

<b>SYSTEM SOFTWARE AND COMPILERS</b> <b>(Effective from the academic year 2018 -2019)</b> <b>SEMESTER – VI</b>			
<b>Course Code</b>	<b>18CS61</b>	<b>CIE Marks</b>	40
<b>Number of Contact Hours/Week</b>	3:2:0	<b>SEE Marks</b>	60
<b>Total Number of Contact Hours</b>	50	<b>Exam Hours</b>	03
<b>CREDITS –4</b>			
<b>Course Learning Objectives:</b> This course (18CS61) will enable students to: <ul style="list-style-type: none"> <li>• Define System Software.</li> <li>• Familiarize with source file, object file and executable file structures and libraries</li> <li>• Describe the front-end and back-end phases of compiler and their importance to students</li> </ul>			
<b>Module 1</b>			<b>Contact Hours</b>
Introduction to System Software, Machine Architecture of SIC and SIC/XE. <b>Assemblers:</b> Basic assembler functions, machine dependent assembler features, machine independent assembler features, assembler design options. Basic Loader Functions <b>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</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 2</b>			
<b>Introduction:</b> Language Processors, The structure of a compiler, The evaluation of programming languages, The science of building compiler, Applications of compiler technology. <b>Lexical Analysis:</b> The role of lexical analyzer, Input buffering, Specifications of token, recognition of tokens. <b>Text book 2:Chapter 1 1.1-1.5 Chapter 3: 3.1 – 3.4</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 3</b>			
Syntax Analysis: Introduction, Context Free Grammars, Writing a grammar, Top Down Parsers, Bottom-Up Parsers <b>Text book 2: Chapter 4 4.1, 4.2 4.3 4.4 4.5</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 4</b>			
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. <b>Text book 3: Chapter 1,2 and 3.</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 5</b>			
Syntax Directed Translation, Intermediate code generation, Code generation <b>Text book 2: Chapter 5.1, 5.2, 5.3, 6.1, 6.2, 8.1, 8.2</b> <b>RBT: L1, L2, L3</b>			10
<b>Course Outcomes:</b> The student will be able to : <ul style="list-style-type: none"> <li>• Explain system software</li> <li>• Design and develop lexical analyzers, parsers and code generators</li> <li>• Utilize lex and yacc tools for implementing different concepts of system software</li> </ul>			

<b>Question Paper Pattern:</b>
<ul style="list-style-type: none"> <li>• The question paper will have ten questions.</li> <li>• Each full Question consisting of 20 marks</li> <li>• There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>• Each full question will have sub questions covering all the topics under a module.</li> <li>• The students will have to answer 5 full questions, selecting one full question from each module.</li> </ul>
<b>Textbooks:</b>
<ol style="list-style-type: none"> <li>1. System Software by Leland. L. Beck, D Manjula, 3<sup>rd</sup> edition, 2012</li> <li>2. Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman , Compilers-Principles, Techniques and Tools, Pearson, 2<sup>nd</sup> edition, 2007</li> <li>3. Doug Brown, John Levine, Tony Mason, lex &amp; yacc, O'Reilly Media, October 2012.</li> </ol>
<b>Reference Books:</b>
<ol style="list-style-type: none"> <li>1. Systems programming – Srimanta Pal , Oxford university press, 2016</li> <li>2. System programming and Compiler Design, K C Loudon, Cengage Learning</li> <li>3. System software and operating system by D. M. Dhamdhere TMG</li> <li>4. Compiler Design, K Muneeswaran, Oxford University Press 2013.</li> </ol>

<b>COMPUTER GRAPHICS AND VISUALIZATION</b> <b>(Effective from the academic year 2018 -2019)</b> <b>SEMESTER – VI</b>			
<b>Course Code</b>	<b>18CS62</b>	<b>CIE Marks</b>	40
<b>Number of Contact Hours/Week</b>	3:2:0	<b>SEE Marks</b>	60
<b>Total Number of Contact Hours</b>	50	<b>Exam Hours</b>	03
<b>CREDITS –4</b>			
<b>Course Learning Objectives:</b> This course (18CS62) will enable students to: <ul style="list-style-type: none"> <li>• Explain hardware, software and OpenGL Graphics Primitives.</li> <li>• Illustrate interactive computer graphic using the OpenGL.</li> <li>• Design and implementation of algorithms for 2D graphics Primitives and attributes.</li> <li>• Demonstrate Geometric transformations, viewing on both 2D and 3D objects.</li> <li>• Infer the representation of curves, surfaces, Color and Illumination models</li> </ul>			
<b>Module 1</b>			<b>Contact Hours</b>
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). <b>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</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 2</b>			
<b>Fill area Primitives, 2D Geometric Transformations and 2D viewing:</b> 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. <b>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</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 3</b>			
<b>Clipping,3D Geometric Transformations, Color and Illumination Models:</b> 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. <b>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</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 4</b>			
<b>3D Viewing and Visible Surface Detection:</b> 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><b>Text-1:Chapter: 7-1 to 7-10(Excluding 7-7), 9-1,9-3, 9-14</b></p> <p><b>RBT: L1, L2, L3</b></p>	
<b>Module 5</b>	
<p><b>Input&amp; interaction, Curves and Computer Animation:</b> 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><b>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</b></p> <p><b>Text-2:Chapter 3: 3-1 to 3.11: Input&amp; interaction</b></p> <p><b>RBT: L1, L2, L3</b></p>	10
<b>Course Outcomes:</b> The student will be able to :	
<ul style="list-style-type: none"> <li>• Design and implement algorithms for 2D graphics primitives and attributes.</li> <li>• Illustrate Geometric transformations on both 2D and 3D objects.</li> <li>• Apply concepts of clipping and visible surface detection in 2D and 3D viewing, and Illumination Models.</li> <li>• Decide suitable hardware and software for developing graphics packages using OpenGL.</li> </ul>	
<b>Question Paper Pattern:</b>	
<ul style="list-style-type: none"> <li>• The question paper will have ten questions.</li> <li>• Each full Question consisting of 20 marks</li> <li>• There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>• Each full question will have sub questions covering all the topics under a module.</li> <li>• The students will have to answer 5 full questions, selecting one full question from each module.</li> </ul>	
<b>Textbooks:</b>	
<ol style="list-style-type: none"> <li>1. Donald Hearn &amp; Pauline Baker: Computer Graphics with OpenGL Version,3<sup>rd</sup> / 4<sup>th</sup> Edition, Pearson Education,2011</li> <li>2. Edward Angel: Interactive Computer Graphics- A Top Down approach with OpenGL, 5<sup>th</sup> edition. Pearson Education, 2008</li> </ol>	
<b>Reference Books:</b>	
<ol style="list-style-type: none"> <li>1. James D Foley, Andries Van Dam, Steven K Feiner, John F Huges Computer graphics with OpenGL: pearson education</li> <li>2. Xiang, Plastock : Computer Graphics , sham's outline series, 2<sup>nd</sup> edition, TMG.</li> <li>3. Kelvin Sung, Peter Shirley, steven Baer : Interactive Computer Graphics, concepts and applications, Cengage Learning</li> <li>4. M M Raikar &amp; Shreedhara K S Computer Graphics using OpenGL, Cengage publication</li> </ol>	

<b>WEB TECHNOLOGY AND ITS APPLICATIONS</b> <b>(Effective from the academic year 2018 -2019)</b> <b>SEMESTER – VI</b>			
<b>Course Code</b>	<b>18CS63</b>	<b>CIE Marks</b>	40
<b>Number of Contact Hours/Week</b>	3:2:0	<b>SEE Marks</b>	60
<b>Total Number of Contact Hours</b>	50	<b>Exam Hours</b>	03
<b>CREDITS –4</b>			
<b>Course Learning Objectives:</b> This course (18CS63) will enable students to:			
<ul style="list-style-type: none"> <li>• Illustrate the Semantic Structure of HTML and CSS</li> <li>• Compose forms and tables using HTML and CSS</li> <li>• Design Client-Side programs using JavaScript and Server-Side programs using PHP</li> <li>• Infer Object Oriented Programming capabilities of PHP</li> <li>• Examine JavaScript frameworks such as jQuery and Backbone</li> </ul>			
<b>Module 1</b>			<b>Contact Hours</b>
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. <b>Textbook 1: Ch. 2, 3</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 2</b>			
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. <b>Textbook 1: Ch. 4,5</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 3</b>			
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 <b>Textbook 1: Ch. 6, 8</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 4</b>			
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 <b>Textbook 1: Ch. 9, 10</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 5</b>			
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. <b>Textbook 1: Ch. 13, 15,17</b> <b>RBT: L1, L2, L3</b>	
<b>Course Outcomes:</b> The student will be able to :	
<ul style="list-style-type: none"> <li>Adapt HTML and CSS syntax and semantics to build web pages.</li> <li>Construct and visually format tables and forms using HTML and CSS</li> <li>Develop Client-Side Scripts using JavaScript and Server-Side Scripts using PHP to generate and display the contents dynamically.</li> <li>Appraise the principles of object oriented development using PHP</li> <li>Inspect JavaScript frameworks like jQuery and Backbone which facilitates developer to focus on core features.</li> </ul>	
<b>Question Paper Pattern:</b>	
<ul style="list-style-type: none"> <li>The question paper will have ten questions.</li> <li>Each full Question consisting of 20 marks</li> <li>There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>Each full question will have sub questions covering all the topics under a module.</li> <li>The students will have to answer 5 full questions, selecting one full question from each module.</li> </ul>	
<b>Textbooks:</b>	
1. Randy Connolly, Ricardo Hoar, " <b>Fundamentals of Web Development</b> ", 1 <sup>st</sup> Edition, Pearson Education India. (ISBN:978-9332575271)	
<b>Reference Books:</b>	
1. Robin Nixon, " <b>Learning PHP, MySQL &amp; JavaScript with jQuery, CSS and HTML5</b> ", 4 <sup>th</sup> Edition, O'Reilly Publications, 2015. (ISBN:978-9352130153) 2. Luke Welling, Laura Thomson, " <b>PHP and MySQL Web Development</b> ", 5 <sup>th</sup> Edition, Pearson Education, 2016. (ISBN:978-9332582736) 3. Nicholas C Zakas, " <b>Professional JavaScript for Web Developers</b> ", 3 <sup>rd</sup> Edition, Wrox/Wiley India, 2012. (ISBN:978-8126535088) 4. David Sawyer Mcfarland, " <b>JavaScript &amp; jQuery: The Missing Manual</b> ", 1 <sup>st</sup> Edition, O'Reilly/Shroff Publishers & Distributors Pvt Ltd, 2014	
<b>Mandatory Note:</b>	
Distribution of CIE Marks is as follows (Total 40 Marks): <ul style="list-style-type: none"> <li>20 Marks through IA Tests</li> <li>20 Marks through practical assessment</li> </ul>	
<b>Maintain a copy of the report for verification during LIC visit.</b>	
<b>Possible list of practicals:</b>	
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: <ul style="list-style-type: none"> <li>a. Parameter: A string</li> <li>b. Output: The position in the string of the left-most vowel</li> </ul>	

- 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.

<b>DATA MINING AND DATA WAREHOUSING</b> <b>(Effective from the academic year 2018 -2019)</b> <b>SEMESTER – VI</b>			
<b>Course Code</b>	<b>18CS641</b>	<b>CIE Marks</b>	40
<b>Number of Contact Hours/Week</b>	3:0:0	<b>SEE Marks</b>	60
<b>Total Number of Contact Hours</b>	40	<b>Exam Hours</b>	03
<b>CREDITS –3</b>			
<b>Course Learning Objectives:</b> This course (18CS641) will enable students to: <ul style="list-style-type: none"> <li>• Define multi-dimensional data models.</li> <li>• Explain rules related to association, classification and clustering analysis.</li> <li>• Compare and contrast between different classification and clustering algorithms</li> </ul>			
<b>Module 1</b>			<b>Contact Hours</b>
<b>Data Warehousing &amp; modeling:</b> 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 <b>Textbook 2: Ch.4.1,4.2</b> <b>RBT: L1, L2, L3</b>			08
<b>Module 2</b>			
<b>Data warehouse implementation&amp; Data mining:</b> 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. <b>Textbook 2: Ch.4.4</b> <b>Textbook 1: Ch.1.1,1.2,1.4, 2.1 to 2.4</b> <b>RBT: L1, L2, L3</b>			08
<b>Module 3</b>			
<b>Association Analysis:</b> Association Analysis: Problem Definition, Frequent Item set Generation, Rule generation. Alternative Methods for Generating Frequent Item sets, FP-Growth Algorithm, Evaluation of Association Patterns. <b>Textbook 1: Ch 6.1 to 6.7 (Excluding 6.4)</b> <b>RBT: L1, L2, L3</b>			08
<b>Module 4</b>			
<b>Classification:</b> Decision Trees Induction, Method for Comparing Classifiers, Rule Based Classifiers, Nearest Neighbor Classifiers, Bayesian Classifiers. <b>Textbook 1: Ch 4.3,4.6,5.1,5.2,5.3</b> <b>RBT: L1, L2, L3</b>			08
<b>Module 5</b>			
<b>Clustering Analysis:</b> Overview, K-Means, Agglomerative Hierarchical Clustering, DBSCAN, Cluster Evaluation, Density-Based Clustering, Graph-Based Clustering, Scalable Clustering Algorithms. <b>Textbook 1: Ch 8.1 to 8.5, 9.3 to 9.5</b> <b>RBT: L1, L2, L3</b>			08
<b>Course Outcomes:</b> 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, 3<sup>rd</sup> 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 edition, 2012.

<b>CLOUD COMPUTING AND ITS APPLICATIONS</b> <b>(Effective from the academic year 2018 -2019)</b> <b>SEMESTER – VI</b>			
<b>Course Code</b>	<b>18CS643</b>	<b>CIE Marks</b>	40
<b>Number of Contact Hours/Week</b>	3:0:0	<b>SEE Marks</b>	60
<b>Total Number of Contact Hours</b>	40	<b>Exam Hours</b>	03
<b>CREDITS –3</b>			
<b>Course Learning Objectives:</b> This course (18CS643) will enable students to: <ul style="list-style-type: none"> <li>• Explain the fundamentals of cloud computing</li> <li>• Illustrate the cloud application programming and aneka platform</li> <li>• Contrast different cloud platforms used in industry</li> </ul>			
<b>Module 1</b>			<b>Contact Hours</b>
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 <b>Textbook 1: Ch. 1,3</b> <b>RBT: L1, L2</b>			08
<b>Module 2</b>			
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 <b>Textbook 1: Ch. 4,5</b> <b>RBT: L1, L2</b>			08
<b>Module 3</b>			
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

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. <b>Textbook 1: Ch. 6, 7</b> <b>RBT: L1, L2</b>	
<b>Module 4</b>	
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 <b>Textbook 1: Ch. 8</b> <b>RBT: L1, L2</b>	08
<b>Module 5</b>	
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. 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. <b>Textbook 1: Ch. 9,10</b> <b>RBT: L1, L2</b>	08
<b>Course Outcomes:</b> The student will be able to :	
<ul style="list-style-type: none"> <li>• Explain cloud computing, virtualization and classify services of cloud computing</li> <li>• Illustrate architecture and programming in cloud</li> <li>• Describe the platforms for development of cloud applications and List the application of cloud.</li> </ul>	
<b>Question Paper Pattern:</b>	
<ul style="list-style-type: none"> <li>• The question paper will have ten questions.</li> <li>• Each full Question consisting of 20 marks</li> <li>• There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>• Each full question will have sub questions covering all the topics under a module.</li> <li>• The students will have to answer 5 full questions, selecting one full question from each module.</li> </ul>	
<b>Textbooks:</b>	
1. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi Mastering Cloud. Computing McGraw Hill Education	
<b>Reference Books:</b>	
1. Dan C. Marinescu, Cloud Computing Theory and Practice, Morgan Kaufmann, Elsevier 2013.	

<b>SYSTEM SOFTWARE LABORATORY</b> <b>(Effective from the academic year 2018 -2019)</b> <b>SEMESTER – VI</b>			
<b>Course Code</b>	<b>18CSL66</b>	<b>CIE Marks</b>	40
<b>Number of Contact Hours/Week</b>	0:2:2	<b>SEE Marks</b>	60
<b>Total Number of Lab Contact Hours</b>	36	<b>Exam Hours</b>	03
<b>Credits – 2</b>			
<b>Course Learning Objectives:</b> This course (18CSL66) will enable students to: <ul style="list-style-type: none"> <li>To make students familiar with Lexical Analysis and Syntax Analysis phases of Compiler Design and implement programs on these phases using LEX &amp; YACC tools and/or C/C++/Java</li> <li>To enable students to learn different types of CPU scheduling algorithms used in operating system.</li> <li>To make students able to implement memory management - page replacement and deadlock handling algorithms</li> </ul>			
<b>Descriptions (if any):</b> Exercises to be prepared with minimum three files (Where ever necessary): <ol style="list-style-type: none"> <li>Header file.</li> <li>Implementation file.</li> <li>Application file where main function will be present.</li> </ol> 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.			
<b>Programs List:</b>			
<b>Installation procedure of the required software must be demonstrated, carried out in groups and documented in the journal.</b>			
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's</i> using the grammar <i>a<sup>n</sup> 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: <i>A → aBa</i> , <i>B → bB / ε</i> . 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: <i>E → E+T / T</i> , <i>T → T*F / F</i> , <i>F → (E) / id</i> 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 <i>A = -B * (C +D)</i> 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.
<b>Laboratory Outcomes:</b> The student should be able to:	
<ul style="list-style-type: none"> <li>• Implement and demonstrate Lexer's and Parser's</li> <li>• Evaluate different algorithms required for management, scheduling, allocation and communication used in operating system.</li> </ul>	
<b>Conduct of Practical Examination:</b>	
<ul style="list-style-type: none"> <li>• Experiment distribution <ul style="list-style-type: none"> <li>○ For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.</li> <li>○ 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.</li> </ul> </li> <li>• Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.</li> <li>• Marks Distribution (<i>Courseed to change in accordance with university regulations</i>) <ul style="list-style-type: none"> <li>m) For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+70+15 = 100 Marks</li> <li>n) For laboratories having PART A and PART B <ul style="list-style-type: none"> <li>i. Part A – Procedure + Execution + Viva = 6 + 28 + 6 = 40 Marks</li> <li>ii. Part B – Procedure + Execution + Viva = 9 + 42 + 9 = 60 Marks</li> </ul> </li> </ul> </li> </ul>	

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			
<b>Course Learning Objectives:</b> This course (18CSL67) will enable students to:			
<ul style="list-style-type: none"><li>• Demonstrate simple algorithms using OpenGL Graphics Primitives and attributes.</li><li>• Implementation of line drawing and clipping algorithms using OpenGL functions</li><li>• Design and implementation of algorithms Geometric transformations on both 2D and 3D objects.</li></ul>			
<b>Descriptions (if any): --</b>			
<b>Installation procedure of the required software must be demonstrated, carried out in groups and documented in the journal.</b>			
<b>Programs List:</b>			
<b>PART A</b>			
<b>Design, develop, and implement the following programs using OpenGL API</b>			
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. <b>Refer:Text-1: Chapter 5-4</b>		
3.	Draw a colour cube and spin it using OpenGL transformation matrices. <b>Refer:Text-2: Modelling a Coloured Cube</b>		
4.	Draw a color cube and allow the user to move the camera suitably to experiment with perspective viewing. <b>Refer:Text-2: Topic: Positioning of Camera</b>		
5.	Clip a lines using Cohen-Sutherland algorithm <b>Refer:Text-1: Chapter 6.7</b> <b>Refer:Text-2: Chapter 8</b>		
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. <b>Refer:Text-2: Topic: Lighting and Shading</b>		
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. <b>Refer: Text-2: Topic: sierpinski gasket.</b>		
8.	Develop a menu driven program to animate a flag using Bezier Curve algorithm <b>Refer: Text-1: Chapter 8-10</b>		
9.	Develop a menu driven program to fill the polygon using scan line algorithm		
<b>PART B MINI PROJECT</b>			
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. <b>(During the practical exam: the students should demonstrate and answer Viva-Voce)</b>			
<b>Sample Topics:</b>			
<b>Simulation of concepts of OS, Data structures, algorithms etc.</b>			
<b>Laboratory Outcomes:</b> The student should be able to:			
<ul style="list-style-type: none"><li>• Apply the concepts of computer graphics</li></ul>			

- 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**

<b>Course Code</b>	<b>18CSMP68</b>	<b>IA Marks</b>	40
<b>Number of Contact Hours/Week</b>	0:0:2	<b>Exam Marks</b>	60
<b>Total Number of Contact Hours</b>	3 Hours/Week	<b>Exam Hours</b>	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):**

1. The installation procedure of the Android Studio/Java software must be demonstrated and carried out in groups.
2. Students should use the latest version of Android Studio/Java/ Kotlin to execute these programs. Diagrams given are for representational purposes only, students are expected to improvise on them.
3. **Part B programs should be developed as an application and are to be demonstrated as a mini project in a group by adding extra features or the students can also develop their application and demonstrate it as a mini-project. (Projects/programs are not limited to the list given in Part B).**

**Programs List:**

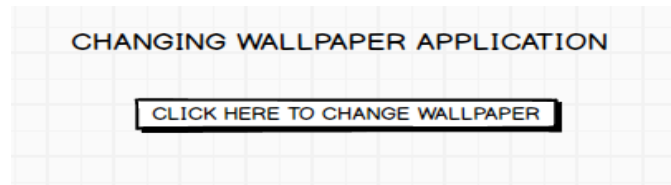
**PART – A**

<b>1</b>	<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; margin-top: 20px;">  </div>
<b>2</b>	<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>



	<div><h3>SIMPLE CALCULATOR</h3><p>Result</p><div>Input &lt;Edit Text&gt;</div><div><div>7</div><div>8</div><div>9</div><div>/</div></div><div><div>4</div><div>5</div><div>6</div><div>*</div></div><div><div>1</div><div>2</div><div>3</div><div>-</div></div><div><div>.</div><div>0</div><div>=</div><div>+</div></div><div>C</div></div>
3	<p>Create a <b>SIGN Up</b> activity with Username and Password. Validation of password should happen based on the following rules:</p> <ul style="list-style-type: none"><li>• Password should contain uppercase and lowercase letters.</li><li>• Password should contain letters and numbers.</li><li>• Password should contain special characters.</li><li>• Minimum length of the password (the default value is 8).</li></ul> <p>On successful <b>SIGN UP</b> proceed to the next Login activity. Here the user should <b>SIGN IN</b> 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.</p> <div><div><h3>SIGNUP ACTIVITY</h3><div>Username: <input type="text"/></div><div>Password: <input type="password"/></div><div>SIGN UP</div></div><div><h3>LOGIN ACTIVITY</h3><div>Username: <input type="text"/></div><div>Password: <input type="password"/></div><div>SIGN IN</div></div></div>

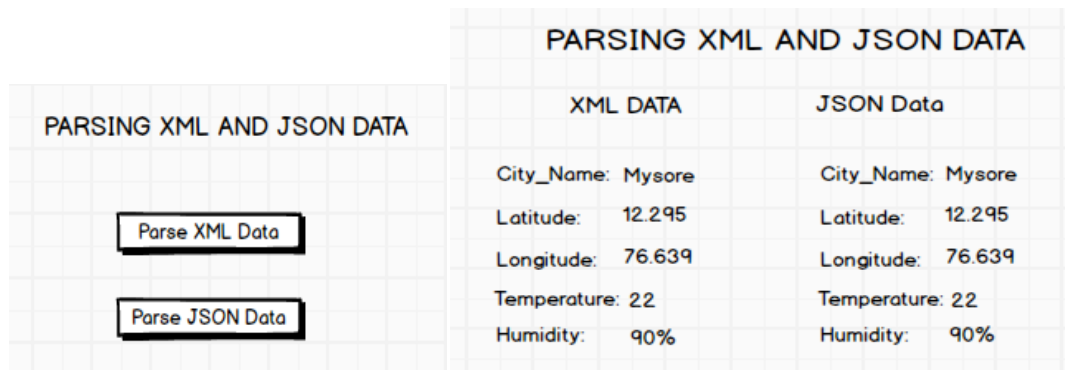
- 4 Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.

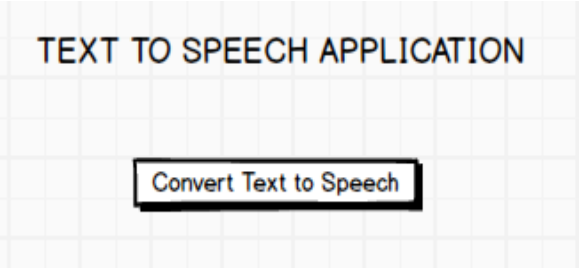
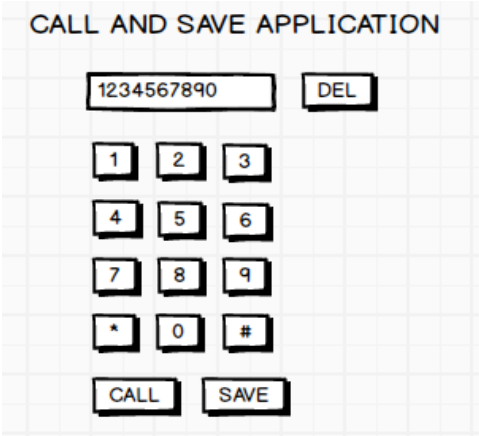
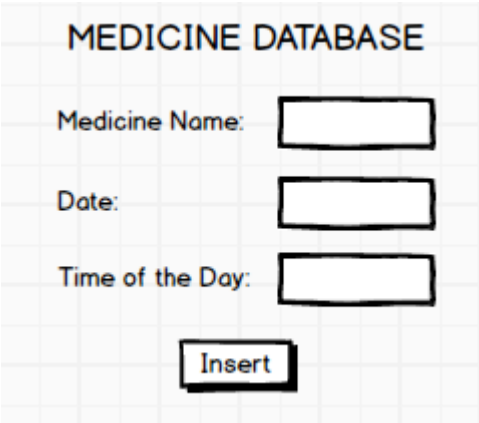


- 5 Write a program to create an activity with two buttons START and STOP. On pressing of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextView control.

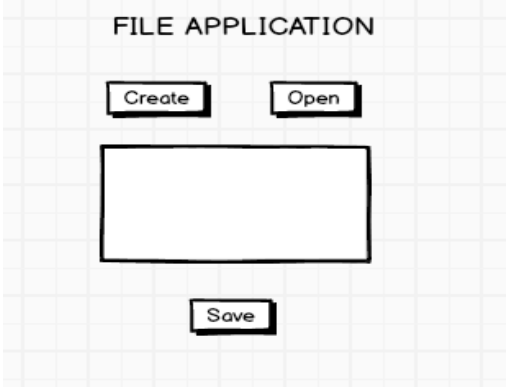
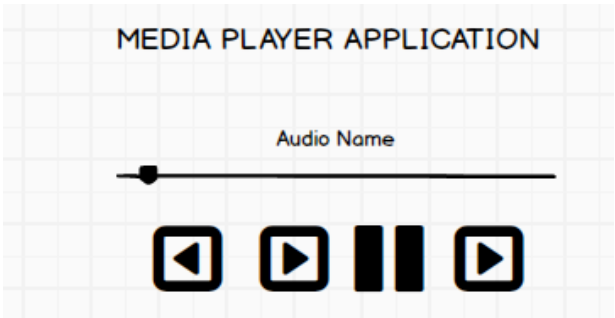
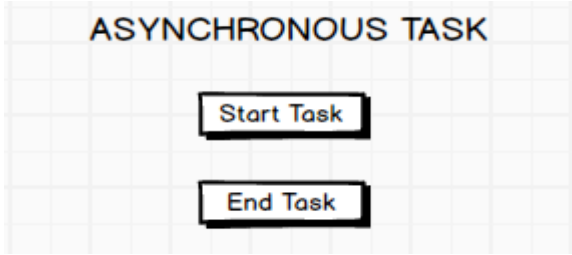


- 6 Create two files of XML and JSON type with values for City\_Name, Latitude, Longitude, Temperature, and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.



7	<p>Develop a simple application with one <code>EditText</code> 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.</p> 
8	<p>Create an activity like a phone dialer with <code>CALL</code> and <code>SAVE</code> buttons. On pressing the <code>CALL</code> button, it must call the phone number and on pressing the <code>SAVE</code> button it must save the number to the phone contacts.</p> 
<b>PART - B</b>	
1	<p>Write a program to enter Medicine Name, Date and Time of the Day as input from the user and store it in the <code>SQLite</code> 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>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> <div data-bbox="284 405 1378 982"> <p>The image shows two screenshots of Android applications. The left screenshot, titled 'MEETING SCHEDULE', features three text input fields labeled 'Date:', 'Time:', and 'Meeting Agenda:', each followed by a rectangular 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 format ' / /' and a calendar icon. Below this is a calendar for 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> </div>
3	<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> <div data-bbox="638 1205 1076 1453"> <p>The image shows a screenshot of an application titled 'SMS APPLICATION'. It contains two text labels: 'Display SMS Number' and 'Display SMS Message', each followed by a rectangular text box.</p> </div>
4	<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>

	
5	<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> 
6	<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 <b>Start Task</b> button, the banner message should scroll from right to left. On pressing the <b>Stop Task</b> button, the banner message should stop. Let the banner message be “Demonstration of Asynchronous Task”.</p> 
7	<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>

	<div data-bbox="641 191 1071 447" data-label="Image"> </div>
8	<p>Create an AIDL service that calculates Car Loan EMI. The formula to calculate EMI is</p> $E = P * (r(1+r)^n)/((1+r)^n-1)$ <p>where</p> <p>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</p> <p>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.</p> <div data-bbox="410 1037 1297 1589" data-label="Image"> </div>
	<p><b>Laboratory Outcomes:</b>After studying theselaboratory programs, students will be able to</p> <ul style="list-style-type: none"> <li>• Create, test and debug Android application by setting up Android development environment.</li> <li>• Implement adaptive, responsive user interfaces that work across a wide range of devices.</li> <li>• Infer long running tasks and background work in Android applications.</li> <li>• Demonstrate methods in storing, sharing and retrieving data in Android applications.</li> </ul>

- 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 (Courseed 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**", 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