JavaScript frameworks: MEAN stack frameworks - Introduction to Angular JS - Binding and Expression – Directives – Controllers – Filters – Modules – Services – Scopes – Tables & forms – AngularJS DOM – Animations – Simple application

**UNIT IV EXPRESS.JS AND NODE.JS FRAMEWORK 6**

Introduction to Node.js – Node.js Architecture – NPM (Node Packaging Manager) -Installing NPM module - Creating and locating modules - Creating a simple HTTP server - Overview of Express.js framework – Simple Express application – Routing – MVC in Express – middleware – templates – Error handling – Debugging – using process managers.

**UNIT V      MYSQL, MONGODB WITH NODEJS FRAMEWORK****6**

Introduction to MySQL with Node.js– Basic DDL & DML Operations, NoSQL Database -Basic CRUD Operations – Indexing –Aggregation -Data Modelling concepts - Connecting MongoDB using Mongoose

**THEORY : 30 PERIODS****LIST OF EXPERIMENTS:**

1. Build responsive website to display information about your department details using HTML5, CSS3, Bootstrap4. Use Bootstrap4 features like grids, jumbotron, carousel, etc.
2. Implement simple random quote generator using JavaScript. When the user clicks on New Quote button, it will generate new quote and display it to the user randomly. There is a collection of pre-defined quotes collected and store in constant. Whenever the new quote is displayed to the user, background colour should also be changed.
3. Create password strength checker application using jQuery.
4. Design simple scientific calculator using AngularJS framework
5. Develop to-do-list application using AngularJS.
6. Develop a BMI calculator application using node.js framework.
7. Build simple server application using node.js.
8. Build simple library management system using node.js and MySQL/MongoDB.

**PRACTICAL : 30 PERIODS****LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS**

<b>Sl. No.</b>	<b>DESCRIPTION OF EQUIPMENT</b>	<b>QUANTITY REQUIRED</b>
1.	Personal Computers (Intel Core i3, 500 GB, 8 GB RAM)	30
2.	Printer	1
3.	Node.js, MongoDB, Angular Framework, React Framework, MySQL	Open source

**TOTAL: 60 PERIODS****OUTCOMES:**

**Upon successful completion of the course, students will be able to**

CO1: Build the responsive web pages using front-end UI tools like HTML5, CSS3 and Bootstrap4.

CO2: Build dynamic web pages using JavaScript and jQuery.

CO3: Make use of AngularJS framework for developing simple web applications.

CO4: Construct web applications using node.js framework.

CO5: Make use of backend tools to build the real time web applications.

**TEXT BOOKS:**

1. Brad Dayley & Brendan Dayley, 2018, *Node.js, MongoDB and Angular Web Development*, 2<sup>nd</sup> ed, Pearson Education.
2. Deitel, Deitel & Nieto, 2011, *Internet and World Wide Web - How to Program*, 5<sup>th</sup> ed, Prentice Hall.

**REFERENCE BOOKS:**

1. Chris Northwood, *The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer*, Apress publisher.
2. Adam Bretz & Colin J Ihrig, *Full Stack JavaScript Development With MEAN*, Sitepoint publisher.
3. Vasan Subramanian, *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node*, Apress publisher.

**CS1637 MOBILE ADHOC NETWORKS**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**PREREQUISITE:** IT1571 - Computer Networks

**OBJECTIVES:**

To enable the students to

- Understand the various features of ad-hoc wireless networks and their applications.
- Gain knowledge about functioning of different access protocols and standards of ad-hoc networks.
- Familiarize the issues and classification in routing protocols for ad-hoc networks.
- Learn the need for security and secure routing protocol for ad-hoc networks.
- Know the challenges and role of cross layer design in enhancing the network performance.

**UNIT I INTRODUCTION TO AD-HOC NETWORKS & MANET 9**

Ad-hoc networks – definition, characteristics features, applications. Ad-hoc mobility models: indoor and outdoor models. Mobile Ad-hoc Networks (MANET) – architecture & applications.

**UNIT II MEDIUM ACCESS PROTOCOLS FOR MANET 9**

MAC Protocols: Design issues, goals and classification. Contention based protocols – with reservation, scheduling algorithms, protocols using directional antennas. IEEE standards: 802.11a, 802.11b, 802.11g, 802.15. HIPERLAN.

**UNIT III NETWORK PROTOCOLS FOR MANET 9**

Routing Protocols: Design issues, goals and classification. Proactive vs Reactive routing, unicast routing algorithms, Multicast routing algorithms, hybrid routing algorithm, energy aware routing algorithm, hierarchical routing, QoS aware routing.

**UNIT IV END – END DELIVERY AND SECURITY 9**

Transport Layer: Issues in designing – Transport layer classification, ad-hoc transport protocols. Security issues in ad-hoc networks: issues and challenges, network security attacks, secure routing protocols.

## UNIT V CROSS LAYER DESIGN

9

Cross layer Design: Need for cross layer design, cross layer optimization, parameter optimization techniques, cross layer cautionary perspective. Integration of ad-hoc with Mobile IP networks.

**TOTAL : 45 PERIODS**

### OUTCOMES:

**Upon successful completion of the course, students will be able to**

- CO1: Interpret the characteristics and mobility model for ad-hoc networks.
- CO2: Infer about the MAC protocols and different standards used in this layer.
- CO3: Summarize the different types of routing protocols for mobile ad-hoc networks.
- CO4: Compare ad-hoc transport protocols and security issues in ad-hoc networks.
- CO5: Illustrate cross layer design and ad-hoc with mobile IP networks.

### TEXT BOOK:

1. Siva Ram Murthy, C & Manoj, BS, 2007, *Ad hoc Wireless Networks Architecture and Protocols*, 2<sup>nd</sup> ed, Pearson Edition.

### REFERENCE BOOKS:

1. Charles E Perkins, 2000, *Ad hoc Networking*, Addison – Wesley.
2. Stefano Basagni, Marco Conti, Silvia Giordano & Ivan Stojmenovic, *Mobile ad-hoc networking*, Wiley-IEEE press.
3. Mohammad Ilyas, 2002, *The handbook of ad-hoc wireless networks*, CRC press.
4. Camp, T, Boleng, J & Davies, V, *A Survey of Mobility Models for Ad-hoc Network*, Wireless Commun, and Mobile Comp, Special Issue on Mobile Ad-hoc Networking Research, Trends and Applications, Vol. 2, no. 5, 2002, pp. 483 – 502.



analytics: Overview of channel analytics - Overview of video analytics - Tracking links with Bit.ly - Google analytics

**UNIT IV DATA COLLECTION AND VISUALIZATION 9**

Processing and Visualizing Data - Influence Maximization - Link Prediction - Collective Classification - Applications in Advertising and Game Analytics - Collecting and analyzing social media data using python - visualization and exploration of data using python

**UNIT V TEXT AND SENTIMENT ANALYTICS 9**

Introduction to Text Analytics - Nature of Text - Culture - Concordance - Content Analysis - Frequency Analysis: Co-Occurrence & Ngrams - Introduction to Sentiment Analysis - Learning Prior Polarity - Sentiment Prediction - Emotion Annotation and Prediction

**TOTAL: 45 PERIODS**

**OUTCOMES:**

**Upon successful completion of the course, students will be able to**

- CO1: Infer knowledge about basic concepts of social media analytics.
- CO2: Demonstrate the usage of web analytics tools.
- CO3: Apply different tools for studying and processing social media data.
- CO4: Illustrate how social media data is visualized.
- CO5: Explain about the usage of text and sentiment analytics in studying social media data.

**TEXT BOOKS:**

1. Jim Sterne, 2010, *Social Media Metrics: How to Measure and Optimize Your Marketing Investment* - Wiley publishers.
2. Marshall Sponder, 2010, *Social Media Analytics*, McGraw Hill.

**REFERENCE BOOKS:**

1. Matthew Ganis & Avinash Kohirkar, *Social Media Analytics: Techniques and Insights for Extracting Business Value Out of Social Media*, Pearson Education.
2. Oliver Blanchard, *Social Media ROI: Managing and Measuring Social Media Efforts in Your Organization*, Que Publishing.
3. Tracy L Tuten & Michael R Solomon, *Social Media Marketing*, Sage publishers.

**CS1639**

**SOFTWARE DESIGN**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**PREREQUISITE:** CS1402- Software Engineering with UML Design  
CS1356 Agile Development  
CS1633 Software Verification and Validation.

**OBJECTIVES:**

To enable the students to

- Understand the need for the pattern in software development
- Familiarize with basic functionalities of the design pattern
- Understand the creational and structural design patterns.
- Gain knowledge about behaviour of the design pattern
- Familiarize with the principles of concurrent pattern

**UNIT I INTRODUCTION**

**9**

Introduction to pattern – Pattern Categories– Relationship between Patterns- Patterns and software architecture Layers Architectural Pattern- Design Pattern- Design Patterns in Smalltalk MVC – Describing Design patterns – Catalog of Design Patterns- Organizing the Catalog – Solving Design Problems using Design patterns –Design Pattern Selection – Design Pattern Usage

**UNIT II FUNDAMENTAL DESIGN PATTERNS**

**9**

Whole Part – Master Slave –Command Processor – View Handler – Forward Receiver – Client Dispatcher Server.

**UNIT III CREATIONAL AND STRUCTURAL DESIGN PATTERNS**

**9**

Abstract Factory - Factory Method – Prototype - Singleton – Adapter Pattern – Bridge - Composite - Decorator – Façade – Proxy

**UNIT IV BEHAVIORAL DESIGN PATTERNS**

**9**

Chain of Responsibility – Command – Iterator - Mediator – Observer – Strategy – Visitor

## UNIT V INTRODUCTION TO SIMULINK AND ITS APPLICATIONS

9

Single Threaded Execution – Lock Object-Guarded Suspension – Balking – Scheduler -  
Read/Write Lock - An introduction to Anti-Pattern

**TOTAL: 45 PERIODS**

### OUTCOMES:

**Upon successful completion of the course, students will be able to**

CO1: Identify the appropriate pattern for the software development.

CO2: Illustrate the issues in the design pattern.

CO3: Infer the structure of the design pattern.

CO4: Utilize the behaviour of the design pattern.

CO5: Make use of the concurrent design pattern.

### TEXT BOOKS:

1. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, 2004, *Design patterns: Elements of Reusable Object Oriented Software*, Addison-Wesley, reprint.
2. Frank Bachmann, Regine Meunier & Hans Rohnert, 2008, *Pattern Oriented Software Architecture*, Volume 1, reprint.
3. Mark Grand, 2002, *Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML*, volume 1, 2<sup>nd</sup> ed, Wiley.

### REFERENCE BOOKS:

1. William J Brown et al, 1998, *Anti-Patterns: Refactoring Software, Architectures and Projects in Crisis*, John Wiley.
2. Praseed Pai & Shine Xavier, 2017, ".NET Design Patterns", Packt publication.
3. Christopher G Lasate, 2007, *Design Patterns*, Wordware Publishing Inc.

L	T	P	C
3	0	0	3

**OBJECTIVES:**

To enable the students to:

- Read and write simple Python programs.
- Develop Python programs with conditionals and loops.
- Define Python functions and call them.
- Use Python data structures – lists, tuples, dictionaries.
- Do input/output with files in Python.

**UNIT I DATA, EXPRESSIONS, STATEMENTS****9**

Python interpreter and interactive mode; values and types: int, float, boolean, string, and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; modules and functions, function definition and use, flow of execution, parameters and arguments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.

**UNIT II CONTROL FLOW****9**

Conditionals: Boolean values and operators, conditional (if), alternative (if-else), chained conditional (if-elif-else); Iteration: state, while, for, break, continue, pass; Illustrative programs: square root, gcd, exponentiation

**UNIT III FUNCTIONS****9**

Fruitful functions: return values, parameters, local and global scope, function composition, recursion; Strings: string slices, immutability, string functions and methods, string module; Lists as arrays. Illustrative programs: sum an array of numbers, linear search, binary search.

**UNIT IV LISTS, TUPLES, DICTIONARIES****9**

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension; Illustrative programs: selection sort, insertion sort, merge sort.

## UNIT V FILES, MODULES, PACKAGES

9

Files and exception: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file.

**TOTAL: 45 PERIODS**

### OUTCOMES:

**Upon completion of the course, students will be able to**

CO1: Interpret the basic syntaxes & concepts of python to code simple programs

CO2: Develop simple python programs using control structures

CO3: Demonstrate the purpose of functions in coding python programs

CO4: Compare the usage of different data structures in python programming

CO5: Illustrate various file operations in python programming

### TEXT BOOK:

1. Reema Thareja, 2019, *Problem Solving and Programming with Python*, 2<sup>nd</sup> ed, Oxford University Press.

### REFERENCE BOOKS:

1. Allen B Downey, 2016, *Think Python: How to Think Like a Computer Scientist*, 2<sup>nd</sup> ed, Updated for Python 3, Shroff/O'Reilly Publishers. (<http://greenteapress.com/wp/think-python/>)
2. Guido van Rossum & Fred L Drake Jr, 2011, *An Introduction to Python – Revised and updated for Python 3.2*, Network Theory Ltd.