

SYSTEM SOFTWARE AND COMPILERS (Effective from the academic year 2018 -2019) SEMESTER – VI			
Course Code	18CS61	CIE Marks	40
Number of Contact Hours/Week	3:2:0	SEE Marks	60
Total Number of Contact Hours	50	Exam Hours	03
CREDITS –4			
Course Learning Objectives: This course (18CS61) will enable students to:			
<ul style="list-style-type: none"> • Define System Software. • Familiarize with source file, object file and executable file structures and libraries • Describe the front-end and back-end phases of compiler and their importance to students 			
Module 1			Contact Hours
Introduction to System Software, Machine Architecture of SIC and SIC/XE. Assemblers: Basic assembler functions, machine dependent assembler features, machine independent assembler features, assembler design options. Basic Loader Functions Text book 1: Chapter 1: 1.1,1.2,1.3.1,1.3.2, Chapter2 : 2.1 to 2.4, Chapter 3 ,3.1 RBT: L1, L2, L3			10
Module 2			
Introduction: Language Processors, The structure of a compiler, The evaluation of programming languages, The science of building compiler, Applications of compiler technology. Lexical Analysis: The role of lexical analyzer, Input buffering, Specifications of token, recognition of tokens. Text book 2:Chapter 1 1.1-1.5 Chapter 3: 3.1 – 3.4 RBT: L1, L2, L3			10
Module 3			
Syntax Analysis: Introduction, Context Free Grammars, Writing a grammar, Top Down Parsers, Bottom-Up Parsers Text book 2: Chapter 4 4.1, 4.2 4.3 4.4 4.5 RBT: L1, L2, L3			10
Module 4			
Lex and Yacc –The Simplest Lex Program, Grammars, Parser-Lexer Communication, A YACC Parser, The Rules Section, Running LEX and YACC, LEX and Hand- Written Lexers, Using LEX - Regular Expression, Examples of Regular Expressions, A Word Counting Program, Using YACC – Grammars, Recursive Rules, Shift/Reduce Parsing, What YACC Cannot Parse, A YACC Parser - The Definition Section, The Rules Section, The LEXER, Compiling and Running a Simple Parser, Arithmetic Expressions and Ambiguity. Text book 3: Chapter 1,2 and 3. RBT: L1, L2, L3			10
Module 5			
Syntax Directed Translation, Intermediate code generation, Code generation Text book 2: Chapter 5.1, 5.2, 5.3, 6.1, 6.2, 8.1, 8.2 RBT: L1, L2, L3			10
Course Outcomes: The student will be able to :			
<ul style="list-style-type: none"> • Explain system software • Design and develop lexical analyzers, parsers and code generators • Utilize lex and yacc tools for implementing different concepts of system software 			

Question Paper Pattern:

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

Textbooks:

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

Reference Books:

1. Systems programming – Srimanta Pal , Oxford university press, 2016
2. System programming and Compiler Design, K C Louden, Cengage Learning
3. System software and operating system by D. M. Dhamdhare TMG
4. Compiler Design, K Muneeswaran, Oxford University Press 2013.

COMPUTER GRAPHICS AND VISUALIZATION (Effective from the academic year 2018 -2019) SEMESTER – VI			
Course Code	18CS62	CIE Marks	40
Number of Contact Hours/Week	3:2:0	SEE Marks	60
Total Number of Contact Hours	50	Exam Hours	03
CREDITS –4			
Course Learning Objectives: This course (18CS62) will enable students to:			
<ul style="list-style-type: none"> • Explain hardware, software and OpenGL Graphics Primitives. • Illustrate interactive computer graphic using the OpenGL. • Design and implementation of algorithms for 2D graphics Primitives and attributes. • Demonstrate Geometric transformations, viewing on both 2D and 3D objects. • Infer the representation of curves, surfaces, Color and Illumination models 			
Module 1			Contact Hours
<p>Overview: Computer Graphics and OpenGL: Computer Graphics: Basics of computer graphics, Application of Computer Graphics, Video Display Devices: Random Scan and Raster Scan displays, graphics software. OpenGL: Introduction to OpenGL ,coordinate reference frames, specifying two-dimensional world coordinate reference frames in OpenGL, OpenGL point functions, OpenGL line functions, point attributes, line attributes, curve attributes, OpenGL point attribute functions, OpenGL line attribute functions, Line drawing algorithms(DDA, Bresenham’s), circle generation algorithms (Bresenham’s).</p> <p>Text-1:Chapter -1: 1-1 to 1-9, 2-1(page 39 to 41),2.8,2.9,3-1 to 3-5,3-9,3-20</p> <p>RBT: L1, L2, L3</p>			10
Module 2			
<p>Fill area Primitives, 2D Geometric Transformations and 2D viewing: Fill area Primitives: Polygon fill-areas, OpenGL polygon fill area functions, fill area attributes, general scan line polygon fill algorithm, OpenGL fill-area attribute functions. 2DGeometric Transformations: Basic 2D Geometric Transformations, matrix representations and homogeneous coordinates. Inverse transformations, 2DComposite transformations, other 2D transformations, raster methods for geometric transformations, OpenGL raster transformations, OpenGL geometric transformations function, 2D viewing: 2D viewing pipeline, OpenGL 2D viewing functions.</p> <p>Text-1:Chapter 3-14 to 3-16,4-9,4-10,4-14,5-1 to 5-7,5-17,6-1,6-4</p> <p>RBT: L1, L2, L3</p>			10
Module 3			
<p>Clipping,3D Geometric Transformations, Color and Illumination Models: Clipping: clipping window, normalization and viewport transformations, clipping algorithms,2D point clipping, 2D line clipping algorithms: cohen-sutherland line clipping only -polygon fill area clipping: Sutherland-Hodgeman polygon clipping algorithm only.3DGeometric Transformations: 3D translation, rotation, scaling, composite 3D transformations, other 3D transformations, affine transformations, OpenGL geometric transformations functions. Color Models: Properties of light, color models, RGB and CMY color models. Illumination Models: Light sources, basic illumination models-Ambient light, diffuse reflection, specular and phong model, Corresponding openGL functions.</p> <p>Text-1:Chapter :6-2 to 6-08 (Excluding 6-4),5-9 to 5-17(Excluding 5-15),12-1,12-2,12-4,12-6,10-1,10-3</p> <p>RBT: L1, L2, L3</p>			10
Module 4			
3D Viewing and Visible Surface Detection: 3DViewing:3D viewing concepts, 3D viewing			10

<p>pipeline, 3D viewing coordinate parameters , Transformation from world to viewing coordinates, Projection transformation, orthogonal projections, perspective projections, The viewport transformation and 3D screen coordinates. OpenGL 3D viewing functions. Visible Surface Detection Methods: Classification of visible surface Detection algorithms, depth buffer method only and OpenGL visibility detection functions.</p> <p>Text-1:Chapter: 7-1 to 7-10(Excluding 7-7), 9-1,9-3, 9-14</p> <p>RBT: L1, L2, L3</p>	
<p>Module 5</p>	
<p>Input& interaction, Curves and Computer Animation: Input and Interaction: Input devices, clients and servers, Display Lists, Display Lists and Modeling, Programming Event Driven Input, Menus Picking, Building Interactive Models, Animating Interactive programs, Design of Interactive programs, Logic operations .Curved surfaces, quadric surfaces, OpenGL Quadric-Surface and Cubic-Surface Functions, Bezier Spline Curves, Bezier surfaces, OpenGL curve functions. Corresponding openGL functions.</p> <p>Text-1:Chapter :8-3 to 8-6 (Excluding 8-5),8-9,8-10,8-11,3-8,8-18,13-11,3-2,13-3,13-4,13-10</p> <p>Text-2:Chapter 3: 3-1 to 3.11: Input& interaction</p> <p>RBT: L1, L2, L3</p>	<p>10</p>
<p>Course Outcomes: The student will be able to :</p>	
<ul style="list-style-type: none"> • Design and implement algorithms for 2D graphics primitives and attributes. • Illustrate Geometric transformations on both 2D and 3D objects. • Apply concepts of clipping and visible surface detection in 2D and 3D viewing, and Illumination Models. • Decide suitable hardware and software for developing graphics packages using OpenGL. 	
<p>Question Paper Pattern:</p>	
<ul style="list-style-type: none"> • The question paper will have ten questions. • Each full Question consisting of 20 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. 	
<p>Textbooks:</p>	
<ol style="list-style-type: none"> 1. Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version,3rd / 4th Edition, Pearson Education,2011 2. Edward Angel: Interactive Computer Graphics- A Top Down approach with OpenGL, 5th edition. Pearson Education, 2008 	
<p>Reference Books:</p>	
<ol style="list-style-type: none"> 1. James D Foley, Andries Van Dam, Steven K Feiner, John F Huges Computer graphics with OpenGL: pearson education 2. Xiang, Plastock : Computer Graphics , sham’s outline series, 2nd edition, TMG. 3. Kelvin Sung, Peter Shirley, steven Baer : Interactive Computer Graphics, concepts and applications, Cengage Learning 4. M M Raikar & Shreedhara K S Computer Graphics using OpenGL, Cengage publication 	

WEB TECHNOLOGY AND ITS APPLICATIONS (Effective from the academic year 2018 -2019) SEMESTER – VI			
Course Code	18CS63	CIE Marks	40
Number of Contact Hours/Week	3:2:0	SEE Marks	60
Total Number of Contact Hours	50	Exam Hours	03
CREDITS –4			
Course Learning Objectives: This course (18CS63) will enable students to:			
<ul style="list-style-type: none"> • Illustrate the Semantic Structure of HTML and CSS • Compose forms and tables using HTML and CSS • Design Client-Side programs using JavaScript and Server-Side programs using PHP • Infer Object Oriented Programming capabilities of PHP • Examine JavaScript frameworks such as jQuery and Backbone 			
Module 1			Contact Hours
Introduction to HTML, What is HTML and Where did it come from?, HTML Syntax, Semantic Markup, Structure of HTML Documents, Quick Tour of HTML Elements, HTML5 Semantic Structure Elements, Introduction to CSS, What is CSS, CSS Syntax, Location of Styles, Selectors, The Cascade: How Styles Interact, The Box Model, CSS Text Styling. Textbook 1: Ch. 2, 3 RBT: L1, L2, L3			10
Module 2			
HTML Tables and Forms, Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, Table and Form Accessibility, Microformats, Advanced CSS: Layout, Normal Flow, Positioning Elements, Floating Elements, Constructing Multicolumn Layouts, Approaches to CSS Layout, Responsive Design, CSS Frameworks. Textbook 1: Ch. 4,5 RBT: L1, L2, L3			10
Module 3			
JavaScript: Client-Side Scripting, What is JavaScript and What can it do?, JavaScript Design Principles, Where does JavaScript Go?, Syntax, JavaScript Objects, The Document Object Model (DOM), JavaScript Events, Forms, Introduction to Server-Side Development with PHP, What is Server-Side Development, A Web Server’s Responsibilities, Quick Tour of PHP, Program Control, Functions Textbook 1: Ch. 6, 8 RBT: L1, L2, L3			10
Module 4			
PHP Arrays and Superglobals, Arrays, \$_GET and \$_POST Superglobal Arrays, \$_SERVER Array, \$_FILES Array, Reading/Writing Files, PHP Classes and Objects, Object-Oriented Overview, Classes and Objects in PHP, Object Oriented Design, Error Handling and Validation, What are Errors and Exceptions?, PHP Error Reporting, PHP Error and Exception Handling Textbook 1: Ch. 9, 10 RBT: L1, L2, L3			10
Module 5			
Managing State, The Problem of State in Web Applications, Passing Information via Query Strings, Passing Information via the URL Path, Cookies, Serialization, Session State, HTML5 Web Storage, Caching, Advanced JavaScript and jQuery, JavaScript Pseudo-Classes, jQuery Foundations, AJAX, Asynchronous File Transmission, Animation, Backbone			10

MVC Frameworks, XML Processing and Web Services, XML Processing, JSON, Overview of Web Services. Textbook 1: Ch. 13, 15,17 RBT: L1, L2, L3	
Course Outcomes: The student will be able to :	
<ul style="list-style-type: none"> • Adapt HTML and CSS syntax and semantics to build web pages. • Construct and visually format tables and forms using HTML and CSS • Develop Client-Side Scripts using JavaScript and Server-Side Scripts using PHP to generate and display the contents dynamically. • Appraise the principles of object oriented development using PHP • Inspect JavaScript frameworks like jQuery and Backbone which facilitates developer to focus on core features. 	
Question Paper Pattern:	
<ul style="list-style-type: none"> • The question paper will have ten questions. • Each full Question consisting of 20 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. 	
Textbooks:	
1. Randy Connolly, Ricardo Hoar, " Fundamentals of Web Development ", 1 st Edition, Pearson Education India. (ISBN:978-9332575271)	
Reference Books:	
<ol style="list-style-type: none"> 1. Robin Nixon, "Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5", 4thEdition, O'Reilly Publications, 2015. (ISBN:978-9352130153) 2. Luke Welling, Laura Thomson, "PHP and MySQL Web Development", 5th Edition, Pearson Education, 2016. (ISBN:978-9332582736) 3. Nicholas C Zakas, "Professional JavaScript for Web Developers", 3rd Edition, Wrox/Wiley India, 2012. (ISBN:978-8126535088) 4. David Sawyer Mcfarland, "JavaScript & jQuery: The Missing Manual", 1st Edition, O'Reilly/Shroff Publishers & Distributors Pvt Ltd, 2014 	
Mandatory Note:	
Distribution of CIE Marks is as follows (Total 40 Marks):	
<ul style="list-style-type: none"> • 20 Marks through IA Tests • 20 Marks through practical assessments 	
Maintain a copy of the report for verification during LIC visit.	
Possible list of practicals:	
<ol style="list-style-type: none"> 1. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient. 2. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format. 3. Write a JavaScript code that displays text "TEXT-GROWING" with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays "TEXT-SHRINKING" in BLUE color. Then the font size decreases to 5pt. 4. Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems: <ol style="list-style-type: none"> a. Parameter: A string b. Output: The position in the string of the left-most vowel 	

- c. Parameter: A number
 - d. Output: The number with its digits in the reverse order
5. Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, and Name of the College, Programme, Year of Joining, and email id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.
6. Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.
7. Write a PHP program to display a digital clock which displays the current time of the server.
8. Write the PHP programs to do the following:
 - a. Implement simple calculator operations.
 - b. Find the transpose of a matrix.
 - c. Multiplication of two matrices.
 - d. Addition of two matrices.
9. Write a PHP program named states.py that declares a variable states with value "Mississippi Alabama Texas Massachusetts Kansas". write a PHP program that does the following:
 - a. Search for a word in variable states that ends in xas. Store this word in element 0 of a list named statesList.
 - b. Search for a word in states that begins with k and ends in s. Perform a case-insensitive comparison. [Note: Passing re.I as a second parameter to method compile performs a case-insensitive comparison.] Store this word in element1 of statesList.
 - c. Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list.
 - d. Search for a word in states that ends in a. Store this word in element 3 of the list.
10. Write a PHP program to sort the student records which are stored in the database using selection sort.

DATA MINING AND DATA WAREHOUSING (Effective from the academic year 2018 -2019) SEMESTER – VI			
Course Code	18CS641	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Course Learning Objectives: This course (18CS641) will enable students to:			
<ul style="list-style-type: none"> • Define multi-dimensional data models. • Explain rules related to association, classification and clustering analysis. • Compare and contrast between different classification and clustering algorithms 			
Module 1			Contact Hours
Data Warehousing & modeling: Basic Concepts: Data Warehousing: A multitier Architecture, Data warehouse models: Enterprise warehouse, Data mart and virtual warehouse, Extraction, Transformation and loading, Data Cube: A multidimensional data model, Stars, Snowflakes and Fact constellations: Schemas for multidimensional Data models, Dimensions: The role of concept Hierarchies, Measures: Their Categorization and computation, Typical OLAP Operations Textbook 2: Ch.4.1,4.2 RBT: L1, L2, L3			08
Module 2			
Data warehouse implementation& Data mining: Efficient Data Cube computation: An overview, Indexing OLAP Data: Bitmap index and join index, Efficient processing of OLAP Queries, OLAP server Architecture ROLAP versus MOLAP Versus HOLAP. : Introduction: What is data mining, Challenges, Data Mining Tasks, Data: Types of Data, Data Quality, Data Preprocessing, Measures of Similarity and Dissimilarity. Textbook 2: Ch.4.4 Textbook 1: Ch.1.1,1.2,1.4, 2.1 to 2.4 RBT: L1, L2, L3			08
Module 3			
Association Analysis: Association Analysis: Problem Definition, Frequent Item set Generation, Rule generation. Alternative Methods for Generating Frequent Item sets, FP-Growth Algorithm, Evaluation of Association Patterns. Textbook 1: Ch 6.1 to 6.7 (Excluding 6.4) RBT: L1, L2, L3			08
Module 4			
Classification: Decision Trees Induction, Method for Comparing Classifiers, Rule Based Classifiers, Nearest Neighbor Classifiers, Bayesian Classifiers. Textbook 1: Ch 4.3,4.6,5.1,5.2,5.3 RBT: L1, L2, L3			08
Module 5			
Clustering Analysis: Overview, K-Means, Agglomerative Hierarchical Clustering, DBSCAN, Cluster Evaluation, Density-Based Clustering, Graph-Based Clustering, Scalable Clustering Algorithms. Textbook 1: Ch 8.1 to 8.5, 9.3 to 9.5 RBT: L1, L2, L3			08
Course Outcomes: The student will be able to :			

- Identify data mining problems and implement the data warehouse
- Write association rules for a given data pattern.
- Choose between classification and clustering solution.

Question Paper Pattern:

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

Textbooks:

1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining, Pearson, First impression,2014.
2. Jiawei Han, Micheline Kamber, Jian Pei: Data Mining -Concepts and Techniques, 3rd Edition, Morgan Kaufmann Publisher, 2012.

Reference Books:

1. Sam Anahory, Dennis Murray: Data Warehousing in the Real World, Pearson,Tenth Impression,2012.
2. Michael.J.Berry,Gordon.S.Linoff: Mastering Data Mining , Wiley Edition, second edtion,2012.

OBJECT ORIENTED MODELING AND DESIGN (Effective from the academic year 2018 -2019) SEMESTER – VI			
Course Code	18CS642	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Course Learning Objectives: This course (18CS642) will enable students to:			
<ul style="list-style-type: none"> • Describe the concepts involved in Object-Oriented modelling and their benefits. • Demonstrate concept of use-case model, sequence model and state chart model for a given problem. • Explain the facets of the unified process approach to design and build a Software system. • Translate the requirements into implementation for Object Oriented design. • Choose an appropriate design pattern to facilitate development procedure. 			
Module 1			Contact Hours
Advanced object and class concepts; Association ends; N-ary associations; Aggregation; Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived Data; Packages. State Modeling: Events, States, Transitions and Conditions, State Diagrams, State diagram behaviour. Text Book-1: 4, 5 RBT: L1, L2			08
Module 2			
UseCase Modelling and Detailed Requirements: Overview; Detailed object-oriented Requirements definitions; System Processes-A use case/Scenario view; Identifying Input and outputs-The System sequence diagram; Identifying Object Behaviour-The state chart Diagram; Integrated Object-oriented Models. Text Book-2:Chapter- 6:Page 210 to 250 RBT: L1, L2, L3			08
Module 3			
Process Overview, System Conception and Domain Analysis: Process Overview: Development stages; Development life Cycle; System Conception: Devising a system concept; elaborating a concept; preparing a problem statement. Domain Analysis: Overview of analysis; Domain Class model: Domain state model; Domain interaction model; Iterating the analysis. Text Book-1:Chapter- 10,11,and 12			08
Module 4			
Use case Realization :The Design Discipline within up iterations: Object Oriented Design-The Bridge between Requirements and Implementation; Design Classes and Design within Class Diagrams; Interaction Diagrams-Realizing Use Case and defining methods; Designing with Communication Diagrams; Updating the Design Class Diagram; Package Diagrams-Structuring the Major Components; Implementation Issues for Three-Layer Design. Text Book-2: Chapter 8: page 292 to 346 RBT: L1, L2, L3			08
Module 5			
Design Patterns: Introduction; what is a design pattern?, Describing design patterns, the catalogue of design patterns, Organizing the catalogue, How design patterns solve design problems, how to select a design patterns, how to use a design pattern; Creational patterns: prototype and singleton (only); structural patterns adaptor and proxy (only).			08

Text Book-3: Ch-1: 1.1, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8,Ch-3,Ch-4. RBT: L1, L2, L3	
Course Outcomes: The student will be able to :	
<ul style="list-style-type: none"> • Describe the concepts of object-oriented and basic class modelling. • Draw class diagrams, sequence diagrams and interaction diagrams to solve problems. • Choose and apply a befitting design pattern for the given problem. 	
Question Paper Pattern:	
<ul style="list-style-type: none"> • The question paper will have ten questions. • Each full Question consisting of 20 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. 	
Textbooks:	
<ol style="list-style-type: none"> 3. Michael Blaha, James Rumbaugh: Object Oriented Modelling and Design with UML,2nd Edition, Pearson Education,2005 4. Satzinger, Jackson and Burd: Object-Oriented Analysis & Design with the Unified Process, Cengage Learning, 2005. 5. Erich Gamma, Richard Helm, Ralph Johnson and john Vlissides: Design Patterns –Elements of Reusable Object-Oriented Software, Pearson Education,2007. 	
Reference Books:	
<ol style="list-style-type: none"> 1. Grady Booch et. al.: Object-Oriented Analysis and Design with Applications,3rd Edition,Pearson Education,2007. 2. 2.Frank Buschmann, RegineMeunier, Hans Rohnert, Peter Sommerlad, Michel Stal: Pattern – Oriented Software Architecture. A system of patterns , Volume 1, John Wiley and Sons.2007. 3. 3. Booch, Jacobson, Rambaugh : Object-Oriented Analysis and Design with Applications, 3rd edition, pearson, Reprint 2013 	

CLOUD COMPUTING AND ITS APPLICATIONS (Effective from the academic year 2018 -2019) SEMESTER – VI			
Course Code	18CS643	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Course Learning Objectives: This course (18CS643) will enable students to:			
<ul style="list-style-type: none"> • Explain the fundamentals of cloud computing • Illustrate the cloud application programming and aneka platform • Contrast different cloud platforms used in industry 			
Module 1			Contact Hours
Introduction ,Cloud Computing at a Glance, The Vision of Cloud Computing, Defining a Cloud, A Closer Look, Cloud Computing Reference Model, Characteristics and Benefits, Challenges Ahead, Historical Developments, Distributed Systems, Virtualization, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing, Building Cloud Computing Environments, Application Development, Infrastructure and System Development, Computing Platforms and Technologies, Amazon Web Services (AWS), Google AppEngine, Microsoft Azure, Hadoop, Force.com and Salesforce.com, Manjrasoft Aneka Virtualization, Introduction, Characteristics of Virtualized, Environments Taxonomy of Virtualization Techniques, Execution Virtualization, Other Types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples Xen: Paravirtualization, VMware: Full Virtualization, Microsoft Hyper-V Textbook 1: Ch. 1,3 RBT: L1, L2			08
Module 2			
Cloud Computing Architecture, Introduction, Cloud Reference Model, Architecture, Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds, Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges, Cloud Definition, Cloud Interoperability and Standards Scalability and Fault Tolerance Security, Trust, and Privacy Organizational Aspects Aneka: Cloud Application Platform, Framework Overview, Anatomy of the Aneka Container, From the Ground Up: Platform Abstraction Layer, Fabric Services, foundation Services, Application Services, Building Aneka Clouds, Infrastructure Organization, Logical Organization, Private Cloud Deployment Mode, Public Cloud Deployment Mode, Hybrid Cloud Deployment Mode, Cloud Programming and Management, Aneka SDK, Management Tools Textbook 1: Ch. 4,5 RBT: L1, L2			08
Module 3			
Concurrent Computing: Thread Programming, Introducing Parallelism for Single Machine Computation, Programming Applications with Threads, What is a Thread?, Thread APIs, Techniques for Parallel Computation with Threads, Multithreading with Aneka, Introducing the Thread Programming Model, Aneka Thread vs. Common Threads, Programming Applications with Aneka Threads, Aneka Threads Application Model, Domain Decomposition: Matrix Multiplication, Functional Decomposition: Sine, Cosine, and Tangent. High-Throughput Computing: Task Programming, Task Computing, Characterizing a Task,			08

<p>Computing Categories, Frameworks for Task Computing, Task-based Application Models, Embarrassingly Parallel Applications, Parameter Sweep Applications, MPI Applications, Workflow Applications with Task Dependencies, Aneka Task-Based Programming, Task Programming Model, Developing Applications with the Task Model, Developing Parameter Sweep Application, Managing Workflows.</p> <p>Textbook 1: Ch. 6, 7 RBT: L1, L2</p>	
<p>Module 4</p>	
<p>Data Intensive Computing: Map-Reduce Programming, What is Data-Intensive Computing?, Characterizing Data-Intensive Computations, Challenges Ahead, Historical Perspective, Technologies for Data-Intensive Computing, Storage Systems, Programming Platforms, Aneka MapReduce Programming, Introducing the MapReduce Programming Model, Example Application</p> <p>Textbook 1: Ch. 8 RBT: L1, L2</p>	08
<p>Module 5</p>	
<p>Cloud Platforms in Industry, Amazon Web Services, Compute Services, Storage Services, Communication Services, Additional Services, Google AppEngine, Architecture and Core Concepts, Application Life-Cycle, Cost Model, Observations, Microsoft Azure, Azure Core Concepts, SQL Azure, Windows Azure Platform Appliance.</p> <p>Cloud Applications Scientific Applications, Healthcare: ECG Analysis in the Cloud, Biology: Protein Structure Prediction, Biology: Gene Expression Data Analysis for Cancer Diagnosis, Geoscience: Satellite Image Processing, Business and Consumer Applications, CRM and ERP, Productivity, Social Networking, Media Applications, Multiplayer Online Gaming.</p> <p>Textbook 1: Ch. 9,10 RBT: L1, L2</p>	08
<p>Course Outcomes: The student will be able to :</p>	
<ul style="list-style-type: none"> • Explain cloud computing, virtualization and classify services of cloud computing • Illustrate architecture and programming in cloud • Describe the platforms for development of cloud applications and List the application of cloud. 	
<p>Question Paper Pattern:</p>	
<ul style="list-style-type: none"> • The question paper will have ten questions. • Each full Question consisting of 20 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. 	
<p>Textbooks:</p>	
<p>1. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi Mastering Cloud. Computing McGraw Hill Education</p>	
<p>Reference Books:</p>	
<p>1. Dan C. Marinescu, Cloud Computing Theory and Practice, Morgan Kaufmann, Elsevier 2013.</p>	

ADVANCED JAVA AND J2EE (Effective from the academic year 2018 -2019) SEMESTER – VI			
Course Code	18CS644	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Course Learning Objectives: This course (18CS644) will enable students to: <ul style="list-style-type: none"> • Identify the need for advanced Java concepts like Enumerations and Collections • Construct client-server applications using Java socket API • Make use of JDBC to access database through Java Programs • Adapt servlets to build server side programs • Demonstrate the use of JavaBeans to develop component-based Java software 			
Module 1			Contact Hours
Enumerations, Autoboxing and Annotations(metadata): Enumerations, Enumeration fundamentals, the values() and valueOf() Methods, java enumerations are class types, enumerations Inherits Enum, example, type wrappers, Autoboxing, Autoboxing and Methods, Autoboxing/Unboxing occurs in Expressions, Autoboxing/Unboxing, Boolean and character values, Autoboxing/Unboxing helps prevent errors, A word of Warning. Annotations, Annotation basics, specifying retention policy, Obtaining Annotations at run time by use of reflection, Annotated element Interface, Using Default values, Marker Annotations, Single Member annotations, Built-In annotations. Textbook 1: Lesson 12 RBT: L1, L2, L3			08
Module 2			
The collections and Framework: Collections Overview, Recent Changes to Collections, The Collection Interfaces, The Collection Classes, Accessing a collection Via an Iterator, Storing User Defined Classes in Collections, The Random Access Interface, Working With Maps, Comparators, The Collection Algorithms, Why Generic Collections?, The legacy Classes and Interfaces, Parting Thoughts on Collections. Text Book 1: Ch.17 RBT: L1, L2, L3			08
Module 3			
String Handling : The String Constructors, String Length, Special String Operations, String Literals, String Concatenation, String Concatenation with Other Data Types, String Conversion and toString() Character Extraction, charAt(), getChars(), getBytes() toCharArray(), String Comparison, equals() and equalsIgnoreCase(), regionMatches() startsWith() and endsWith(), equals() Versus == , compareTo() Searching Strings, Modifying a String, substring(), concat(), replace(), trim(), Data Conversion Using valueOf(), Changing the Case of Characters Within a String, Additional String Methods, StringBuffer , StringBuffer Constructors, length() and capacity(), ensureCapacity(), setLength(), charAt() and setCharAt(), getChars(),append(), insert(), reverse(), delete() and deleteCharAt(), replace(), substring(), Additional StringBuffer Methods, StringBuilder			08
Module 4			

<p>Background; The Life Cycle of a Servlet; Using Tomcat for Servlet Development; A simple Servlet; The Servlet API; The Javax.servlet Package; Reading Servlet Parameter; The Javax.servlet.http package; Handling HTTP Requests and Responses; Using Cookies; Session Tracking. Java Server Pages (JSP): JSP, JSP Tags, Tomcat, Request String, User Sessions, Cookies, Session Objects</p> <p>Text Book 1: Ch 31 Text Book 2: Ch 11</p> <p>RBT: L1, L2, L3</p>	08
Module 5	
<p>The Concept of JDBC; JDBC Driver Types; JDBC Packages; A Brief Overview of the JDBC process; Database Connection; Associating the JDBC/ODBC Bridge with the Database; Statement Objects; ResultSet; Transaction Processing; Metadata, Data types; Exceptions.</p> <p>Text Book 2: Ch 06</p> <p>RBT: L1, L2, L3</p>	08
Course Outcomes: The student will be able to :	
<ul style="list-style-type: none"> • Interpret the need for advanced Java concepts like enumerations and collections in developing modular and efficient programs • Build client-server applications and TCP/IP socket programs • Illustrate database access and details for managing information using the JDBC API • Describe how servlets fit into Java-based web application architecture • Develop reusable software components using Java Beans 	
Question Paper Pattern:	
<ul style="list-style-type: none"> • The question paper will have ten questions. • Each full Question consisting of 20 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. 	
Textbooks:	
<ol style="list-style-type: none"> 1. Herbert Schildt: JAVA the Complete Reference, 7th/9th Edition, Tata McGraw Hill, 2007. 2. Jim Keogh: J2EE-TheCompleteReference, McGraw Hill, 2007. 	
Reference Books:	
<ol style="list-style-type: none"> 1. Y. Daniel Liang: Introduction to JAVA Programming, 7thEdition, Pearson Education, 2007. 2. Stephanie Bodoff et al: The J2EE Tutorial, 2nd Edition, Pearson Education,2004. 3. Uttam K Roy, Advanced JAVA programming, Oxford University press, 2015. 	

SYSTEM MODELLING AND SIMULATION
(Effective from the academic year 2018 -2019)
SEMESTER – VI

Course Code	18CS645	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Course Learning Objectives: This course (18CS645) will enable students to:			
<ul style="list-style-type: none"> • Explain the basic system concept and definitions of system; • Discuss techniques to model and to simulate various systems; • Analyze a system and to make use of the information to improve the performance. 			
Module 1			Contact Hours
Introduction: When simulation is the appropriate tool and when it is not appropriate, Advantages and disadvantages of Simulation; Areas of application, Systems and system environment; Components of a system; Discrete and continuous systems, Model of a system; Types of Models, Discrete-Event System Simulation Simulation examples: Simulation of queuing systems. General Principles. Textbook 1: Ch. 1, 2, 3.1.1, 3.1.3 RBT: L1, L2, L3			08
Module 2			
Statistical Models in Simulation :Review of terminology and concepts, Useful statistical models,Discrete distributions. Continuous distributions,Poisson process, Empirical distributions. Queuing Models: Characteristics of queuing systems,Queuing notation,Long-run measures of performance of queuing systems,Long-run measures of performance of queuing systems cont...,Steady-state behavior of M/G/1 queue, Networks of queues, Textbook 1: Ch. 5,6.1 to 6.3, 6.4.1,6.6 RBT: L1, L2, L3			08
Module 3			
Random-Number Generation: Properties of random numbers; Generation of pseudo-random numbers, Techniques for generating random numbers,Tests for Random Numbers, Random-Variate Generation: ,Inverse transform technique Acceptance-Rejection technique. Textbook 1: Ch. 7,8.1, 8.2 RBT: L1, L2, L3			08
Module 4			
Input Modeling: Data Collection; Identifying the distribution with data, Parameter estimation, Goodness of Fit Tests, Fitting a non-stationary Poisson process, Selecting input models without data, Multivariate and Time-Series input models. Estimation of Absolute Performance: Types of simulations with respect to output analysis ,Stochastic nature of output data, Measures of performance and their estimation, Contd.. Textbook 1: Ch. 9, 11.1 to 11.3 RBT: L1, L2, L3			08
Module 5			
Measures of performance and their estimation,Output analysis for terminating simulations Continued...,Output analysis for steady-state simulations. Verification, Calibration And Validation: Optimization: Model building, verification and validation, Verification of simulation models, Verification of simulation models,Calibration and validation of models, Optimization via Simulation.			08

Textbook 1: Ch. 11.4, 11.5, 10 RBT: L1, L2, L3	
Course Outcomes: The student will be able to :	
<ul style="list-style-type: none"> • Explain the system concept and apply functional modeling method to model the activities of a static system • Describe the behavior of a dynamic system and create an analogous model for a dynamic system; • Simulate the operation of a dynamic system and make improvement according to the simulation results. 	
Question Paper Pattern:	
<ul style="list-style-type: none"> • The question paper will have ten questions. • Each full Question consisting of 20 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. 	
Textbooks:	
1. Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol: Discrete-Event System Simulation, 5 th Edition, Pearson Education, 2010.	
Reference Books:	
1. Lawrence M. Leemis, Stephen K. Park: Discrete – Event Simulation: A First Course, Pearson Education, 2006.	
2. Averill M. Law: Simulation Modeling and Analysis, 4 th Edition, Tata McGraw-Hill, 2007	

**MOBILE APPLICATION DEVELOPMENT
(OPEN ELECTIVE)
(Effective from the academic year 2018 -2019)
SEMESTER – VI**

Course Code	18CS651	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Course Learning Objectives: This course (18CS651) will enable students to:			
<ul style="list-style-type: none"> • Learn to setup Android application development environment • Illustrate user interfaces for interacting with apps and triggering actions • Interpret tasks used in handling multiple activities • Identify options to save persistent application data • Appraise the role of security and performance in Android applications 			
Module – 1			Teaching Hours
Get started, Build your first app, Activities, Testing, debugging and using support libraries Textbook 1: Lesson 1,2,3 RBT: L1, L2			08
Module – 2			
User Interaction, Delightful user experience, Testing your UI Textbook 1: Lesson 4,5,6 RBT: L1, L2			08
Module – 3			
Background Tasks, Triggering, scheduling and optimizing background tasks Textbook 1: Lesson 7,8 RBT: L1, L2			08
Module – 4			
All about data, Preferences and Settings, Storing data using SQLite, Sharing data with content providers, Loading data using Loaders Textbook 1: Lesson 9,10,11,12 RBT: L1, L2			08
Module – 5			
Permissions, Performance and Security, Firebase and AdMob, Publish// Textbook 1: Lesson 13,14,15 RBT: L1, L2			08
Course outcomes: The students should be able to:			
<ul style="list-style-type: none"> • Create, test and debug Android application by setting up Android development environment • Implement adaptive, responsive user interfaces that work across a wide range of devices. • Infer long running tasks and background work in Android applications • Demonstrate methods in storing, sharing and retrieving data in Android applications • Analyze performance of android applications and understand the role of permissions and security • Describe the steps involved in publishing Android application to share with the world 			
Question Paper Pattern:			
<ul style="list-style-type: none"> • The question paper will have ten questions. • Each full Question consisting of 20 marks 			

- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.

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

Textbooks:

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

INTRODUCTION TO DATA STRUCTURES AND ALGORITHM (OPEN ELECTIVE) (Effective from the academic year 2018 -2019) SEMESTER – VI			
Course Code	18CS652	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Course Learning Objectives: This course (18CS652) will enable students to:			
<ul style="list-style-type: none"> • Identify different data structures in C programming language • Appraise the use of data structures in problem solving • Implement data structures using C programming language. 			
Module 1			Contact Hours
Introduction to C, constants, variables, data types, input output operations, operators and expressions, control statements, arrays, strings, built-in functions, user defined functions, structures, unions and pointers Text Book 1: Chapter 1 and 2 RBT: L1, L2			08
Module 2			
Algorithms, Asymptotic notations, Introduction to data structures, Types of data structures, Arrays. Text Book 1: Chapter 3 and 4 RBT: L1, L2			08
Module 3			
Linked lists, Stacks Text Book 1: Chapter 5 and 6 RBT: L1, L2			08
Module 4			
Queues, Trees Text Book 1: Chapter 7 and 8 RBT: L1, L2			08
Module 5			
Graphs, Sorting ,(selection, insertion, bubble, quick)and searching(Linear, Binary, Hash) Text Book 1: Chapter 7 and 8 RBT: L1, L2			08
Course Outcomes: The student will be able to :			
<ul style="list-style-type: none"> • Identify different data structures in C programming language • Appraise the use of data structures in problem solving • Implement data structures using C programming language. 			
Question Paper Pattern:			
<ul style="list-style-type: none"> • The question paper will have ten questions. • Each full Question consisting of 20 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. 			
Textbooks:			
1. Data structures using C , E Balagurusamy, McGraw Hill education (India) Pvt. Ltd, 2013.			
Reference Books:			

1. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014.
2. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.

PROGRAMMING IN JAVA
(OPEN ELECTIVE)
(Effective from the academic year 2018 -2019)
SEMESTER – VI

Course Code	18CS653	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Course Learning Objectives: This course (18CS653) will enable students to:			
<ul style="list-style-type: none"> • Learn fundamental features of object oriented language and JAVA • Set up Java JDK environment to create, debug and run simple Java programs. • Learn object oriented concepts using programming examples. • Study the concepts of importing of packages and exception handling mechanism. • Discuss the String Handling examples with Object Oriented concepts 			
Module – 1			Teaching Hours
<p>An Overview of Java: Object-Oriented Programming, A First Simple Program, A Second Short Program, Two Control Statements, Using Blocks of Code, Lexical Issues, The Java Class Libraries, Data Types, Variables, and Arrays: Java Is a Strongly Typed Language, The Primitive Types, Integers, Floating-Point Types, Characters, Booleans, A Closer Look at Literals, Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays, A Few Words About Strings</p> <p>Text book 1: Ch 2, Ch 3 RBT: L1, L2</p>			08
Module – 2			
<p>Operators: Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses, Control Statements: Java’s Selection Statements, Iteration Statements, Jump Statements.</p> <p>Text book 1: Ch 4, Ch 5 RBT: L1, L2</p>			08
Module – 3			
<p>Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection, The finalize() Method, A Stack Class, A Closer Look at Methods and Classes: Overloading Methods, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Arrays Revisited, Inheritance: Inheritance, Using super, Creating a Multilevel Hierarchy, When Constructors Are Called, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, The Object Class.</p> <p>Text book 1: Ch 6, Ch 7.1-7.9, Ch 8. RBT: L1, L2</p>			08
Module – 4			
<p>Packages and Interfaces: Packages, Access Protection, Importing Packages, Interfaces, Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java’s Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Using Exceptions.</p>			08

Text book 1: Ch 9, Ch 10 RBT: L1, L2	
Module – 5	
Enumerations, Type Wrappers, I/O, Applets, and Other Topics: I/O Basics, Reading Console Input, Writing Console Output, The PrintWriter Class, Reading and Writing Files, Applet Fundamentals, The transient and volatile Modifiers, Using instanceof, strictfp, Native Methods, Using assert, Static Import, Invoking Overloaded Constructors Through this(), String Handling: The String Constructors, String Length, Special String Operations, Character Extraction, String Comparison, Searching Strings, Modifying a String, Data Conversion Using valueOf(), Changing the Case of Characters Within a String , Additional String Methods, StringBuffer, StringBuilder. Text book 1: Ch 12.1,12.2, Ch 13, Ch 15 RBT: L1, L2	08
Course outcomes: The students should be able to:	
<ul style="list-style-type: none"> • Explain the object-oriented concepts and JAVA. • Develop computer programs to solve real world problems in Java. Develop simple GUI interfaces for a computer program to interact with users	
Question Paper Pattern:	
<ul style="list-style-type: none"> • The question paper will have ten questions. • Each full Question consisting of 20 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. 	
Text Books:	
1. Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007. (Chapters 2, 3, 4, 5, 6,7, 8, 9,10, 12,13,15)	
Reference Books:	
1. Cay S Horstmann, "Core Java - Vol. 1 Fundamentals", Pearson Education, 10th Edition, 2016. 2. Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft, "Java 8 in Action", Dreamtech Press/Manning Press, 1st Edition, 2014.	

INTRODUCTION TO OPERATING SYSTEM (OPEN ELECTIVE) (Effective from the academic year 2018 -2019) SEMESTER – VII			
Course Code	18CS654	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Course Learning Objectives: This course (18CS654) will enable students to:			
<ul style="list-style-type: none"> • Explain the fundamentals of operating system • Comprehend multithreaded programming, process management, memory management and storage management. • Familiar with various types of operating systems 			
Module – 1			Teaching Hours
Introduction: What OS do, Computer system organization, architecture, structure, Operations, Process, memory and storage management, Protection and security, Distributed systems, Special purpose systems, computing environments. System Structure: OS Services, User OSI, System calls, Types of system calls, System programs, OS design and implementation, OS structure, Virtual machines, OS generation, system boot Textbook1: Chapter 1, 2 RBT: L1, L2			08
Module – 2			
Process Concept: Overview, Process scheduling, Operations on process, IPC, Examples in IPC, Communication in client-server systems. Multithreaded Programming: Overview, Models, Libraries, Issues, OS Examples Textbook1: Chapter 3,4 RBT: L1, L2			08
Module – 3			
Process Scheduling: Basic concept, Scheduling criteria, Algorithm, multiple processor scheduling, thread scheduling, OS Examples, Algorithm Evaluation. Synchronization: Background, the critical section problem, Petersons solution, Synchronization hardware, Semaphores, Classic problems of synchronization, Monitors, Synchronization examples, Atomic transactions Textbook1: Chapter 5, 6 RBT: L1, L2			08
Module – 4			
Deadlocks: System model, Deadlock characterization, Method of handling deadlock, Deadlock prevention, Avoidance, Detection, Recovery from deadlock Memory management strategies: Background, swapping, contiguous memory allocation, paging, structure of page table, segmentation, Textbook1: Chapter 7, 8 RBT: L1, L2			08
Module – 5			
Virtual Memory management: Background, Demand paging, Copy-on-write, Page replacement, allocation of frames, Trashing, Memory mapped files, Allocating Kernel memory, Operating system examples			08

File system: File concept, Access methods, Directory structure, File system mounting, File sharing, protection Textbook1: Chapter 9, 10 RBT: L1, L2	
Course outcomes: The students should be able to:	
<ul style="list-style-type: none"> • Explain the fundamentals of operating system • Comprehend process management, memory management and storage management. • Familiar with various types of operating systems 	
Question Paper Pattern:	
<ul style="list-style-type: none"> • The question paper will have ten questions. • Each full Question consisting of 20 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. 	
Text Books:	
1. A. Silberschatz, P B Galvin, G Gagne, Operating systems, 7 th edition, John Wiley and sons,.	
Reference Books:	
<ol style="list-style-type: none"> 1. William Stalling, "Operating Systems: Internals and Design Principles", Pearson Education, 1st Edition, 2018. 2. Andrew S Tanenbaum, Herbert BOS, "Modern Operating Systems", Pearson Education, 4th Edition, 2016 	

<p style="text-align: center;">SYSTEM SOFTWARE LABORATORY (Effective from the academic year 2018 -2019) SEMESTER – VI</p>			
Course Code	18CSL66	CIE Marks	40
Number of Contact Hours/Week	0:2:2	SEE Marks	60
Total Number of Lab Contact Hours	36	Exam Hours	03
Credits – 2			
Course Learning Objectives: This course (18CSL66) will enable students to:			
<ul style="list-style-type: none"> • To make students familiar with Lexical Analysis and Syntax Analysis phases of Compiler Design and implement programs on these phases using LEX & YACC tools and/or C/C++/Java • To enable students to learn different types of CPU scheduling algorithms used in operating system. • To make students able to implement memory management - page replacement and deadlock handling algorithms 			
Descriptions (if any):			
Exercises to be prepared with minimum three files (Where ever necessary):			
<ol style="list-style-type: none"> 1. Header file. 2. Implementation file. 3. Application file where main function will be present. 			
<p>The idea behind using three files is to differentiate between the developer and user sides. In the developer side, all the three files could be made visible. For the user side only header file and application files could be made visible, which means that the object code of the implementation file could be given to the user along with the interface given in the header file, hiding the source file, if required. Avoid I/O operations (printf/scanf) and use <i>data input file</i> where ever it is possible.</p>			
Programs List:			
Installation procedure of the required software must be demonstrated, carried out in groups and documented in the journal.			
1.			
a.	Write a LEX program to recognize valid <i>arithmetic expression</i> . Identifiers in the expression could be only integers and operators could be + and *. Count the identifiers & operators present and print them separately.		
b.	Write YACC program to evaluate <i>arithmetic expression</i> involving operators: +, -, *, and /		
2.	Develop, Implement and Execute a program using YACC tool to recognize all strings ending with <i>b</i> preceded by <i>n a</i> 's using the grammar <i>aⁿ b</i> (note: input <i>n</i> value)		
3.	Design, develop and implement YACC/C program to construct <i>Predictive / LL(1) Parsing Table</i> for the grammar rules: $A \rightarrow aBa$, $B \rightarrow bB \mid \epsilon$. Use this table to parse the sentence: <i>abba</i> \$		
4.	Design, develop and implement YACC/C program to demonstrate <i>Shift Reduce Parsing</i> technique for the grammar rules: $E \rightarrow E+T \mid T$, $T \rightarrow T * F \mid F$, $F \rightarrow (E) \mid id$ and parse the sentence: <i>id + id * id</i> .		
5.	Design, develop and implement a C/Java program to generate the machine code using <i>Triples</i> for the statement $A = -B * (C + D)$ whose intermediate code in three-address form:		
	$T1 = -B$ $T2 = C + D$ $T3 = T1 + T2$ $A = T3$		

6.	
a.	Write a LEX program to eliminate <i>comment lines</i> in a C program and copy the resulting program into a separate file.
b.	Write YACC program to recognize valid <i>identifier, operators and keywords</i> in the given text (C program) file.
7.	Design, develop and implement a C/C++/Java program to simulate the working of Shortest remaining time and Round Robin (RR) scheduling algorithms. Experiment with different quantum sizes for RR algorithm.
8.	Design, develop and implement a C/C++/Java program to implement Banker's algorithm. Assume suitable input required to demonstrate the results
9.	Design, develop and implement a C/C++/Java program to implement page replacement algorithms LRU and FIFO. Assume suitable input required to demonstrate the results.
Laboratory Outcomes: The student should be able to:	
<ul style="list-style-type: none"> • Implement and demonstrate Lexer's and Parser's • Evaluate different algorithms required for management, scheduling, allocation and communication used in operating system. 	
Conduct of Practical Examination:	
<ul style="list-style-type: none"> • Experiment distribution <ul style="list-style-type: none"> ○ For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity. ○ For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity. • Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only. • Marks Distribution (<i>Courseed to change in accordance with university regulations</i>) <ul style="list-style-type: none"> m) For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+70+15 = 100 Marks n) For laboratories having PART A and PART B <ul style="list-style-type: none"> i. Part A – Procedure + Execution + Viva = 6 + 28 + 6 = 40 Marks ii. Part B – Procedure + Execution + Viva = 9 + 42 + 9 = 60 Marks 	

COMPUTER GRAPHICS LABORATORY WITH MINI PROJECT (Effective from the academic year 2018 -2019) SEMESTER – VI			
Course Code	18CSL67	CIE Marks	40
Number of Contact Hours/Week	0:2:2	SEE Marks	60
Total Number of Lab Contact Hours	36	Exam Hours	03
Credits – 2			
Course Learning Objectives: This course (18CSL67) will enable students to:			
<ul style="list-style-type: none"> • Demonstrate simple algorithms using OpenGL Graphics Primitives and attributes. • Implementation of line drawing and clipping algorithms using OpenGL functions • Design and implementation of algorithms Geometric transformations on both 2D and 3D objects. 			
Descriptions (if any): --			
Installation procedure of the required software must be demonstrated, carried out in groups and documented in the journal.			
Programs List:			
PART A			
Design, develop, and implement the following programs using OpenGL API			
1.	Implement Brenham’s line drawing algorithm for all types of slope. Refer:Text-1: Chapter 3.5 Refer:Text-2: Chapter 8		
2.	Create and rotate a triangle about the origin and a fixed point. Refer:Text-1: Chapter 5-4		
3.	Draw a colour cube and spin it using OpenGL transformation matrices. Refer:Text-2: Modelling a Coloured Cube		
4.	Draw a color cube and allow the user to move the camera suitably to experiment with perspective viewing. Refer:Text-2: Topic: Positioning of Camera		
5.	Clip a lines using Cohen-Sutherland algorithm Refer:Text-1: Chapter 6.7 Refer:Text-2: Chapter 8		
6.	To draw a simple shaded scene consisting of a tea pot on a table. Define suitably the position and properties of the light source along with the properties of the surfaces of the solid object used in the scene. Refer:Text-2: Topic: Lighting and Shading		
7.	Design, develop and implement recursively subdivide a tetrahedron to form 3D sierpinski gasket. The number of recursive steps is to be specified by the user. Refer: Text-2: Topic: sierpinski gasket.		
8.	Develop a menu driven program to animate a flag using Bezier Curve algorithm Refer: Text-1: Chapter 8-10		
9.	Develop a menu driven program to fill the polygon using scan line algorithm		
PART B MINI PROJECT			
Student should develop mini project on the topics mentioned below or similar applications using Open GL API. Consider all types of attributes like color, thickness, styles, font, background, speed etc., while doing mini project. (During the practical exam: the students should demonstrate and answer Viva-Voce)			
Sample Topics:			
Simulation of concepts of OS, Data structures, algorithms etc.			
Laboratory Outcomes: The student should be able to:			
<ul style="list-style-type: none"> • Apply the concepts of computer graphics 			

- Implement computer graphics applications using OpenGL
- Animate real world problems using OpenGL

Conduct of Practical Examination:

- Experiment distribution
 - For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
 - For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.
- Marks Distribution (*Courseed to change in accordance with university regulations*)
 - o) For laboratories having only one part – Procedure + Execution + Viva-Voce: $15+70+15 = 100$ Marks
 - p) For laboratories having PART A and PART B
 - i. Part A – Procedure + Execution + Viva = $6 + 28 + 6 = 40$ Marks
 - ii. Part B – Procedure + Execution + Viva = $9 + 42 + 9 = 60$ Marks

MOBILE APPLICATION DEVELOPMENT
(Effective from the academic year 2018 -2019)
SEMESTER – VI

Course Code	18CSMP68	IA Marks	40
Number of Contact Hours/Week	0:0:2	Exam Marks	60
Total Number of Contact Hours	3 Hours/Week	Exam Hours	03

CREDITS – 02

Laboratory Objectives:This laboratory (18CSMP68) will enable students to

- Learn and acquire the art of Android Programming.
- Configure Android studio to run the applications.
- Understand and implement Android's User interface functions.
- Create, modify and query on SQLite database.
- Inspect different methods of sharing data using services.

Descriptions (if any):


Installation procedure of the Android Studio/Java software must be demonstrated, carried out in groups.

Students should use the latest version of Android Studio/Java to execute these programs.

All of these diagrams are for representational purpose only. Students are expected to improvise on it.

Programs List:

PART – A

1	<p>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.</p> <div style="text-align: center;">  </div>
2	<p>Develop an Android application using controls like Button, TextView, EditText for designing a calculator having basic functionality like Addition, Subtraction, Multiplication, and Division.</p>

SIMPLE CALCULATOR

Result

Input <Edit Text>

7	8	9	/
4	5	6	*
1	2	3	-
.	0	=	+
C			

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 lowercase 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.

SIGNUP ACTIVITY

Username:

Password:

SIGN UP

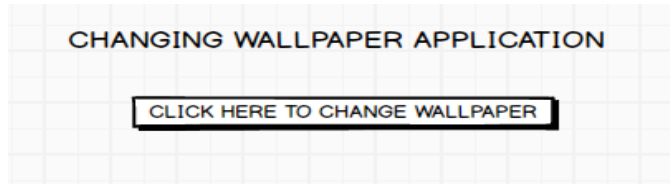
LOGIN ACTIVITY

Username:

Password:

SIGN IN

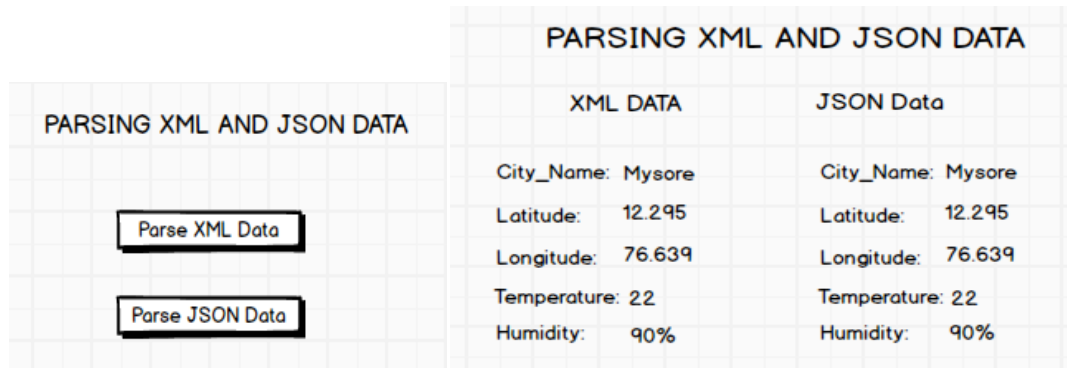
4 Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.



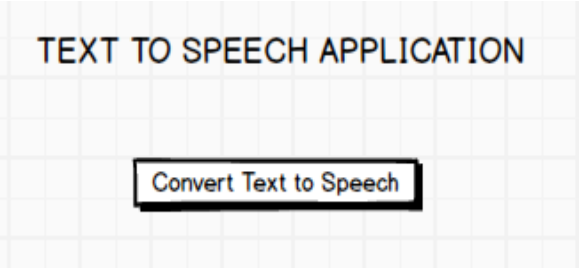
5 Write a program to create an activity with two buttons START and STOP. On pressing of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextView control.



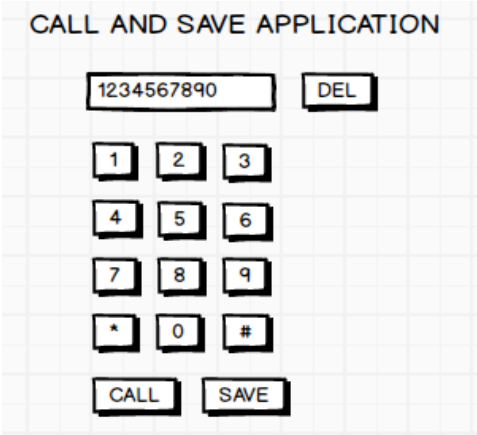
6 Create two files of XML and JSON type with values for City_Name, Latitude, Longitude, Temperature, and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.



7 Develop a simple application with one EditText 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.

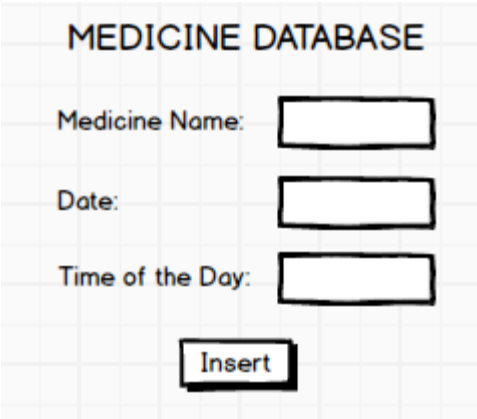


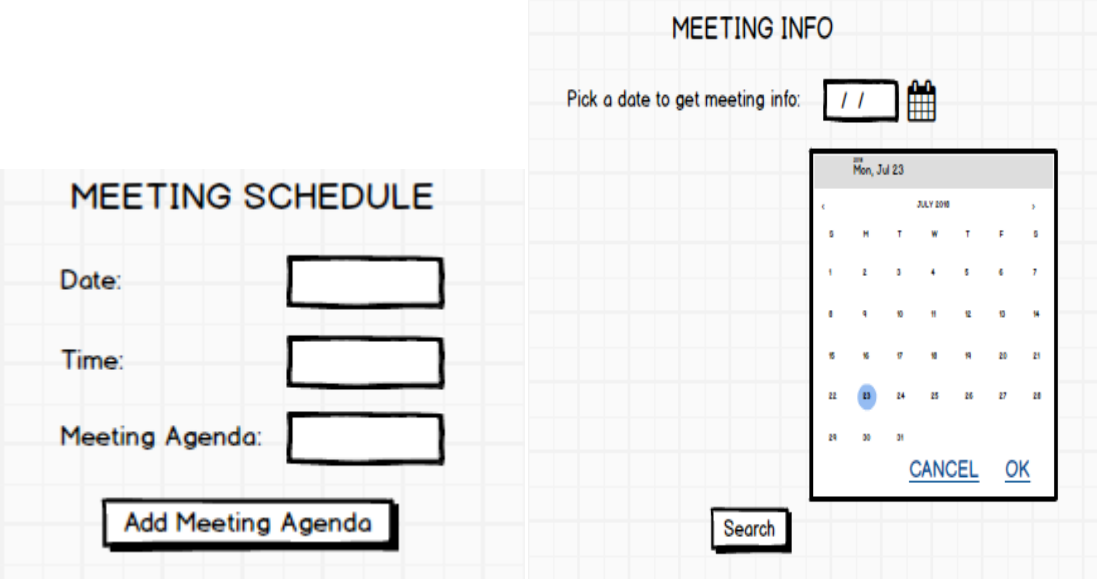
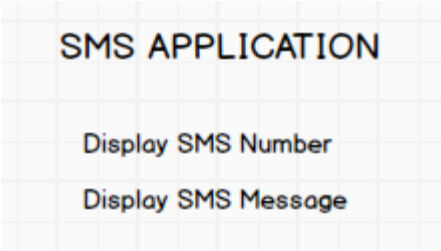
8 Create an activity like a phone dialer with CALL and SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phone contacts.

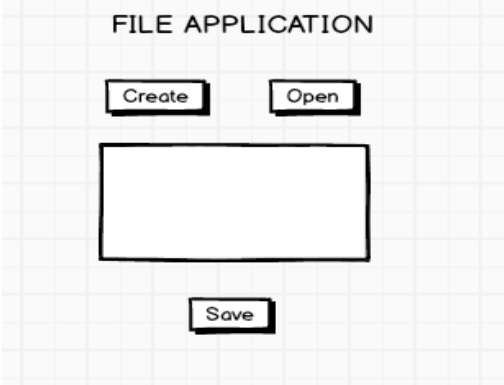
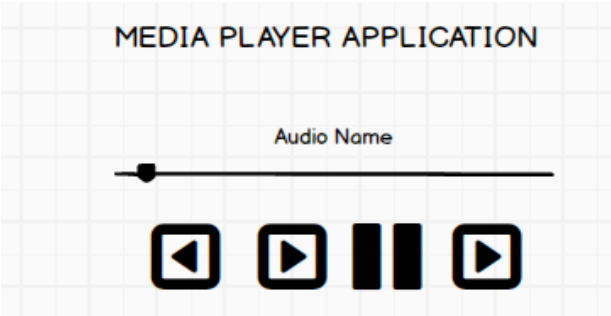




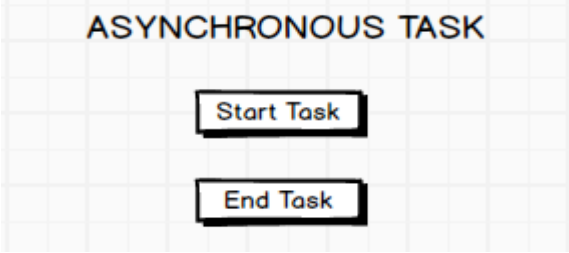


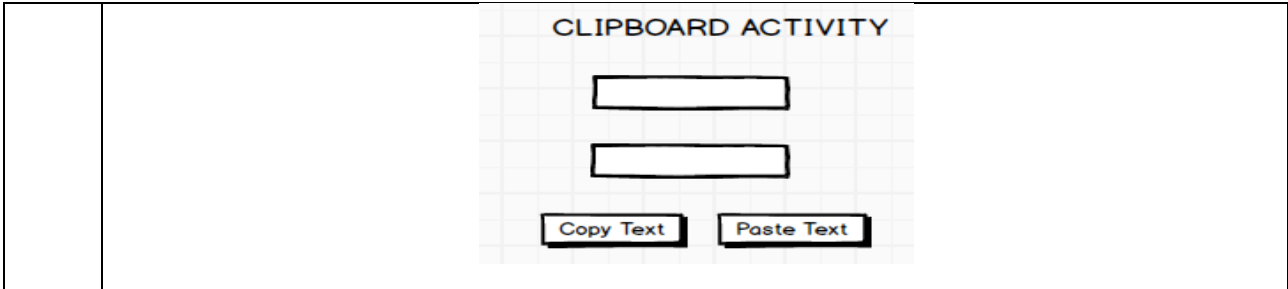
PART - B

1 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.



<p>2</p>	<p>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 DatePicker 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”.</p>  <p>The image shows two screenshots of an Android application. The left screenshot, titled "MEETING SCHEDULE", features three input fields labeled "Date:", "Time:", and "Meeting Agenda:", each with a corresponding empty text box. Below these fields is a button labeled "Add Meeting Agenda". The right screenshot, titled "MEETING INFO", shows a date picker interface. It includes the text "Pick a date to get meeting info:" followed by a date input field containing " / /" and a calendar icon. A calendar is displayed below, showing the month of July 2018 with the 23rd selected. At the bottom of the calendar are "CANCEL" and "OK" buttons. Below the calendar is a "Search" button.</p>
<p>3</p>	<p>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.</p>  <p>The image shows a screenshot of an Android application titled "SMS APPLICATION". It contains two buttons: "Display SMS Number" and "Display SMS Message".</p>
<p>4</p>	<p>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”.</p>

	 <p style="text-align: center;">FILE APPLICATION</p> <p style="text-align: center;">Create Open</p> <div style="border: 1px solid black; width: 150px; height: 50px; margin: 10px auto;"></div> <p style="text-align: center;">Save</p>
<p>5</p>	<p>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.</p>  <p style="text-align: center;">MEDIA PLAYER APPLICATION</p> <p style="text-align: center;">Audio Name</p> <div style="text-align: center;">  </div> <div style="text-align: center; margin-top: 10px;">    </div>
<p>6</p>	<p>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”.</p>  <p style="text-align: center;">ASYNCHRONOUS TASK</p> <p style="text-align: center;">Start Task</p> <p style="text-align: center;">End Task</p>
<p>7</p>	<p>Develop an application that makes use of the clipboard framework for copying and pasting of the text. The activity consists of two EditText controls and two Buttons to trigger the copy and paste functionality.</p>



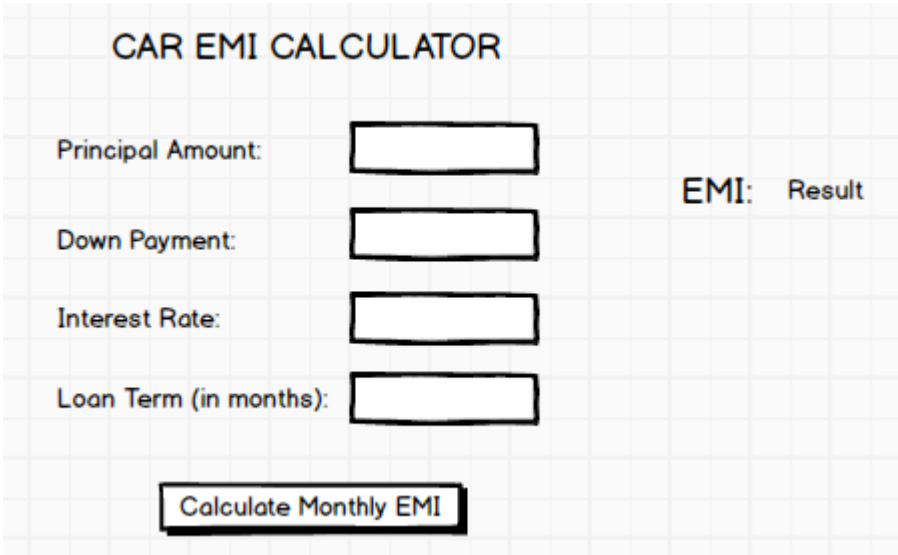
8 Create an AIDL service that calculates Car Loan EMI. The formula to calculate EMI is

$$E = P * (r(1+r)^n)/((1+r)^n-1)$$

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 EditText to read the PrincipalAmount, 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 TextView. Also, calculate the EMI by varying the Loan Term and Interest Rate values.



Laboratory Outcomes:After studying these laboratory programs, students will be able to

- Create, test and debug Android application by setting up Android development environment.
- Implement adaptive, responsive user interfaces that work across a wide range of devices.
- Infer long running tasks and background work in Android applications.
- Demonstrate methods in storing, sharing and retrieving data in Android applications.

- Infer the role of permissions and security for Android applications.

Procedure to Conduct Practical Examination

- Experiment distribution
 - For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
 - For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.
- Marks Distribution (Course to change in accordance with university regulations)
 - For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+70+15= 100 Marks
 - For laboratories having PART A and PART B
 - i. Part A – Procedure + Execution + Viva = 6 + 28 + 6 = 40 Marks
 - ii. Part B – Procedure + Execution + Viva = 9 + 42 + 9 = 60 Marks

Text Books:

1. Google Developer Training, "**Android Developer Fundamentals Course – Concept Reference**", 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**", 1st Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197
2. Dawn Griffiths and David Griffiths, "**Head First Android Development**", 1st 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**", 3rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054