

**SCHEME & SYLLABUS**

**OF**

**V & VI SEMESTERS**

**B.E.**

**COMPUTER SCIENCE AND ENGINEERING**

**2020-21**

# **V - SEMESTER COURSES**

**SCHEME OF TEACHING AND EXAMINATION : 2020-21 (BATCH : 2018-19)**

<b>V Semester</b>												
Sl. No.	Course	Course Code	Course Title	Teaching Dept.	Teaching hrs/week			Examination				Credits
					Theory lecture	Tutorial	Practical/ Drawing	Duration in hrs.	SEE Marks	CIE Marks	Total Marks	
1	HSS	HSS06	Management & Entrepreneurship	HSS	3	--	--	3	50	50	100	<b>3.0</b>
2	PC	5RCS01	Database Management Systems	CSE	4	--	--	3	50	50	100	<b>4.0</b>
3	PC	5RCS02	Data Communication	CSE	4	--	--	3	50	50	100	<b>4.0</b>
4	PC	5RCS03	Software Engineering	CSE	4	--	--	3	50	50	100	<b>4.0</b>
5	PCL	5RCSL01	Java programming Laboratory	CSE	--	1	2	3	50	50	100	<b>1.5</b>
6	PCL	5RCSL02	Database Management Systems Laboratory	CSE	--	--	3	3	50	50	100	<b>1.5</b>
7	NCMC	HSS08	Soft Skills	T & P	36 Hours/ Semester			Continuous Evaluation	--	100	100	<b>0.0</b>
8	Project		Mini Project– Phase I	CSE	--	--	2	--	--	--	--	<b>0.0</b>
9	PE	RCSEXX	Professional Elective -1	CSE	3	--	--	3	50	50	100	<b>3.0</b>
10	OE	OEXX	Open Elective – 1	OD	3	--	--	3	50	50	100	<b>3.0</b>
<b>Total</b>					<b>21</b>	<b>01</b>	<b>07</b>		<b>400</b>	<b>500</b>	<b>900</b>	<b>24.0</b>

**Mini project:** Batch formation, Guide allotment and Identification of topic for Mini-project has to be done during the 5<sup>th</sup> Semester.

Lateral entry students will study Mathematics IV – "Probability and its Applications for IT (4RMAT4)" for 4 credits in addition to the above courses.

**OD:** Offering Department    **NCMC :** Non Credit Mandatory Course

**SCHEME OF TEACHING AND EXAMINATION : 2020-21 (BATCH : 2018-19)**

<b>VI Semester</b>												
Sl. No.	Course	Course Code	Course Title	Teaching Dept.	Teaching hrs/week			Examination				Credits
					Theory lecture	Tutorial	Practical/ Drawing	Duration in hrs.	SEE Marks	CIE Marks	Total Marks	
1	HSS	HSS07A	Software Project Management	CSE	3	--	--	3	50	50	100	<b>3.0</b>
2	PC	6RCS01	Computer Networks	CSE	4	--	--	3	50	50	100	<b>4.0</b>
3	PC	6RCS02	System Software & Compiler Design	CSE	3	---	3	3	50	50	100	<b>4.5</b>
4	PCL	6RCSL01	Computer Networks Laboratory	CSE	--	--	3	3	50	50	100	<b>1.5</b>
5	PCL	6RCSL02	Mobile Application Development Laboratory	CSE	--	1	3	3	50	50	100	<b>2.0</b>
6	Project	6RCSP	Mini Project– Phase II	CSE	To be completed during the 6 <sup>th</sup> Semester, One day per Week should be earmarked for Mini project			3	50	50	100	<b>2.0</b>
7	VAC	ARAS	Aptitude Related Analytical Skills	CSE	36 Hrs. / Semester			1.5	50	50	100	<b>1.0</b>
8	PE	RCSEXX	Professional Elective – 2	CSE	3	--	--	3	50	50	100	<b>3.0</b>
9	OE	OEXX	Open Elective – 2	HSS	3	--	--	3	50	50	100	<b>3.0</b>
<b>Total</b>					<b>16</b>	<b>01</b>	<b>09</b>		<b>450</b>	<b>450</b>	<b>900</b>	<b>24.0</b>

**VAC:** Value Added Course

## DATABASE MANAGEMENT SYSTEMS

---

Contact Hours/Week	: 4 (Lecture)	Credits	: 4.0
Total Lecture Hours	: 52	CIE Marks	: 50
Total Tutorial Hours	: --	SEE Marks	: 50
Course Code	: 5RCS01		

---

### Course objectives:

1. Define a Database, characteristics and functions of Database Management System and distinguish between a Traditional File System and a Database System
2. Describe the Entity–Relationship (ER) modeling and model the real world database systems using Entity Relationship Diagrams (ERD) from the requirements specification
3. Apply the Relational Data Model, its Constraints and the Relational Database Schemas
4. Formulate queries in Relational Algebra & SQL
5. Apply normalization techniques to normalize a database
6. Illustrate how a DBMS enforces recovery from failure and concurrency control

### Course outcomes:

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

- CO1** : **Describe** the fundamentals of database technologies, Design an ER diagram and transform it to a relational model for a given database specification.
- CO2** : **Discuss** the relational model concepts and Design relational algebraic expressions for queries.
- CO3** : **Explain** the various concepts of SQL and Design SQL queries to perform CRUD (Create, Retrieve, Update and delete) operations on database.
- CO4** : **Discuss** the database design concepts such as functional dependency and solve the problems on minimal set, equivalence set.
- CO5** : **Discuss** the database design concepts such as Normalization, Relational decomposition and concepts of transaction processing. **Apply** the normalization techniques to improve database design.

### UNIT- I

**DATABASES AND DATABASE USERS:** Introduction; An example; characteristics of the database approach; actors on the scene; workers behind the scene; advantages of using the DBMS approach; A brief history of database Applications; when Not to use a DBMS.

**3 Hrs**

**DATABASE SYSTEM – CONCEPTS AND ARCHITECTURE:** Data models, schemas, and instances; three schema architecture and data independence; database languages and interfaces; the database system environment; centralized and client/server/architectures for DBMSs. Classification of database management system.

**3 Hrs**

**ENTITY-RELATIONSHIP MODEL:** Using High-Level Conceptual Data Models for Database Design; An Example Database Application; Entity Types, Entity Sets, Attributes

and Keys; Relationship types, Relationship Sets, Roles and Structural Constraints; Weak Entity Types; Refining the ER Design for the COMPANY Database; ER Diagrams, Naming Conventions and Design Issues. **5 Hrs**

## UNIT- II

**RELATIONAL MODEL AND RELATIONAL ALGEBRA:** Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations and Dealing with Constraint Violations; Unary Relational Operations: SELECT and PROJECT; Relational Algebra Operations from Set Theory; Binary Relational Operations: JOIN and DIVISION; Additional Relational Operations; Examples of Queries in Relational Algebra; Relational Database Design using ER-to-Relational Mapping. **9 Hrs**

## UNIT-III

**SQL-THE RELATIONAL DATABASE STANDARD:** SQL Data Definition and Data Types, Specifying Basic Constraints in SQL, Schema Change Statements in SQL; Basic Queries in SQL; More Complex SQL Queries; Insert, Delete and Update Statements in SQL; Additional Features of SQL; Specifying General Constraints as Assertion; Views (Virtual Tables) in SQL; Database Programming: Issues and Techniques; Embedded SQL, Dynamic **11 Hrs**

## UNIT-IV

**DATABASE DESIGN:** Informal Design Guidelines for Relation Schemas; Functional Dependencies; Normal Forms Based on Primary Keys; General Definitions of Second and Third Normal Forms; Boyce-Codd Normal Form; Properties of Relational Decompositions. **10 Hrs**

## UNIT-V

**TRANSACTION PROCESSING CONCEPT:** Introduction to transaction processing; transaction and system concepts; desirable properties of transactions, characterizing schedules based on recoverability and serializability; transaction support in SQL. **5 Hrs**

**CONCURRENCY CONTROL & DATABASE RECOVERY TECHNIQUES:** Two phase locking techniques, Concurrency control based on Timestamp ordering; Recovery concepts; recovery based on deferred update and Immediate Update, Shadow Paging, ARIES Recovery Algorithm **6 Hrs**

### TEXT BOOKS:

1.	Elmasri and Navathe.	Fundamentals of Database Systems.. Pearson Education, 7 <sup>th</sup> Edition,2016
----	----------------------	--

### REFERENCE BOOKS:

1.	Abraham Silberschatz, Henry F Korth ,Sudharshan.	Data base System Concepts. Mc GrawHil ( Indian edition) 6 <sup>th</sup> Edition,2013
2.	Raghu Ramakrishnan and Johannes Gehrke	Database Management Systems, McGraw-Hill Education, 3 <sup>rd</sup> Edition,2014

## DATA COMMUNICATION

---

Contact Hours/ Week	: 4 (Lecture)	Credits :	4
Total Lecture Hours	: 52	CIE Marks :	50
Total Tutorial Hours	: --	SEE Marks :	50
Course Code	: 5RCS02		

---

### Course objectives:

1. Understand and Explain basic networking concepts with layers of network models.
2. Identify different types of transmission media and network devices.
3. To analyze and choose appropriate line coding, error detection and correction techniques in transmission of data.
4. Compare and contrast flow control and error control protocols.
5. Interpret random access protocols and Ethernet LAN standards.

### Course outcomes:

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

- CO1** : **Apply** basics of data communication and its components to understand computer networks technology. **Enumerate** the layers of TCP/IP and explain the functions of each layer.
- CO2** : **Analyze** and **Solve** problems on signals, Digital and Analog transmission.
- CO3** : **Experiment** with error detection and correction techniques and **explain** various transmission media.
- CO4** : **Design** multistage switch using basics of switching technology. **Analyze** various flow control and error control and channelization techniques.
- CO5** : **Analyze** various data link layer services and multiple access techniques.

## UNIT - I

### DATA COMMUNICATIONS AND NETWORK MODELS.

**DATA COMMUNICATIONS:** Components, Data Representation, Data Flow, NETWORKS, Network Criteria, Physical Structures, NETWORK TYPES, Local Area Network, Wide Area Network, Switching, The Internet, Accessing the Internet

PROTOCOL LAYERING, Scenarios, Principles of Protocol Layering, Logical Connections, CONTENTS, TCP/IP PROTOCOL SUITE, Layered Architecture, Layers in the TCP/IP Protocol Suite, Description of Each Layer, Encapsulation and Decapsulation, Addressing, Multiplexing and Demultiplexing, THE OSI MODEL, OSI versus TCP/IP, Lack of OSI Model's Success.

**10 Hrs**

## UNIT – II

**DATA AND SIGNALS:** Analog and Digital Data, Analog and Digital Signals, Periodic and Non periodic, Periodic Analog Signals, Sine Wave, Phase, .3 Wavelength, Time and Frequency Domains, Composite Signals, Bandwidth, Digital Signals, Bit Rate, Bit Length, Digital Signal as a Composite Analog Signal, Transmission of Digital Signals, Transmission impairment, Attenuation,

Distortion, Noise, Data Rate Limits, Noiseless Channel: Nyquist Bit Rate, Noisy Channel: Shannon Capacity, Using Both Limits, PERFORMANCE, Bandwidth, Throughput, Latency (Delay), Bandwidth-Delay Product, Jitter.

**DIGITAL-TO-DIGITAL CONVERSION:** Line Coding, Line Coding Schemes, Block Coding, Scrambling, Analog-To-Digital Conversion, Pulse Code Modulation (PCM), Delta Modulation (DM), TRANSMISSION MODES, Parallel Transmission, Serial Transmission.

**DIGITAL-TO-ANALOG CONVERSION:** Aspects of Digital-to-Analog Conversion, Amplitude Shift Keying, Frequency Shift Keying, Phase Shift Keying, Quadrature Amplitude Modulation.

**12 Hrs**

### UNIT - III

#### **ERROR DETECTION & CORRECTION AND TRANSMISSION MEDIA.**

**INTRODUCTION:** Types of Errors, Redundancy, Detection versus Correction, Coding, BLOCK CODING, Error Detection, CYCLIC CODES, Cyclic Redundancy Check, Polynomials, Cyclic Code Encoder Using Polynomials, Cyclic Code Analysis, Advantages of Cyclic Codes, Other Cyclic Codes, Checksum, Concept, Other Approaches to the Checksum, Forward Error Correction, Using Hamming Distance, Using XOR, Chunk Interleaving, Combining Hamming Distance and Interleaving, Compounding High- and Low-Resolution Packet.

**GUIDED MEDIA:** Twisted-Pair Cable, Coaxial Cable, Fiber-Optic Cable, **UNGUIDED MEDIA:** Wireless, Radio Waves, Microwaves, Infrared.

**10 Hrs**

**Web Links for the topic :Guided and Unguided media:**

1. <https://www.geeksforgeeks.org/types-transmission-media/>
2. <https://www.slideshare.net/RajeshYadav330/guided-and-unguided-media-6585852>

### UNIT - IV

#### **SWITCHING AND DATA LINK CONTROL (DLC)**

**SWITCHING:** Introduction, Circuit switched networks, Packet switching, Structure of a switch.

**DLC Services:** Framing, Flow and Error Control, Connectionless and Connection- Oriented, Data-Link Layer Protocols, Simple Protocol, Stop-and-Wait Protocol, Piggybacking, HDLC, Configurations and Transfer Modes, Framing, Point-To-Point Protocol (PPP), Services, Framing, Transition Phases, Multiplexing.

**10 Hrs**

### UNIT - V

#### **MEDIA ACCESS CONTROL (MAC)**

**RANDOM ACCESS:** ALOHA, CSMA, CSMA/CD, CSMA/CA, **CONTROLLED ACCESS:** Reservation, Polling, Token Passing. **CHANNELIZATION:** FDMA, TDMA, CDMA.

**WIRED LANS:** Ethernet, Ethernet Protocol, IEEE Project, Ethernet Evolution, Standard Ethernet, Characteristics, Addressing, Access Method, Efficiency of Standard Ethernet, Implementation, Changes in the Standard.

**10 Hrs**



### Web Links for Ethernet Standards:

1. [http://www.rhyshaden.com/eth\\_intr.htm](http://www.rhyshaden.com/eth_intr.htm)
2. <https://www.computernetworkingnotes.com/networking-tutorials/ethernet-standards-and-protocols-explained.html>
3. <https://www.scribd.com/document/189251933/Ethernet-Standards-pdf>

### TEXT BOOK

1.	Behrouz A. Forouzan	Data Communications and Networking. Tata McGraw-Hill. 5 <sup>th</sup> Edition (Chapters 1.1.- 1.3, 2.1 – 2.3, 3.1 – 3.6, 4.1 – 4.3, 5.1, 7.1-7.3, 8.1-8.4, 10.1 – 10.5, 11.1 – 11.4, 12.1 – 12.3, 13.1- 13.2)
----	---------------------	---

### REFERENCE BOOK

1	Alberto Leon Garcia Indra Widjaja	Communication Networks : Fundamental Concepts & Key Architecture, Tata McGraw – Hill, 2 <sup>nd</sup> Edition
2	William Stallings	Data and Computer Communication. PHI, 5 <sup>th</sup> Edition

## SOFTWARE ENGINEERING

---

Contact Hours/Week	: 4(Lecture)	Credits	: 4.0
Total Lecture Hours	: 52	CIE Marks	: 50
Total Tutorial Hours	: --	SEE Marks	: 50
Course Code	: 5RCS03		

---

### Course objectives:

1. Define the importance of software and process models in developing the large scale Software systems.
2. Identify the requirements for the software to be developed by various mechanisms and interpreting the same as structured, unambiguous, consistent, correct and clear SRS
3. Apply the various design techniques to choose the best architecture and user interface design
4. Design various test cases well in advance to make sure that it is considered in all dimensions
5. Apply analysis and design techniques to Web and Mobile apps.

### Course outcomes:

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

- CO1** : **Analyze** the process model chosen for the development of software and its merits and demerits
- CO2** : **Identify** the clear, correct and consistent requirements for the project
- CO3** : **Design** suitable data, architecture and user interface that copes with the requirements
- CO4** : **Estimate** the cyclomatic complexity and **design** the corresponding test cases.
- CO5** : **Conduct** various integration testing approaches and note down pit falls in requirements, design and test cases
- CO6** : **Analyze** the requirements and design suitable components for Web and Mobile Apps

## UNIT – I

**THE NATURE OF SOFTWARE:** The Nature of Software, The Changing Nature of Software

**SOFTWARE ENGINEERING:** Defining the Discipline, The Software Process, Software Engineering Practice, Software Development Myths.

**THE SOFTWARE PROCESS STRUCTURE:** A Generic Process Model, Defining a Framework Activity, Identifying a Task set, Process Patterns, Process Assessment and Improvement.

**PROCESS MODELS:** Perspective Process Models, Specialized Process Models, The Unified Process, Personal and Team Process Models.

**AGILE DEVELOPMENT:** What is Agility, Agility and the Cost of Change, What is an Agile Process, Extreme Programming, Scrum. **12 Hrs**

## UNIT – II

**UNDERSTANDING REQUIREMENTS:** Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Developing Use Cases, Building the Analysis Model, Negotiating Requirements and Validating Requirements.

**REQUIREMENTS MODELING: SCENARIOS and CLASS BASED METHODS:**

Requirements Analysis, Scenario-Based Modeling, UML Models That Supplement the Use Case, Identifying Analysis Classes, Specifying Attributes, Defining Operations, Class-Responsibility-Collaborator Modeling, Association and Dependencies, Analysis Packages.

**REQUIREMENTS MODELING: BEHAVIOR, PATTERNS :** Creating a Behavioral Model, Identifying Events with the Use Case, State Representations

**11 Hrs**

## UNIT - III

**DESIGN CONCEPTS:** Design within the Context of Software Engineering, The Design Process, Design Concepts, The Design Model .

**ARCHITECTURAL DESIGN:** Software Architecture, Architectural Genres, Architectural Styles, Architectural considerations, Architectural Decisions, Architectural Design (Except 13.6.5 and 13.6.6), Assessing Alternative Architectural Designs.

**COMPONENT-LEVEL DESIGN:** What Is a Component, Designing Class-Based Components, Conducting Component-Level Design, Designing Traditional Components and Component-Based Development.

**USER INTERFACE DESIGN:** The Golden Rules, User interface Analysis and Design.

**11 Hrs**

## UNIT - IV

**SOFTWARE TESTING STRATEGIES:** A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, Test Strategies for Object-Oriented Software, Validation Testing, System Testing and The Art of Debugging.

**TESTING CONVENTIONAL APPLICATIONS:** Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, Basis Path Testing, Control Structure Testing and Black-Box Testing.

**10 Hrs**

## UNIT - V

**WEB AND MOBILE APPS ENGINEERING:** Requirements Modeling for Web and Mobile Apps(11.5),Architectural Design for Web and Mobile Apps (13.6.5 and 13.6.6),Component-Level Design for Web and Mobile Apps(14.4 and 14.5),Web and Mobile Apps interface design(15.5).Test strategies for Web and Mobile Apps(22.5 and 22.6)

**WEBAPP DESIGN:** Design quality, goals, design pyramid, Web App interface, Aesthetic content, architectural, navigation and Component level designs.

**MOBILEAPP DESIGN:** Changes, developing mobile apps, mobile apps- design best practices, mobile environments, The Cloud, The applicability of Conventional software engineering.

**8 Hrs**

**TEXT BOOKS:**

1	Roger S Pressman	Software Engineering- A Practitioner's Approach, 8 <sup>th</sup> edition, TMH publication, 2014.
---	------------------	--

**REFERENCE BOOKS:**

1	Ian Sommerville	Software Engineering, Pearson Education limited, 8 <sup>th</sup> Edition 2007.
2	Pankaj Jalote	An Integrated Approach to Software Engineering, Narosa Publications, 3 <sup>rd</sup> Edition 2005.
3	Rajib Mall	Fundamentals of Software Engineering, PHI India Publications. 5 <sup>th</sup> Edition ,2018.

## JAVA PROGRAMMING LABORATORY

---

Lab Hours/Week	: 2	Credits	: 1.5
Tutorial Hours/ Week	: 1	CIE Marks	: 50
Course Code	: 5RCSL01	SEE Marks	: 50

### Course Outcomes:

After the completion of the course, the student will be able to:

- CO1** : **Design, develop and execute** programs on classes, objects, packages, Principles of inheritance and polymorphism, encapsulation, method overloading, thread priority, exception handling.
- CO2** : **Design** and **develop** web application using servlets and applets
- CO3** : **Design and develop** programs on GUI application using swings, to demonstrate JDBC for database transactions and client server communication

### Course Contents:

1. Programs on packages
2. Programs on interfaces
3. Programs on multithreading and exception handling
4. Programs on applets
5. Programs on servlets
6. Programs on swings
7. Programs on JDBC
8. Programs on sockets

Note: Question bank for SEE will be announced on last working day of the semester.

### TEXT BOOK

1.	Herbert Schildt	Java – The Complete Reference, Tata McGraw-Hill Publications, 11 <sup>th</sup> Edition 2017.
----	-----------------	--

**DATABASE MANAGEMENT SYSTEMS LABORATORY**

---

Lab Hours/Week	: 3	Credits	: 1.5
Tutorial Hours/ Week	: --	CIE Marks	: 50
Course Code	: 5RCSL02	SEE Marks	: 50

---

**Course Outcomes:**

After the completion of the course, the student will be able to:

- CO1** : **Apply** the knowledge of database management system development process and conduct the experiments using SQL queries to find the solution for given database problem.
- CO2** : **Analyze** and design solutions for database system components to meet the specified needs of online transaction processing and information systems like Banking systems, Ticket Reservation systems etc..
- CO3** : **Develop** code for stored programs, triggers assertions and to generate reports.
- CO4** : **Contribute** to the team as a member, lead the team

1. Suppose a movie\_studio has several film crews. The crews might be designated by a given studio as crew1, crew 2, and so on. However, other studios might use the same designations for crews, so the attribute crew\_number is not a key for crews. Movie\_studio holds the information like name, branch and several locations. Each crew holds information like sector and strength.

- Establish the database by normalizing up to 3NF and considering all schema level constraints
- Write SQL insertion query to insert few tuples to all the relations
- List all movie studios which are not used a single crews.
- Retrieve the movie studio which uses highest strength crew.
- Write a before insert trigger to check maximum number of crews to any studio is limited to 5.
- Write a procedure retrieve all crews used by specific studio.

2. The production company is organized into different studios. We store each studio's name branch and location; every studio must own at least one movie. We store each movie's title, sensor number and year of production. Star may act in any number of movies and we store each actors name and address.

- Establish the database by normalizing up to 3NF and considering all schema level constraints
- Write SQL insertion query to insert few tuples to all the relations
- List all the studios of the movie "xyz";
- List all the actors , acted in a movie 'xyz'
- Write a procedure to list all movies produced during the specific year.

- vi. Write a deletion trigger, does not allow to deleting current year movies.
3. The production company is organized into different studios. We store each studio's name branch and location; a studio own any number of Cartoon-serials. We store each Cartoon-Serial's title, sensor number and year of production. Star may do voices in any number of Cartoon-Serials and we store each actors name and address.
- i. Establish the database by normalizing up to 3NF and considering all schema level constraints
  - ii. Write SQL insertion query to insert few tuples to all the relations
  - iii. Find total no of actors, do voiced in a Cartoon-Serials 'xyz'
  - iv. Retrieve name of studio, location and Cartoon-Serials title in which star "abc" is voiced.
- vii. Write a procedure to list all Cartoon-Serials produced during the specific year.
- v. Write a deletion trigger, does not allow to deleting current year Cartoon-Serials.
4. Car marketing company wants keep track of marketed cars and their owner. Each car must be associated with a single owner and owner may have any number of cars. We store car's registration number, model & color and owner's name, address & SSN. We also store date of purchase of each car.
- i. Establish the database by normalizing up to 3NF and considering all schema level constraints
  - ii. Write SQL insertion query to insert few tuples to all the relations
  - iii. Find a person who owns highest number of cars
  - iv. Retrieve persons and cars information purchased on the day 11-11-11
  - v. Write a insertion trigger to check date of purchase must be less than current date (must use system date)
  - vi. Write a procedure to list all cars and owner information purchased during the specific year.
5. Puppy pet shop wants to keep track of dogs and their owners. The person can buy maximum three pet dogs. We store person's name, SSN and address and dog's name, date of purchase and sex. The owner of the pet dogs will be identified by SSN since the dog's names are not distinct.
- i. Establish the database by normalizing up to 3NF and considering all schema level constraints
  - ii. Write SQL insertion query to insert few tuples to all the relations
  - iii. List all pets owned by a person 'Abhiman'.
  - iv. List all persons who are not owned a single pet

- v. Write a trigger to check the constraint that the person can buy maximum three pet dogs
  - vi. Write a procedure to list all dogs and owner details purchased on the specific date.
- 6.** Education institute is managing the on line course enrollment system. Students can enroll maximum of six courses of their choice and a maximum student to be enrolled to any course is 60. We store student details like name, USN, semester and several addresses, course details like unique title, unique id and credits.
- i. Establish the database by normalizing up to 3NF and considering all schema level constraints
  - ii. Write SQL insertion query to insert few tuples to all the relations
  - iii. Find number of students enrolled for the course 'DBMS'
  - iv. Retrieve student names that are enrolled for data structure course but not enrolled for logic design.
  - v. Write a trigger to establish the constraint that the students can enroll maximum of six courses of their choice.
  - vi. Write a procedure to list all the courses enrolled by the seventh semester students.
- 7.** The commercial bank wants keep track of the customer's account information. The each customer may have any number of accounts and account can be shared by any number of customers. The system will keep track of the date of last transaction. We store the following details.
- a) Account: unique account-number, type and balance
  - b) Customer: unique customer-id, name and several addresses composed of street, city and state
- i. Establish the database by normalizing up to 3NF and considering all schema level constraints
  - ii. Write SQL insertion query to insert few tuples to all the relations
  - iii. Add 5% interest to the customer who have less than 10000 balances and 6% interest to remaining customers.
  - iv. List joint accounts involving more than three customers
  - v. Write a insertion trigger to allow only current date for date of last transaction field.
  - vi. Write a procedure to find the customer who has highest number of accounts, the customer who has lowest balance, the customer who involved in most of joint accounts.
- 8.** The commercial bank wants keep track of the customer's loan information. The customer can take any number of loans from the bank and loan will not be shared. The system will also keep track of the date of last transaction. We store the following details.



- a) Customer: unique customer-id, name, Annual Income and several addresses composed of street, city and state
  - b) Loan: unique loan-number, type and amount
  - i. Establish the database by normalizing up to 3NF and considering all schema level constraints
  - ii. Write SQL insertion query to insert few tuples to all the relations
  - iii. Add 12% interest to the customer who have less than 50000 amounts and 14% interest to remaining customers.
  - iv. Retrieve the customers who have a single loan in a bank.
  - v. Write an insertion trigger to loan, that does not allow if the loan amount is more than two times of customer annual income.
  - vi. Write a procedure to retrieve all the loans of a specific customer.
- 9.** The XYZ Book shop wants keep track of orders of the book. The book is composed of unique id, title, year of publication, single author and single publisher. Each order will be uniquely identified by order-id and may have any number of books. We keep track of quantity of each book ordered. We store the following details for author and publisher.
- AUTHOR: unique author-id, name, city, country
- PUBLISHER: unique publisher-id, name, city, country.
- i. Establish the database by normalizing up to 3NF
  - ii. Write SQL insertion query to insert few tuples to all the relations
  - iii. Find the author who has published highest number of books
  - iv. List the books published by specific publisher during the year 2011.
  - v. Write before insertion trigger to book to check year of publication should allow current year only.
  - vi. Write a procedure to list all the books published by a specific author during the specific year.

# **VI - SEMESTER COURSES**

## SOFTWARE PROJECT MANAGEMENT

Contact Hours/Week	: 3 + 0 (L+T)	Credits	: 3.0
Total Lecture Hours	: 39	CIE Marks	: 50
Total Tutorial Hours	: --	SEE Marks	: 50
Course Code	: HSS07A		

### Course objectives:

1. Understand the basics of software project management concepts, principles and practices
2. Understand the different methods of estimation for software project.
3. Understand the basic concepts, principles and practices of software project scheduling and risk management.
4. Analyze a software project based on various review metrics with review guidelines.
5. Understand software project maintenance, reengineering and configuration management

### Course outcomes:

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

- CO1** : **Describe** the basics of software project management concepts, principles and practices.
- CO2** : **Explain** and **apply** the different metrics and techniques to measure a software project.
- CO3** : **Explain** and **analyse** the different activities of a project scheduling with risk management.
- CO4** : **Explain** the quality concepts and analyse different review techniques for a software project.
- CO5** : **Describe** and apply software reengineering activities and SCM process on web and mobile applications

### UNIT - I

#### PRINCIPLES THAT GUIDE PRACTICE:

Software Engineering Knowledge, Core Principles – Principles That Guide Process, Principles That Guide Practice, Principles That Guide Each Framework Activity – Communication Principles, Planning Principles, Modelling Principles, Construction Principles, Deployment Principles, Work Practices

**4Hrs**

#### PROJECT MANAGEMENT CONCEPTS:

The Management Spectrum – The People, The Products, The Process, The Project, People -The Stakeholders, Team Leaders, The Software Team, Agile Teams, Coordination And Communication Issues, The Product – Software Scope, Problem Decomposition, The Process – Melding The Products And The Process, Process Decomposition, The Project, The W5HH Principle, Critical Practices

**3Hrs**

### UNIT - II

#### PROCESS AND PROJECT METRICS:

Metrics In The Process And Project Domains -Process Metrics And Software Process Improvement, Project Metrics, Software Measurement – Size-Oriented Metrics, Function-Oriented

Metrics, Reconciling LOC And FP Metrics, Object-Oriented Metrics, Use Cases- Oriented Metrics, Web App Project Metrics, Metrics For Software Quality – Measuring Quality ,Defect Removal Efficiency, Integrating Metrics With The Software Process - Arguments For Software Metrics, Establishing A Baseline, Metrics Collection Computation And Evaluation, Metrics For Small Organization, Establishing A Software Metrics Program

**3Hrs**

### **ESTIMATION FOR SOFTWARE PROJECT:**

Observations On Estimation, The Project Planning Process, Software Scope And Feasibility, Resources – Human Resources, Reusable Software Resources, Environmental Resources, Software Project Estimation, Decomposition Techniques – Software Sizing, Problem Based Estimation, An Example Of LOC Based Estimation, An Example Of FP – Based Estimation, Process-Based Estimation, An Example Of Process- Based Estimation, Estimation With Use cases, An Example Of Estimation Using Use Case Points, Reconciling Estimates, Empirical Estimation Models – The Structure Of Estimation Models, The COCOMO II Model, The Software Equation, Estimation For Object -Oriented Projects, Specialized Estimation Techniques – Estimation For Agile Development, Estimation For Web App Development, The Make/Buy Decision- Creating A Decision Tree, Outsourcing.

**5Hrs**

## **UNIT - III**

### **PROJECT SCHEDULING:**

Basic concepts, Project Scheduling – Basic Principles - The Relationship Between People and Effort – Effort Distribution, Defining a Task Set for The Software Project – a Task Set Example – Refinement of Major Tasks, Defining a Task Network, Scheduling – Timeline Charts – Tracking The Schedule – Tracking Progress for an OO Project – Scheduling for Web App and Mobile Projects, Earned Value Analysis

**4Hrs**

### **RISK MANAGEMENT :**

Reactive Verses Proactive Risk Strategies, Software Risks, Risk Identification – Assessing Overall Project Risk – Components and Drivers, Risk Projection – Developing a Risk Table – Assessing Risk Impact, Risk Refinement, Risk Mitigation, Monitoring, and Management, The RMMM Plan

**3Hrs**

## **UNIT – IV**

### **QUALITY CONCEPTS:**

What is Quality? Software Quality – Garvin's Quality Dimensions, McColl's Quality Factors, ISO 9126 Quality Factors, Targeted Quality Factors, The Transition to a Quantitative View, The Software Quality Dilemma - “Good Enough” Software, The Cost Of Quality, Risks, Negligence and Liability, Quality and Security, The Impact Of Management Actions, Achieving Software Quality – Software Engineering Methods, Project Management Techniques, Quality Control, Quality Assurance

**4Hrs**

### **REVIEW TECHNIQUES:**

Cost Impact of Software Defects, Defect Amplification and Removal, Review Metrics and Their

Use – Analysing Metrics, Cost-Effectiveness of Reviews, Reviews: a Formality Spectrum, Informal Reviews, Formal Technical Reviews – The Review Meeting, Review Reporting and Record Keeping, Review Guidelines, Sample- Driven Reviews, Post-mortem Evaluations

**3Hrs**

### **SOFTWARE QUALITY ASSURANCE:**

Background Issues, Elements of Software Quality Assurance, SQA Processes and Product Characteristics, SQA Tasks, Goals and Metrics – SQA Tasks, Goals Attributes and Metrics, Formal Approaches To SQA, Statistical Software Quality Assurance – A Generic Example, SixSigma for Software Engineering, Software Reliability – Measures of Reliability and Availability, Software Safety, The ISO 9000 Quality Standards, The SQA Plan

**3Hrs**

## **UNIT – V**

### **MAINTENANCE AND REENGINEERING:**

Software Maintenance, Software Supportability, Reengineering, Business Process Reengineering - Business Processes, A BPR Model, Software Reengineering – A Software Reengineering Process Model, Software Reengineering Activities, Reverse Engineering – Reverse Engineering to Understand Data, Reverse Engineering to Understand Processing, Reverse Engineering User Interfaces, Restructuring – Code Restructuring, Data Restructuring, Forward Engineering – Forward Engineering for Client Server Architectures, Forward Engineering for Object- Oriented Architectures, The Economics Of Reengineering

**3Hrs**

### **SOFTWARE CONFIGURATION MANAGEMENT:**

Software Configuration Management – An SCM Scenario, Elements of a Configuration Management System, Baselines, Software Configuration Items, Management of Dependencies And Changes, The SCM Repository – General Features and Content, SCM Features, The SCM Process – Identification of Objects In The Software Configuration, Version Control, Change Control, Impact Management, Configuration Audit, Status Reporting, Configuration Management for Web And Mobile Apps – Dominant Issues, Configuration Objects, Content Management, Change Management, Version Control, Auditing And Reporting

**4Hrs**

### **TEXT BOOKS:**

1.	Roger S. Pressman, Bruce Maxim	Software Engineering: A Practitioner's Approach, McGraw Hill, 8 <sup>th</sup> Edition, 2015
----	-----------------------------------	--

### **REFERENCE BOOKS:**

1.	Hughes, Mike Cotterell, Rajib Mall	Software Project Management, McGraw Hill, 6 <sup>th</sup> Edition 2018
2.	Watts Humphrey	Managing the Software Process, Pearson Education, New Delhi, 2000
3.	Pankaj Jalote	Software Project Management in Practice, Pearson Education, New Delhi, 2002.

## COMPUTER NETWORKS

Contact Hours/Week	: 4 (Lecture)	Credits	: 4.0
Total Lecture Hours	: 52	CIE Marks	: 50
Total Tutorial Hours	: --	SEE Marks	: 50
Course Code	: 6RCS01		

### COURSE OBJECTIVES:

1. Introduces to congestion control and resource allocation. Explores various routing techniques for unicasting and multicasting.
2. Introduces internetworking and describes the key elements of the IP.
3. Analyze the transport-layer concepts: Transport-Layer services Reliable vs. unreliable data transfer -TCP protocol -UDP protocol and QoS.
4. Explore various application layer protocols and addresses different mechanisms used to provide quality of service in IP.

### COURSE OUTCOMES:

After the completion of the course, the student will be able to:

- CO1** : **Apply** the knowledge of Packet switching concepts in computer networking
- CO2** : **Identify** different categories of IP addresses and design subnets.
- CO3** : **Analyze** different Unicast and multicast routing mechanisms.
- CO4** : **Analyze** the transport-layer concepts and services -unreliable vs. reliable data transfer
- CO5** : **Examine** various network protocols and **Appraise** existing QoS and application layer protocol/s.

### UNIT-I

**NETWORK LAYER:** NETWORK-LAYER SERVICES: Packetizing, Routing and Forwarding Other Services. NETWORK-LAYER PERFORMANCE: Delay, Throughput, Packet loss. Congestion Control. IPV4 ADDRESSES: Address Space, Classfull Addressing, Classless Addressing, Dynamic Host Configuration Protocol (DHCP), Network Address Resolution (NAT). FORWARDING OF IP PACKETS: Forwarding Based on Destination Address, Forwarding Based on Label, Routers as Packet Switches.

**10Hrs**

### UNIT-II

#### NETWORK LAYER PROTOCOLS:

**INTERNET PROTOCOL (IP):** Datagram Format, Fragmentation, Options, Security of IPv4 Datagrams.

**UNICAST ROUTING:** INTRODUCTION: General Idea, Least-Cost Routing. ROUTING ALGORITHMS: Distance-Vector Routing, Link-State Routing, Path-Vector Routing.

**UNICAST ROUTING PROTOCOLS:** Internet Structure, Routing Information Protocol (RIP), Open Shortest Path First (OSPF), Border Gateway Protocol Version 4 (BGP4).

**10Hrs**

### **UNIT-III**

**MULTICAST ROUTING:** MULTICASTING BASICS: Multicast Addresses, Delivery at Data-Link Layer, Collecting Information about Groups, Multicast Forwarding, Two Approaches to Multicasting. INTRADOMAIN MULTICAST PROTOCOLS: Multicast Distance Vector (DVMRP), Multicast Link State (MOSPF), Protocol Independent Multicast (PIM).

**NEXT GENERATION IP:** IPv6 ADDRESSING: Representation, Address Space, Address Space Allocation, Auto configuration, Renumbering. THE IPv6 PROTOCOL: Packet Format, Extension Header, TRANSITION FROM IPv4 TO IPv6: Strategies, Use of IP Addresses.

**10Hrs**

### **UNIT-IV**

**TRANSPORT LAYER PROTOCOLS:** INTRODUCTION: Services, Port Numbers.

**USER DATAGRAM PROTOCOL:** User Datagram, UDP Services, UDP Applications.

**TRANSMISSION CONTROL PROTOCOL:** TCP Services, TCP Features, Segment, A TCP Connection, State Transition Diagram, Windows in TCP, Flow Control, Error Control (**except Sender and Receiver FSMs**), TCP Congestion Control, TCP Timers.

**10Hrs**

### **UNIT-V**

#### **STANDARD CLIENT-SERVER PROTOCOLS AND QOS:**

**WORLD WIDE WEB AND HTTP:** World Wide Web. Hypertext Transfer Protocol (HTTP). **ELECTRONIC MAIL:** Architecture, Web-Based Mail, E-Mail Security. **DNS:** Name Space, DNS in the Internet, Resolution, Caching, Resource Records, DNS Messages.

**QUALITY OF SERVICE:** DATA-FLOW CHARACTERISTICS: Definitions, Sensitivity of Applications, Flow Classes. FLOW CONTROL TO IMPROVE QOS; Scheduling, Traffic Shaping or Policing, Resource Reservation, Admission Control. INTEGRATED SERVICES (INTSERV): Flow Specification, Admission, Service Classes. Resource Reservation Protocol (RSVP), Problems with Integrated Services. DIFFERENTIATED SERVICES (DFFSERV): DS Field, Per-Hop Behavior, Traffic Conditioners.

**12Hrs**

#### **TEXT BOOK:**

1	Behrouz A. Forouzan	Data Communications and Networking, , McGraw-Hill Forouzan Networking Series, 5 <sup>th</sup> Edition,
---	---------------------	--

**REFERENCE BOOKS:**

1	James F Kurose and Keith W Ross	Computer Networking, A Top-Down Approach, , Pearson Education, Sixth edition 2017.
2	Larry L Peterson and Bruce S Davie	Computer Networks, ELSEVIER, 6 <sup>Th</sup> Edition
3	Andrew S Tanenbaum	Computer Networks, Pearson Education, 5 <sup>th</sup> Edition
4	Mayank Dave	Computer Networks, Cengage Learning, 5 <sup>th</sup> Edition

**WEB RESOURCES:**

Topics	Web Links
Wireshark	
a. Wireshark Website	<a href="http://www.wireshark.org">http://www.wireshark.org</a>
b. Wireshark Wiki	<a href="http://wiki.wireshark.org">http://wiki.wireshark.org</a>
c. Wireshark documentation	<a href="http://www.wireshark.org/docs/">http://www.wireshark.org/docs/</a>
Tcpdump	
a. Tcpdump Website	<a href="http://www.tcpdump.org/">http://www.tcpdump.org/</a>
b. Tcpdump Tutorial	<a href="https://opensource.com/article/18/10/introduction-tcpdump">https://opensource.com/article/18/10/introduction-tcpdump</a>
C-net	<a href="http://www.csse.uwa.edu.au/cnet/">http://www.csse.uwa.edu.au/cnet/</a>
BGP Demo	<a href="https://www.bgp4.as/looking-glasses">https://www.bgp4.as/looking-glasses</a>



**SYSTEM SOFTWARE AND COMPILER DESIGN**

Contact Hours/Week	: 3 + 3 (Lecture + Lab)	Credits : 4.5
Total Lecture Hours	: 39	CIE Marks : 50
Laboratory Hours/ Week	: 3	SEE Marks : 50
Course Code	: 6RCS02	

---

**Note:** Four quizzes of theory are replaced by four lab tests. Each lab test is conducted for 15 marks and entered for corresponding quiz.

**COURSE OBJECTIVES:**

1. Introduce the major concept areas of language translation and compiler design.
2. Extend the knowledge of parser by parsing LL parser and LR parser.
3. Evaluate the ideas in syntax- directed definitions & Syntax – directed translations.
4. Use the knowledge gained for generating the intermediate code for a typical programming language.
5. Introduce the basic concepts of Assemblers, Macro processors, Linkers & Loaders.

**COURSE OUTCOMES:**

After the completion of the course, the student will be able to:

- CO1** : **Discuss** and **Apply** the fundamentals of compiler design to construct various components of modern compiler.
- CO2** : **Discuss** functions of lexical analyzer and **design** transition diagram based lexical analyzer.
- CO3** : **Analyze** and **apply** the various forms of context free grammars for the construction of parsers.
- CO4** : **Design** LR (0), SLR (1) parsers by using the LR (0) items.
- CO5** : **Discuss** and **Apply** the techniques of Syntax Directed Translation and Intermediate Code Generation in the construction of compiler
- CO6** : **Describe** the functions of various system software's (loaders, linkers and assemblers)

**UNIT-I**

**INTRODUCTION:** Language processors, The structure of a Compiler, The Evolution of Programming Languages , The Move to Higher-level Languages, Impacts on Compilers

**LEXICAL ANALYSIS:** The Role of Lexical Analyzer, Lexical Analysis Versus Parsing, Tokens, Patterns, and Lexemes, Attributes for Tokens, Lexical Errors, Input Buffering, Buffer Pairs, Sentinels, Specification of Tokens, Strings and Languages, Operations on Languages, Regular Expressions, Regular Definitions, Extensions of Regular Expressions.  
[Text book 1 : 1.1,1.2, 1.3.1, 1.3.2, 3.1 to 3.3] **8 Hrs**

**UNIT-II**

**LEXICAL ANALYSIS:** Recognition of Tokens, Transition Diagrams, Recognition of Reserved Words and Identifiers, Completion of the Running Example, Architecture of a Transition-Diagram-Based Lexical Analyzer.

**SYNTAX ANALYSIS :** The role of parser, Representative Grammars, syntax error handling, error recovery strategies, Writing a grammar, lexical versus syntactic analysis, Eliminating ambiguity, Elimination of left-recursion, Left-factoring.

[Text book 1 : 3.4, 4.1, 4.3.1 to 4.3.4]

**8 Hrs**

### UNIT-III

**TOP-DOWN PARSING:** Introduction, Recursive-Descent Parsing, FIRST and FOLLOW, LL(1) grammars , Constructing a predictive parsing table , Non recursive Predictive Parsing, Error Recovery in Predictive Parsing: Panic mode Error Recovery and Phrase level Error Recovery

[ Text book 1 : 4.4.1 to 4.4.5]

**7Hrs**

### UNIT-IV

**BOTTOM-UP PARSING:**, Reductions, Handle Pruning, Shift-reduce parsing and conflicts during Shift-reduce parsing, Introduction to LR Parsing: Simple LR, Why LR parsers?, Items and LR(0) automaton, Closure of Item Sets, The Function GOTO, LR(0) automaton for the expression grammar, The LR-Parsing Algorithm, Constructing SLR-parsing tables.

[Text book 1: 4.5.1 to 4.5.4, 4.6.1 to 4.6.4]

**8Hrs**

### UNIT-V

**SYNTAX-DIRECTED TRANSLATION:** Syntax directed definitions, Inherited and synthesized attributes, evaluating an SDD at the nodes of the parse tree.

[Text book 1 : 5.1.1 to 5.1.2]

**INTERMEDIATE-CODE GENERATION:** Three-address code – Addresses and instructions, Quadruples and Triples.

[Text book 1 : 6.2.1 to 6.2.3]

**INTRODUCTION TO SYSTEM SOFTWARE:** Assemblers: Elements of Assembly Language programming, A Simple Assembly Scheme and Pass Structure of Assemblers. Linkers and Loaders: Relocation, Linking and Loading Concepts.

[Text book 2 : 4.1 to 4.3, 7.1.1, 7.1.2, 7.1.3, 7.6]

**8Hrs**

### TEXTBOOK(S):

1	Compilers- Principles, Techniques and Tools	Alfred V Aho, Monica S.Lam, Ravi Sethi, Jeffrey D Ullman, Pearson Education, 2 <sup>nd</sup> Edition 2007. (Chapters 1.1 to 1.4, 3.1 to 3.4, 4.1, 4.3 to 4.7, 5.1 to 5.2, 6.1 to 6.3)
2	System Programming and Operating Systems	D M Dhamdhare , McGraw Hill. 2 <sup>nd</sup> Revised Edition,, (Chapters 4.1 to 4.3, 5.1 to 5.3 , 7.1 and 7.6)

### REFERENCE BOOKS:

1	Compiler Construction Principles & Practice	Kenneth C Loudon , Thomson Education, 1997.
2	Modern Compiler Implementation in C	Andrew W Appel, First Edition, Cambridge University Press, 2010

## **LABORATORY CONTENT:**

### **Programs on the Operating System concepts:**

1. Process Scheduling.
2. Process Synchronization.
3. Deadlock Avoidance.
4. Memory allocation techniques.
5. Page Replacement Algorithms.

### **Programs on the Compiler Design concepts:**

1. Lexical Analysis.
2. Context Free Grammars.
3. Left recursion and left factoring in top -down parsing.
4. FIRST all nonterminal in the given grammar.
5. Top-Down parsers.
6. LR(0) item set for the given grammar.

Note : Students have to implement the programs using C/C++.

## COMPUTER NETWORKS LABORATORY

Lab Hours/Week	: 3	Credits	: 1.5
Course Code	: 6RCSL01	CIE Marks	: 50
		SEE Marks	: 50

### COURSE OUTCOMES:

After the completion of the course, the student will be able to:

- CO1** : **Implement and analyze** various flow control and congestion control mechanisms in network applications.
- CO2** : **Implement and analyze** error detection and correction techniques in network applications.
- CO3** : **Develop and Implement** network routing algorithms by applying network programming concepts.
- CO4** : **Implement** client server applications with TCP/UDP socket programming..
- CO5** : **Analyze** the performance of networking protocols using NS2/NS3/GNS3 simulator for wired and wireless networks

**OTE** : Students have to implement and execute all the listed programs. List of programs for SEE will be announced before the last working day of the semester.

#### I. Implement the following programs using C/C++ or equivalent on LINUX environment:

##### 1. Implementation of flow control Mechanisms:

- Sliding Window
- Stop & Wait
- Go back N
- Selective Repeat

##### 2. Implementation of Error Detection and Correction Mechanisms:

- Cyclic Redundancy Check
- Checksum
- Forward Error correction
- Hamming Code

##### 3. Implementation of Routing algorithms:

- Link state
- Distance Vector

##### 4. Implementation of congestion control algorithms:

- Leaky bucket
- Token bucket

##### 5. Client server communication using Socket Programming:

- Implementation of unicast communication using TCP
- Implementation of unicast communication using UDP
- Implementation of unicast communication using FIFO
- Multicast application
- Broadcast application

#### II. Simulate and analyze the performance of wired and wireless networks protocols using NS2/ NS3/ GNS3 simulator.

**III. Open ended programs:**

1. Understanding and using of commands like ifconfig, netstat, ping, arp, telnet, ftp, finger, traceroute, who is.
2. Using Wireshark trace the following
  - Hypertext Transfer Protocol.
  - Trace File Transfer protocol, Trace Transmission control protocol
  - Trace Domain Name Server.
  - Trace Internet Protocol and Internet Control Message Protocol.
3. Using Wireshark observe data transferred in client server communication using UDP and identify the UDP datagram.
4. Using Wireshark observe Three Way Handshaking Connection Establishment, Data Transfer and Three-Way Handshaking Connection Termination in client server communication using TCP.
5. Develop a packet capturing and filtering application using raw sockets.
6. Simulate a three nodes point – to – point network with duplex links between them. Set the queue size and vary the bandwidth and find the number of packets dropped.
7. Simulate a four node point-to-point network with the links connected as follows. n0-n2, n1-n2, and n2-n3. Apply TCP agent between n0-n3 and UDP agent between n1-n3. Apply relevant applications over TCP and UDP agents. Changing the parameters and determine the number of packets sent by TCP/UDP.
8. Simulate the different types of Internet traffic such as FTP and TELNET over a network and analyze the throughput.
9. Simulate an Ethernet LAN using n nodes (6-10), change error rate and data rate and compare the throughput.
10. Simulate an Ethernet LAN using n nodes and set multiple traffic nodes and determine the collision across different nodes.
11. Simulate the transmission of ping messages over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.
12. Simulate simple ESS with transmitting nodes in wire-less LAN and determine the performance with respect to transmission of packets.
13. Simulate simple ad-hoc network with transmitting nodes and determine the performance with respect to transmission of packets.
14. Design and configure a network with multiple subnets with wired and wireless LANs using required network devices. Configure the following services in the network- TELNET, SSH, FTP server, Web server, File server, DHCP server and DNS server.
15. Write a program for a HDLC frame to perform the following.
  - i) Bit stuffing
  - ii) Character stuffing.
16. Write a program to demonstrate communication with HTTP server using sockets.
17. Write a program to connect to DNS server to resolve the IP address.
18. Write a program to implement Simple Mail Transfer Protocol.

## MOBILE APPLICATION DEVELOPMENT LABORATORY

Lab Hours/Week	: 3	Credits	: 2.0
Tutorial Hours/Week	: 1	CIE Marks	: 50
Course Code	: 6RCSL02	SEE Marks	: 50

### COURSE OUTCOMES:

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

**CO1 : Create**, test and **debug** Android application by setting up Android development environment.

**CO2 : Implement** adaptive, responsive user interfaces that work across a wide range of devices.

**CO3 : Infer** long running tasks and background work in Android applications.

**CO4 : Demonstrate** methods in storing, sharing and retrieving data in Android applications.

**CO5 : Infer** the role of permissions and security for Android applications.

**NOTE:** Students have to implement and execute all the listed programs. However for the conduction of the lab exams only part A questions need to be asked as part B questions are for demonstration purpose only.

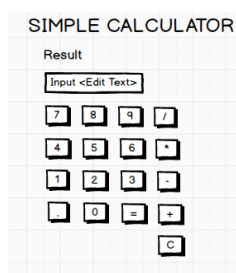
### LAB QUESTIONS:

#### PART – A

- 1 Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address is to be displayed. Insert a horizontal line between the job title and the phone number.



- 2 Develop an Android application using controls like Button, Text View, Edit Text for designing a calculator having basic functionality like Addition, Subtraction, Multiplication and Division.



- 3 Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules:
- Password should contain uppercase and lower case letters.
  - Password should contain letters and numbers.
  - Password should contain special characters.
  - Minimum length of the password (the default value is 8).

On successful **SIGN UP** proceed to the next Login activity. Here the user should **SIGN IN** using the Username and Password created during signup activity. If the Username and Password are matched then navigate to the next activity which displays a message saying “Successful Login” or else display a toast message saying “Login Failed”. The user is given only two attempts and after that display a toast message saying “Failed Login Attempts” and disable the SIGN IN button. Use Bundle to transfer information from one activity to another.

The image shows two side-by-side activity layouts on a grid background. The left layout is titled 'SIGNUP ACTIVITY' and contains labels 'Username:' and 'Password:' followed by empty text input fields. Below the fields is a button labeled 'SIGN UP'. The right layout is titled 'LOGIN ACTIVITY' and contains similar 'Username:' and 'Password:' labels with input fields, and a button labeled 'SIGN IN' below them.

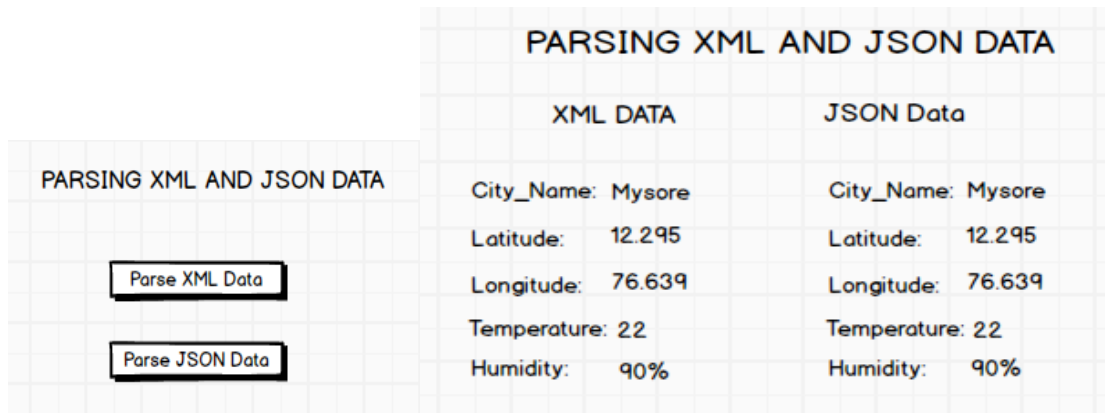
- 4 Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.

The image shows a single activity layout titled 'CHANGING WALLPAPER APPLICATION' centered at the top. Below the title is a large rectangular button with the text 'CLICK HERE TO CHANGE WALLPAPER'.

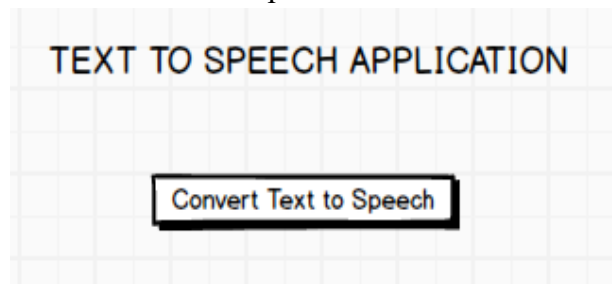
- 5 Write a program to create an activity with two buttons START and STOP. On pressing of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a Text View control.

The image shows an activity layout titled 'COUNTER APPLICATION' centered at the top. Below the title is a text label 'Counter Value'. Underneath the label are two buttons: 'START' and 'STOP', arranged vertically.

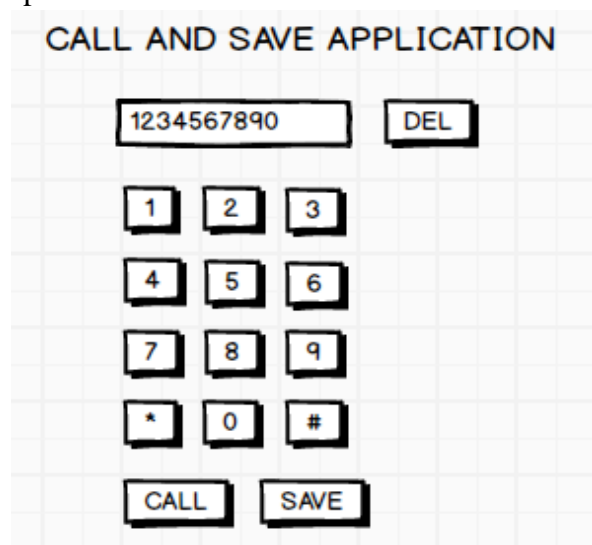
- 6 Create two files of XML and JSON type with values for City Name, Latitude, Longitude, Temperature and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.



- 7 Develop a simple application with one Edit Text so that the user can write some text in it. Create a button called “Convert Text to Speech” that converts the user input text into voice.



- 8 Create an activity like a phone dialer with CALL and SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phonecontacts.





- 9 Write a program to enter Medicine Name, Date and Time of the Day as input from the user and store it in the SQLite database. Input for Time of the Day should be either Morning or Afternoon or Evening or Night. Trigger an alarm based on the Date and Time of the Day and display the Medicine Name.

The screenshot shows a form titled "MEDICINE DATABASE". It contains three input fields: "Medicine Name:", "Date:", and "Time of the Day:". Below these fields is an "Insert" button.

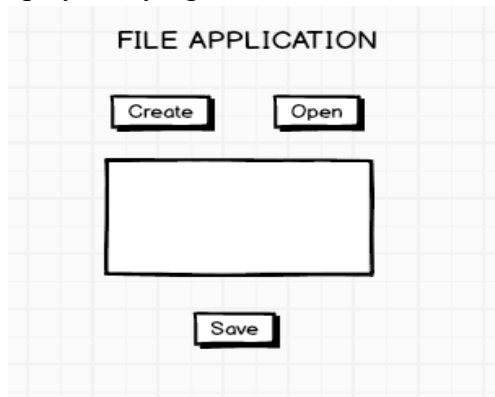
- 10 Develop a content provider application with an activity called "Meeting Schedule" which takes Date, Time and Meeting Agenda as input from the user and store this information into the SQLite database. Create another application with an activity called "Meeting Info" having Date Picker control, which on the selection of a date should display the Meeting Agenda information for that particular date, else it should display a toast message saying "No Meeting on this Date".

The image shows two application interfaces. On the left is the "MEETING SCHEDULE" app, which has input fields for "Date:", "Time:", and "Meeting Agenda:", and an "Add Meeting Agenda" button. On the right is the "MEETING INFO" app, which has a "Pick a date to get meeting info:" label, a date input field, a calendar icon, and a "Search" button. A date picker dialog is shown, displaying the month of July 2018, with the 23rd selected. The dialog has "CANCEL" and "OK" buttons.

- 11 Create an application to receive an incoming SMS which is notified to the user. On clicking this SMS notification, the message content and the number should be displayed on the screen. Use appropriate emulator control to send the SMS message to your application.

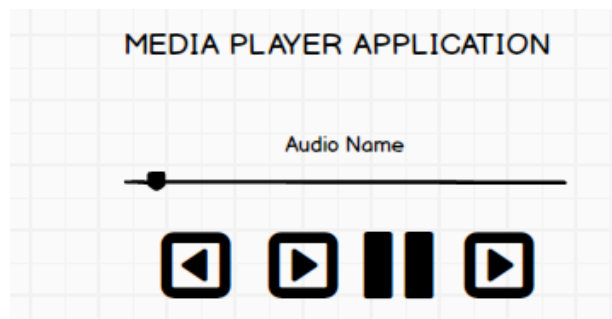
The screenshot shows an application titled "SMS APPLICATION". It contains two text labels: "Display SMS Number" and "Display SMS Message".

- 12 Write a program to create an activity having a Text box, and also Save, Open and Create buttons. The user has to write some text in the Text box. On pressing the Create button the text should be saved as a text file in MkSDcard. On subsequent changes to the text, the Save button should be pressed to store the latest content to the same file. On pressing the Open button, it should display the contents from the previously stored files in the Text box. If the user tries to save the contents in the Textbox to a file without creating it, then a toast message has to be displayed saying “First Create a File”.

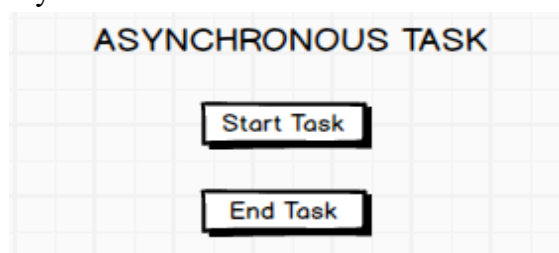


### Part B

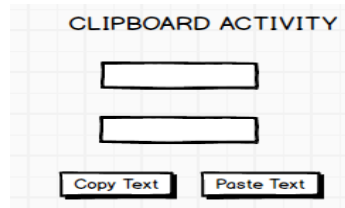
- 1 Create an application to demonstrate a basic media player that allows the user to Forward, Backward, Play and Pause an audio. Also, make use of the indicator in the seek bar to move the audio forward or backward as required.



- 2 Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchronous task should implement the functionality of a simple moving banner. On pressing the **Start Task** button, the banner message should scroll from right to left. On pressing the **Stop Task** button, the banner message should stop. Let the banner message be “Demonstration of Asynchronous Task”.



- 3 Develop an application that makes use of the clipboard framework for copying and pasting of the text. The activity consists of two Edit Text controls and two Buttons to trigger the copy and paste functionality.



- 4 Create an AIDL service that calculates Car Loan EMI. The formula to calculate EMI is

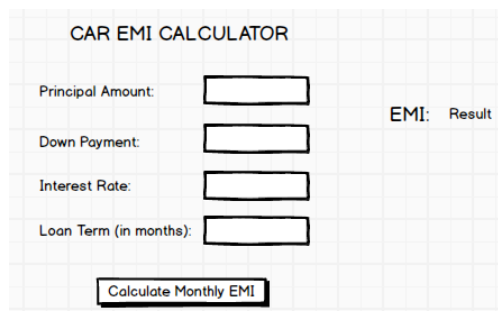
$$E = P * (r(1+r)^n)/((1+r)^n - 1), \text{ where}$$

E = The EMI payable on the car loan

amount P = The Car loan Principal Amount

r = The interest rate value computed on a monthly basis  
n = The loan tenure in the form of months

The down payment amount has to be deducted from the principal amount paid towards buying the Car. Develop an application that makes use of this AIDL service to calculate the EMI. This application should have four Edit Text to read the Principal Amount, Down Payment, Interest Rate, Loan Term (in months) and a button named as “Calculate Monthly EMI”. On click of this button, the result should be shown in a Text View. Also, calculate the EMI by varying the Loan Term and Interest Rate values.



#### Text Books:

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

#### Reference Books:

1. Erik Hellman, “Android Programming – Pushing the Limits”, 1<sup>st</sup> Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197
2. Dawn Griffiths and David Griffiths, “Head First Android Development”, 1<sup>st</sup> Edition, O’Reilly SPD Publishers, 2015. ISBN-13:978-9352131341
3. Bill Phillips, Chris Stewart and Kristin Marsicano, “Android Programming: The Big Nerd Ranch Guide”, 3<sup>rd</sup> Edition, Big Nerd Ranch Guides, 2017. ISBN-13:978-0134706054