

**Semester III (Second year)**

Sl.	Category	Course Code	Course Title	Hours per week			Credits	CIE	SEE	TOTAL
				L	T	P				
1	BS	20ABS9914	Discrete Mathematical Structures	3	0	0	3	30	70	100
2	PC	20APC3601	Digital Electronics and Microprocessors	3	0	0	3	30	70	100
3	PC	20APC3602	Database Management Systems	3	0	0	3	30	70	100
4	PC	20APC3604	Basics of Python Programming	3	0	0	3	30	70	100
5	ES	20AES0205	Basics of Electrical and Electronics Engineering	3	0	0	3	30	70	100
6	PC Lab	20APC3603	Database Management Systems Laboratory	0	0	3	1.5	30	70	100
7	PC Lab	20APC3605	Basics of Python Programming Lab	0	0	3	1.5	30	70	100
8	ES Lab	20AES0206	Basics of Electrical and Electronics Engineering Lab	0	0	3	1.5	30	70	100
9	SC	20ASC3601	Client Side Scripting	1	0	2	2	100	0	100
10	MC	20AMC9902	Constitution of India	2	0	0	0	30	0	30
			<b>Total credits</b>				<b>21.5</b>	<b>370</b>	<b>560</b>	<b>930</b>

Course Code	Discrete Mathematical Structures		L	T	P	C
20ABS9914			3	0	0	3
Pre-requisite	Basic Mathematics	Semester	II-I			
<b>Course Objectives:</b>						
Introduce the concepts of mathematical logic and gain knowledge in sets, relations and functions and Solve problems using counting techniques and combinatorics and to introduce generating functions and recurrence relations. Use Graph Theory for solving real world problems.						
<b>Course Outcomes (CO):</b>						
After completion of the course, students will be able to <b>CO1:</b> Apply mathematical logic to solve problems. <b>CO2:</b> Understand the concepts and perform the operations related to sets, relations and functions. <b>CO3:</b> Apply basic counting techniques to solve combinatorial problems. <b>CO4:</b> Formulate problems to solve recurrence relations <b>CO5:</b> Apply Graph Theory in solving computer science problems						
<b>UNIT – I</b>	<b>Mathematical Logic</b>		9 Hrs			
Introduction, Statements and Notation, Connectives, Well-formed formulas, Tautology, Duality law, Equivalence, Implication, Normal Forms, Functionally complete set of connectives, Inference Theory of Statement Calculus, Predicate Calculus, Inference theory of Predicate Calculus.						
<b>UNIT – II</b>	<b>Set theory</b>		9 Hrs			
Basic Concepts of Set Theory, Relations and Ordering, The Principle of Inclusion- Exclusion, Pigeon hole principle and its application, Functions composition of functions, Inverse Functions, Recursive Functions, Lattices and its properties. Algebraic structures: Algebraic systems-Examples and General Properties, Semi groups and Monoids, groups, sub groups, homomorphism, Isomorphism.						
<b>UNIT – III</b>	<b>Elementary Combinatorics</b>		9 Hrs			
Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutations with Constrained Repetitions, Binomial Coefficients, The Binomial and Multinomial Theorems.						
<b>UNIT – IV</b>	<b>Recurrence Relations</b>		9 Hrs			
Generating Functions of Sequences, Calculating Coefficients of Generating Functions, Recurrence relations, Solving Recurrence Relations by Substitution and Generating functions, The Method of Characteristic roots, Solutions of Inhomogeneous Recurrence Relations.						
<b>UNIT – V</b>	<b>Graphs</b>		9 Hrs			
Basic Concepts, Isomorphism and Sub-graphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler’s Formula, Multigraphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four Color Problem						
<b>Textbooks:</b>						
<ol style="list-style-type: none"> <li>1. Joe L. Mott, Abraham Kandel and Theodore P. Baker, Discrete Mathematics for Computer Scientists &amp; Mathematicians, 2nd Edition, Pearson Education.</li> <li>2. J.P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill, 2002.</li> </ol>						
<b>Reference Books:</b>						
<ol style="list-style-type: none"> <li>1. Kenneth H. Rosen, Discrete Mathematics and its Applications with Combinatorics and Graph Theory, 7th Edition, McGraw Hill Education (India) Private Limited.</li> <li>2. Graph Theory with Applications to Engineering and Computer Science by Narsingh Deo.</li> </ol>						
<b>Online Learning Resources:</b>						
<a href="http://www.cs.yale.edu/homes/aspnes/classes/202/notes.pdf">http://www.cs.yale.edu/homes/aspnes/classes/202/notes.pdf</a>						

**Mapping of course outcomes with program outcomes**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	PO1 2	PSO 1	PSO2
<b>CO1</b>	3	2	2											
<b>CO2</b>	3	2	2	2										
<b>CO3</b>	2	3	2											
<b>CO4</b>	3	2	2		2									
<b>CO5</b>	3	2	2	3	2									

(Levels of Correlation, viz., 1-Low, 2-Moderate, 3 High)

Course Code	Digital Electronics & Microprocessors		L	T	P	C
20APC3601			3	0	0	3
Pre-requisite	Basic Electronics	Semester	II-I			
<b>Course Objectives:</b>						
<ol style="list-style-type: none"> <li>To understand all the concepts of Logic Gates and Boolean Functions.</li> <li>To learn about Combinational Logic and Sequential Logic Circuits.</li> <li>To design logic circuits using Programmable Logic Devices.</li> <li>To understand basics of 8086 Microprocessor and 8051 Microcontroller.</li> <li>To understand architecture of 8086 Microprocessor and 8051 Microcontroller.</li> <li>To learn Assembly Language Programming of 8086 and 8051.</li> </ol>						
<b>Course Outcomes (CO):</b>						
After Completion of this course, the student will be able to: <b>CO1:</b> Design Logic circuit using basic concepts of Boolean algebra. <b>CO2:</b> Design Logic circuit using basic concepts of PLDs. <b>CO3:</b> Design sequential logic circuits. <b>CO4:</b> Design application using 8086 Microprocessor. <b>CO5:</b> Design application using 8051 Microcontroller.						
<b>UNIT - I</b>	<b>Number Systems &amp; Code Conversion</b>		9 Hrs			
Number Systems & Code conversion, Boolean Algebra & Logic Gates, Truth Tables, Universal Gates, Simplification of Boolean functions, SOP and POS methods – Simplification of Boolean functions using K-maps, Signed and Unsigned Binary Numbers.						
<b>UNIT - II</b>	<b>Combinational Circuits</b>		9 Hrs			
Combinational Logic Circuits: Adders & Subtractors, Multiplexers, Demultiplexers, Encoders, Decoders, Programmable Logic Devices.						
<b>UNIT - III</b>	<b>Sequential Circuits</b>		9 Hrs			
Sequential Logic Circuits: RS, Clocked RS, D, JK, Master Slave JK, T Flip-Flops, Shift Registers, Types of Shift Registers, Counters, Ripple Counter, Synchronous Counters, Asynchronous Counters, Up-Down Counter.						
<b>UNIT - IV</b>	<b>Microprocessors - I</b>		9 Hrs			
8085 microprocessor Review (brief details only), 8086 microprocessor, Functional Diagram, register organization 8086, Flag register of 8086 and its functions, Addressing modes of 8086, Pin diagram of 8086, Minimum mode & Maximum mode operation of 8086, Interrupts in 8086.						
<b>UNIT - V</b>	<b>Microprocessors - II</b>		9 Hrs			
Instruction set of 8086, Assembler directives, Procedures and Macros, Simple programs involving arithmetic, logical, branch instructions, Ascending, Descending and Block move programs, String Manipulation Instructions. Overview of 8051 microcontroller, Architecture, I/O ports and Memory organization, addressing modes and instruction set of 8051(Brief details only), Simple Programs.						
<b>Text Books:</b>						
<ol style="list-style-type: none"> <li>M. Morris Mano, Michael D. Ciletti, Digital Design, Pearson Education, 5th Edition, 2013</li> <li>Anil K. Maini, Digital Electronics: Principles, Devices and Applications, John Wiley &amp; Sons, Ltd., 2007.</li> <li>N. Senthil Kumar, M. Saravanan, S. Jeevanathan, Microprocessor and Microcontrollers, Oxford Publishers, 2010.</li> <li>Advanced microprocessors and peripherals-A.K Ray and K.M. Bhurchandani, TMH, 2nd edition, 2006.</li> </ol>						
<b>Reference Books:</b>						
<ol style="list-style-type: none"> <li>Thomas L. Floyd, Digital Fundamentals – A Systems Approach, Pearson, 2013.</li> <li>Charles H. Roth, Fundamentals of Logic Design, Cengage Learning, 5th, Edition, 2004.</li> <li>D.V.Hall, Microprocessors and Interfacing. TMGH, 2nd edition, 2006.</li> </ol>						
<b>Online Learning Resources:</b>						
NPTEL, SWAYAM						

**Mapping of course outcomes with program outcomes**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	PO1 2	PSO 1	PSO2
CO1	3	2	2										1	
CO2	3	2	2										2	
CO3	3	2	2	1									2	
CO4	3	2	2	1									2	1
CO5	3	2	2	1									2	1

(Levels of Correlation, viz., 1-Low, 2-Moderate, 3 High)

Course Code	Database Management Systems		L	T	P	C
20APC3602			3	0	0	3
<b>Pre-requisite</b>	<b>NIL</b>	<b>Semester</b>	<b>II-I</b>			
<b>Course Objectives:</b>						
This course is designed to: <ul style="list-style-type: none"> <li>• Train in the fundamental concepts of database management systems, database modeling and design, SQL, PL/SQL and system implementation techniques.</li> <li>• Enable students to model ER diagrams for any customized application</li> <li>• Inducting appropriate strategies for optimization of queries.</li> <li>• Provide knowledge on transaction and concurrency techniques</li> </ul>						
<b>Course Outcomes (CO):</b>						
After completion of the course, students will be able to <ul style="list-style-type: none"> <li><b>CO1:</b> know the fundamentals of Databases</li> <li><b>CO2:</b> Understand SQL and PL/SQL Concepts</li> <li><b>CO3:</b> Design a database for a real-world information system</li> <li><b>CO4:</b> Process and Optimize the query</li> <li><b>CO5:</b> Working of transaction and concurrency techniques in real time applications</li> </ul>						
<b>UNIT - I</b>	<b>Introduction, Introduction to Relational Model</b>		9Hrs			
Introduction: Database systems applications, Purpose of Database Systems, view of Data, Database Languages, Relational Databases, Database Design, Data Storage and Querying, Transaction Management, Database Architecture, Data Mining and Information Retrieval, Specialty Databases, Database users and Administrators, Introduction to Relational Model: Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, Relational Operations						
<b>UNIT - II</b>	<b>Introduction to SQL, Advanced SQL</b>		9 Hrs			
Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Sub-queries, Modification of the Database. Intermediate SQL: Joint Expressions, Views, Transactions, Integrity Constraints, SQL Data types and schemas, Authorization. Advanced SQL: Accessing SQL from a Programming Language, Functions and Procedures, Triggers, Recursive Queries, OLAP, Formal relational query languages.						
<b>UNIT - III</b>	<b>Database Design and the E-R Model, Relational Database Design</b>		9 Hrs			
Database Design and the E-R Model: Overview of the Design Process, The Entity-Relationship Model, Constraints, Removing Redundant Attributes in Entity Sets, Entity-Relationship Diagrams, Reduction to Relational Schemas, Entity-Relationship Design Issues. Relational Database Design: Features of Good Relational Designs, Atomic Domains and First Normal Form, Decomposition Using Functional Dependencies, Functional-Dependency Theory, Algorithms for Decomposition, Decomposition Using Multivalued Dependencies, More Normal Forms.						
<b>UNIT - IV</b>	<b>Query Processing, Query optimization</b>		9 Hrs			
Query Processing: Overview, Measures of Query cost, Selection operation, sorting, Join Operation, other operations, Evaluation of Expressions. Query optimization: Overview, Transformation of Relational Expressions, Estimating statistics of Expression results, Choice of Evaluation Plans, Materialized views, Advanced Topics in Query Optimization.						
<b>UNIT - V</b>	<b>Transaction Management, Concurrency control and Recovery System</b>		10Hrs			
Transaction Management: Transactions: Concept, A Simple Transactional Model, Storage Structures, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Isolation and Atomicity, Transaction Isolation Levels, Implementation of Isolation Levels, Transactions as SQL Statements. Concurrency Control: Lock-based Protocols, Deadlock Handling, Multiple granularity, Timestamp-based Protocols, and Validation-based Protocols. Recovery System: Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm, Buffer Management, Failure with Loss of Nonvolatile Storage, Early Lock Release and Logical Undo Operations.						
<b>Textbooks:</b>						
1. A. Silberschatz, H.F.Korth, S.Sudarshan, "Database System Concepts",6/e, TMH 2019						
<b>Reference Books:</b>						
1. Database Management System, 6/e Ramez Elmasri, Shamkant B. Navathe, PEA 2. Database Principles Fundamentals of Design Implementation and Management, Carlos Coronel, Steven Morris, Peter Robb, Cengage Learning.						

3. Database Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke, TMH

**Online Learning Resources:**

[https://onlinecourses.nptel.ac.in/noc21\\_cs04/preview](https://onlinecourses.nptel.ac.in/noc21_cs04/preview)

**Mapping of course outcomes with program outcomes**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	PO1 2	PSO 1	PSO2
CO1	2													
CO2	2				2									
CO3	2	1	2											
CO4	3	2	3										2	
CO5	2	2												

(Levels of Correlation, viz., 1-Low, 2-Moderate, 3 High)

Course Code	Basics of Python Programming			L	T	P	C
20APC3604				3	0	0	3
Pre-requisite	NILL	Semester	II-I				
<b>Course Objectives:</b>							
<ul style="list-style-type: none"> <li>To learn the fundamentals of Python</li> <li>To elucidate problem-solving using a Python programming language</li> <li>To introduce a function-oriented programming paradigm through python</li> <li>To get training in the development of solutions using modular concepts</li> <li>To introduce the programming constructs of python</li> </ul>							
<b>Course Outcomes (CO):</b>							
<b>CO1:</b> Understanding the syntax and semantics of Python programming. <b>CO2:</b> Apply modularity to programs. <b>CO3:</b> Select appropriate data structure of Python for solving a problem. <b>CO4:</b> Implement Mutable and Immutable data types <b>CO5:</b> Interpret the concepts of object oriented programming as used in Python							
<b>UNIT - I</b>							9Hrs
<b>Introduction:</b> What is a program, Running python, Arithmetic operators, Value and Types. <b>Variables, Assignments and Statements:</b> Assignment statements, Script mode, Order of operations, string operations, comments. <b>Functions:</b> Function calls, Math functions, Composition, Adding new Functions, Definitions and Uses, Flow of Execution, Parameters and Arguments, Variables and Parameters are local, Stack diagrams, Fruitful Functions and Void Functions, Why Functions.							
<b>UNIT - II</b>							9 Hrs
<b>Case study:</b> The turtle module, Simple Repetition, Encapsulation, Generalization, Interface design, Refactoring, docstring. <b>Conditionals and Recursion:</b> floor division and modulus, Boolean expressions, Logical operators, Conditional execution, Alternative execution, Chained conditionals, Nested conditionals, Recursion, Infinite Recursion, Keyboard input. <b>Fruitful Functions:</b> Return values, Incremental development, Composition, Boolean functions, more recursion, Leap of Faith, Checking types							
<b>UNIT - III</b>							9 Hrs
<b>Iteration:</b> Reassignment, Updating variables, The while statement, Break, Square roots, Algorithms. <b>Strings:</b> A string is a sequence, len, Traversal with a for loop, String slices, Strings are immutable, Searching, Looping and Counting, String methods, The in operator, String comparison. <b>Case Study:</b> Reading word lists, Search, Looping with indices. <b>Lists:</b> List is a sequence, Lists are mutable, Traversing a list, List operations, List slices, List methods, Map filter and reduce, Deleting elements, Lists and Strings, Objects and values, Aliasing, List arguments.							
<b>UNIT - IV</b>							8 Hrs
<b>Dictionaries:</b> A dictionary is a mapping, Dictionary as a collection of counters, Looping and dictionaries, Reverse Lookup, Dictionaries and lists, Memos, Global Variables. <b>Tuples:</b> Tuples are immutable, Tuple Assignment, Tuple as Return values, Variable-length argument tuples, Lists and tuples, Dictionaries and tuples, Sequences of sequences. <b>Files:</b> Persistence, Reading and writing, Format operator, Filename and paths, Catching exceptions, Databases, Pickling, Pipes, Writing modules. <b>Classes and Objects:</b> Programmer-defined types, Attributes, Instances as Return values, Objects are mutable, Copying.							
<b>UNIT - V</b>							10Hrs
<b>Classes and Functions:</b> Time, Pure functions, Modifiers, Prototyping versus Planning <b>Classes and Methods:</b> Object oriented features, Printing objects, The init method, The <code>__str__</code> method, Operator overloading, Type-based Dispatch, Polymorphism, Interface and Implementation <b>Inheritance:</b> Card objects, Class attributes, Comparing cards, decks, Printing the Deck, Add Remove shuffle and sort, Inheritance, Data encapsulation. <b>The Goodies:</b> Conditional expressions, List comprehensions, Generator expressions, any and all, Sets, Counters, default dict, Named tuples, Gathering keyword Args							
<b>Textbooks:</b>							
1. Allen B. Downey, "Think Python", 2nd edition, SPD/O'Reilly, 2016.							
<b>Reference Books:</b>							
1. Martin C.Brown, "The Complete Reference: Python", McGraw-Hill, 2018. 2. Kenneth A. Lambert, B.L. Juneja, "Fundamentals of Python", CENGAGE, 2015. 3. R. Nageswara Rao, "Core Python Programming", 2nd edition, Dreamtech Press, 2019							



**Mapping of course outcomes with program outcomes**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	PO1 2	PSO 1	PSO2
<b>CO1</b>	3		2		2									
<b>CO2</b>	2			2									2	1
<b>CO3</b>	2	2	2	2									2	1
<b>CO4</b>	2		3		2								2	1
<b>CO5</b>	2	2	3		3				2				2	1

(Levels of Correlation, viz., 1-Low, 2-Moderate, 3 High)

Course Code	Basics of Electrical & Electronics Engineering		L	T	P	C
20AES0205			3	0	0	3
Pre-requisite	NIL	Semester	II-I			
<b>Course Outcomes (CO):</b>						
<b>CO1:</b> Apply concepts of KVL/KCL in solving DC circuits <b>CO2:</b> Illustrate working principles of induction motor - DC Motor <b>CO3:</b> Identify type of electrical machine based on their operation <b>CO4:</b> Describe operation and characteristics of diodes and transistors. <b>CO5:</b> Make use of diodes and transistors in simple, typical circuit applications. <b>CO6:</b> Understand operation of basic op-amp circuits.						
<b>PART-A (Electrical Engineering)</b>						
<b>UNIT - I</b>	<b>DC &amp; AC Circuits</b>		9Hrs			
Electrical circuit elements (R - L and C) - Kirchhoff laws - Series and parallel connection of resistances with DC excitation. Superposition Theorem - Representation of sinusoidal waveforms - peak and rms values - phasor representation - real power - reactive power - apparent power - power factor - Analysis of single-phase ac circuits consisting of RL - RC - RLC series circuits.						
<b>UNIT - II</b>	<b>DC &amp; AC Machines</b>		9 Hrs			
Principle and operation of DC Generator - EMF equations - OCC characteristics of DC generator – principle and operation of DC Motor – Performance Characteristics of DC Motor - Speed control of DC Motor – Principle and operation of Single Phase Transformer - OC and SC test on transformer - principle and operation of Induction Motor [ Elementary treatment only ]						
<b>UNIT - III</b>	<b>Basics of Power Systems</b>		9 Hrs			
Layout & operation of Hydro, Thermal, Nuclear Stations - Solar & wind generating stations – Typical AC Power Supply scheme – Elements of Transmission line – Types of Distribution systems: Primary & Secondary distribution systems						
<b>Text Books:</b>						
<ol style="list-style-type: none"> <li>1. D. P. Kothari and I. J. Nagrath - “Basic Electrical Engineering” - Tata McGraw Hill - 2010.</li> <li>2. V.K. Mehta &amp; Rohit Mehta, “Principles of Power System” – S.Chand – 2018.</li> </ol>						
<b>References:</b>						
<ol style="list-style-type: none"> <li>1. L. S. Bobrow - “Fundamentals of Electrical Engineering” - Oxford University Press - 2011.</li> <li>2. E. Hughes - “Electrical and Electronics Technology” - Pearson - 2010.</li> <li>3. C.L. Wadhwa – “Generation Distribution and Utilization of Electrical Energy”, 3rd Edition, New Age International Publications.</li> </ol>						
<b>PART-B (Electronics Engineering)</b>						
<b>UNIT - I</b>	<b>PN JUNCTION DIODE &amp; SPECIAL DIODE CHARACTERISTICS</b>		9 Hrs			
Overview of Semiconductors, PN junction diode, Zener diode, Applications of diode as switch and rectifier, Zener diode as regulator, special purpose diodes: schottky diode, tunnel diode, varactor diode, photodiode, phototransistor and LED.						
<b>UNIT - II</b>	<b>TRANSISTOR CHARACTERISTICS</b>		10Hrs			
BJT construction, operation, configuration and characteristics, JFET and MOSFET construction, operation, characteristics (CS configuration), applications Operational Amplifiers: Introduction, block diagram, basic op-amp circuits: Inverting, Non Inverting, summer, subtractor, voltage follower.						
<b>UNIT - III</b>	<b>COMMUNICATION SYSTEMS</b>					
Introduction, Elements of Communication Systems, EM spectrum, basics of electronic communication, Amplitude and Frequency modulation, Pulse modulation, Communication receivers, Examples of communication systems: Microwave & Satellite, Fibre optic, Television, mobile communication (block diagram approach).						
<b>Textbooks:</b>						
<ol style="list-style-type: none"> <li>1. D.P. Kothari, I.J.Nagrath, Basic Electronics, 2<sup>nd</sup> edition, McGraw Hill Education(India)Private Limited</li> <li>2. S.K. Bhattacharya, Basic Electrical and Electronics Engineering, 2<sup>nd</sup> edition, Pearson India Private Limited</li> </ol>						
<b>Reference Books:</b>						
<ol style="list-style-type: none"> <li>1. R. Muthu subramanian, S. Salivahanan, “Basic Electrical and Electronics Engineering”, Tata McGraw-Hill Education, Reprint 2012.</li> <li>2. David Bell, Electronic Devices and Circuits: Oxford University Press, 5th edition. 2008.</li> </ol>						

**Mapping of course outcomes with program outcomes**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	PO1 2	PSO 1	PSO2
<b>CO1</b>	3	2	1											
<b>CO2</b>	3	2	1											
<b>CO3</b>	3	2	1											
<b>CO4</b>	3	2	1											
<b>CO5</b>	3	2	1											
<b>CO6</b>	3	2	1											

(Levels of Correlation, viz., 1-Low, 2-Moderate, 3 High)

<b>Course Code</b>	<b>Database Management Systems Laboratory</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>												
20APC3603			<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>												
<b>Pre-requisite</b>		<b>Semester</b>	<b>II-I</b>															
<b>Course Objectives:</b>																		
<ul style="list-style-type: none"> <li>To implement the basic knowledge of SQL queries and relational algebra.</li> <li>To construct database models for different database applications.</li> <li>To apply normalization techniques for refining of databases.</li> <li>To practice various triggers, procedures, and cursors using PL/SQL.</li> <li>To design and implementation of a database for an organization</li> </ul>																		
<b>Course Outcomes (CO):</b>																		
<p>After completion of the course, students will be able to</p> <p><b>CO1:</b> Write SQL Queries</p> <p><b>CO2:</b> Implement PL/SQL programs</p> <p><b>CO3:</b> Design database for any real world problem</p>																		
<b>List of Experiments:</b>																		
<b>Week-1: CREATION OF TABLES</b>																		
1. Create a table called Employee with the following structure.																		
<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>Empno</td> <td>Number</td> </tr> <tr> <td>Ename</td> <td>Varchar2(20)</td> </tr> <tr> <td>Job</td> <td>Varchar2(20)</td> </tr> <tr> <td>Mgr</td> <td>Number</td> </tr> <tr> <td>Sal</td> <td>Number</td> </tr> </tbody> </table>							Name	Type	Empno	Number	Ename	Varchar2(20)	Job	Varchar2(20)	Mgr	Number	Sal	Number
Name	Type																	
Empno	Number																	
Ename	Varchar2(20)																	
Job	Varchar2(20)																	
Mgr	Number																	
Sal	Number																	
<ul style="list-style-type: none"> <li>Add a column commission with domain to the Employee table.</li> <li>Insert any five records into the table.</li> <li>Update the column details of job</li> <li>Rename the column of Employ table using alter command.</li> <li>Delete the employee whose empno is 19.</li> </ul>																		
2. Create department table with the following structure.																		
<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>Deptno</td> <td>Number</td> </tr> <tr> <td>Deptname</td> <td>Varchar2(20)</td> </tr> <tr> <td>location</td> <td>Varchar2(20)</td> </tr> </tbody> </table>							Name	Type	Deptno	Number	Deptname	Varchar2(20)	location	Varchar2(20)				
Name	Type																	
Deptno	Number																	
Deptname	Varchar2(20)																	
location	Varchar2(20)																	
<ol style="list-style-type: none"> <li>Add column designation to the department table.</li> <li>Insert values into the table.</li> <li>List the records of emp table grouped by deptno.</li> <li>Update the record where deptno is 9.</li> <li>Delete any column data from the table</li> </ol>																		
3. Create a table called Customer table																		
<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>Cust name</td> <td>Varchar2(20)</td> </tr> <tr> <td>Cust street</td> <td>Varchar2(20)</td> </tr> <tr> <td>Cust city</td> <td>Varchar2(20)</td> </tr> </tbody> </table>							Name	Type	Cust name	Varchar2(20)	Cust street	Varchar2(20)	Cust city	Varchar2(20)				
Name	Type																	
Cust name	Varchar2(20)																	
Cust street	Varchar2(20)																	
Cust city	Varchar2(20)																	
<ol style="list-style-type: none"> <li>Insert records into the table.</li> <li>Add salary column to the table.</li> <li>Alter the table column domain.</li> <li>Drop salary column of the customer table.</li> <li>Delete the rows of customer table whose ust_city is 'hyd'.</li> </ol>																		
4. Create a table called branch table.																		

Name	Type
Branch name	Varchar2(20)
Branch city	Varchar2(20)
asserts	Number

5. Increase the size of data type for asserts to the branch.
  - a. Add and drop a column to the branch table.
  - b. Insert values to the table.
  - c. Update the branch name column
  - d. Delete any two columns from the table

6. Create a table called sailor table

Name	Type
Sid	Number
Sname	Varchar2(20)
rating	Varchar2(20)

- a. Add column age to the sailor table.
- b. Insert values into the sailor table.
- c. Delete the row with rating>8.
- d. Update the column details of sailor.
- e. Insert null values into the table.

7. Create a table called reserves table

Name	Type
Boat id	Integer
sid	Integer
day	Integer

- a. Insert values into the reserves table.
- b. Add column time to the reserves table.
- c. Alter the column day data type to date.
- d. Drop the column time in the table.
- e. Delete the row of the table with some condition.

### Week-2: QUERIES USING DDL AND DML

1.
  - a. Create a user and grant all permissions to the user.
  - b. Insert the any three records in the employee table and use rollback. Check the result.
  - c. Add primary key constraint and not null constraint to the employee table.
  - d. Insert null values to the employee table and verify the result.
2.
  - a. Create a user and grant all permissions to the user.
  - b. Insert the any three records in the employee table and use rollback. Check the result.
  - c. Add primary key constraint and not null constraint to the employee table.
  - d. Insert null values to the employee table and verify the result.
3.
  - a. Create a user and grant all permissions to the user.
  - b. Insert values in the department table and use commit.
  - c. Add constraints like unique and not null to the department table.
  - d. Insert repeated values and null values into the table.
4.
  - a. Create a user and grant all permissions to the user.
  - b. Insert values into the table and use commit.
  - c. Delete any three records in the department table and use rollback.
  - d. Add constraint primary key and foreign key to the table.
5.
  - a. Create a user and grant all permissions to the user.
  - b. Insert records in the sailor table and use commit.
  - c. Add save point after insertion of records and verify save point.
  - d. Add constraints not null and primary key to the sailor table.

- e. Create a user and grant all permissions to the user.
  - f. Use revoke command to remove user permissions.
  - g. Change password of the user created.
  - h. Add constraint foreign key and not null.
- 6.
- a. Create a user and grant all permissions to the user.
  - b. Update the table reserves and use savepoint and rollback.
  - c. Add constraint primary key , foreign key and not null to the reserves table
  - d. Delete constraint not null to the table column

### **Week-3:QUERIES USING AGGREGATE FUNCTIONS**

1.
  - a. By using the group by clause, display the enames who belongs to deptno 10 along with average salary.
  - b. Display lowest paid employee details under each department.
  - c. Display number of employees working in each department and their department number.
  - d. Using built-in functions, display number of employees working in each department and their department name from dept table. Insert deptname to dept table and insert deptname for each row, do the required thing specified above.
  - e. List all employees which start with either B or C.
  - f. Display only these ename of employees where the maximum salary is greater than or equal to 5000.
2.
  - a. Calculate the average salary for each different job.
  - b. Show the average salary of each job excluding manager.
  - c. Show the average salary for all departments employing more than three people.
  - d. Display employees who earn more than the lowest salary in department 30
  - e. Show that value returned by sign (n)function.
  - f. How many days between day of birth to current date
3.
  - a. Show that two substrings as single string.
  - b. List all employee names, salary and 15% rise in salary.
  - c. Display lowest paid emp details under each manager
  - d. Display the average monthly salary bill for each deptno.
  - e. Show the average salary for all departments employing more than two people.
  - f. By using the group by clause, display the eid who belongs to deptno 05 alongwith average salary.
4.
  - a. Count the number of employees in department20

**b. Find the minimum salary earned by clerk.**

  - c. Find minimum, maximum, average salary of all employees.
  - d. List the minimum and maximum salaries for each job type.
  - e. List the employee names in descending order.
  - f. List the employee id, names in ascending order by empid.
5.
  - a. Find the sids ,names of sailors who have reserved all boats called“INTERLAKE  
Find the age of youngest sailor who is eligible to vote for each rating level with at least twosuch sailors.
  - b. Find the sname, bid and reservation date for each reservation.
  - c. Find the ages of sailors whose name begin and end with B and has at least 3characters.
  - d. List in alphabetic order all sailors who have reserved redboat.
  - e. Find the age of youngest sailor for each rating level.
6.
  - a. List the Vendors who have delivered products within 6 months from order date.
  - b. Display the Vendor details who have supplied both Assembled and Subparts.
  - c. Display the Sub parts by grouping the Vendor type (Local or Nonlocal).
  - d. Display the Vendor details in ascending order.
  - e. Display the Sub part which costs more than any of the Assembled parts.
  - f. Display the second maximum cost Assembled part

### **Week-4: PROGRAMS ON PL/SQL**

1.
  - a. Write a PL/SQL program to swap two numbers.
  - b. Write a PL/SQL program to find the largest of three numbers.
2.
  - a. Write a PL/SQL program to find the total and average of 6 subjects and display the grade.
  - b. Write a PL/SQL program to find the sum of digits in a given number.
3.
  - a. Write a PL/SQL program to display the number in reverse order.

- b. Write a PL/SQL program to check whether the given number is prime or not.
- 4. a. Write a PL/SQL program to find the factorial of a given number.
- b. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns radius and area.
- 5. a. Write a PL/SQL program to accept a string and remove the vowels from the string. (When 'hello' passed to the program it should display 'Hll' removing e and o from the world Hello).
- b. Write a PL/SQL program to accept a number and a divisor. Make sure the divisor is less than or equal to 10. Else display an error message. Otherwise Display the remainder in words.

**Week-5: PROCEDURES AND FUNCTIONS**

1. Write a function to accept employee number as parameter and return Basic +HRA together as single column.
2. Accept year as parameter and write a Function to return the total net salary spent for a given year.
3. Create a function to find the factorial of a given number and hence find NCR.
4. Write a PL/SQL block to print prime Fibonacci series using local functions.
5. Create a procedure to find the lucky number of a given birth date.
6. Create function to the reverse of given number

**Week-6: TRIGGERS**

1. Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values:

CUSTOMERS table:

ID	NAME	AGE	ADDRESS	SALARY
1	Alive	24	Khammam	2000
2	Bob	27	Kadappa	3000
3	Catri	25	Guntur	4000
4	Dena	28	Hyderabad	5000
5	Eeshwar	27	Kurnool	6000
6	Farooq	28	Nellore	7000

2. Creation of insert trigger, delete trigger, update trigger practice triggers using the passenger database. Passenger( Passport\_id INTEGER PRIMARY KEY, Name VARCHAR (50)Not NULL, Age Integer Not NULL, Sex Char, Address VARCHAR (50) Not NULL);
  - a. Write a Insert Trigger to check the Passport\_id is exactly six digits or not.
  - b. Write a trigger on passenger to display messages '1 Record is inserted', '1 record is deleted', '1 record is updated' when insertion, deletion and updation are done on passenger respectively.
3. Insert row in employee table using Triggers. Every trigger is created with name any trigger have same name must be replaced by new name. These triggers can be raised before insert, update or delete rows on data base. The main difference between a trigger and a stored procedure is that the former is attached to a table and is only fired when an INSERT, UPDATE or DELETE occurs.
4. Convert employee name into uppercase whenever an employee record is inserted or updated. Trigger to fire before the insert or update.
5. Trigger before deleting a record from emp table. Trigger will insert the row to be deleted into table called delete\_emp and also record user who has deleted the record and date and time of delete.
6. Create a transparent audit system for a table CUST\_MSTR. The system must keep track of the records that are being deleted or updated

**Week-7: PROCEDURES**

1. Create the procedure for palindrome of given number.
2. Create the procedure for GCD: Program should load two registers with two Numbers and then apply the logic for GCD of two numbers. GCD of two numbers is performed by dividing the greater number by the smaller number till the remainder is zero. If it is zero, the divisor is the GCD if not the remainder and the divisors of the previous division are the new set of two numbers. The process is repeated by dividing greater of the two numbers by the smaller number till the remainder is zero and GCD is found.
3. Write the PL/SQL programs to create the procedure for factorial of given number.
4. Write the PL/SQL programs to create the procedure to find sum of N natural number.
5. Write the PL/SQL programs to create the procedure to find Fibonacci series.
6. Write the PL/SQL programs to create the procedure to check the given number is perfect or not

### **Week-8: CURSORS**

1. Write a PL/SQL block that will display the name, dept no, salary of fist highest paid employees.

Update the balance stock in the item master table each time a transaction takes place in the item transaction table. The change in item master table depends on the item id is already present in the item master then update operation is performed to decrease the balance stock bythe quantity specified in the item transaction in case the item id is not present in the item master table then the record is inserted in the item master table.

3. Write a PL/SQL block that will display the employee details along with salary using cursors.
4. To write a Cursor to display the list of employees who are working as a Managers or Analyst.
5. To write a Cursor to find employee with given job and deptno.
6. Write a PL/SQL block using implicit cursor that will display message, the salaries of all the employees in the 'employee' table are updated. If none of the employee's salary is updated we get a message 'None of the salaries were updated'. Else we get a message like for example, 'Salaries for 1000 employees are updated' if there are 1000 rows in 'employee' table

### **Week-9: CASE STUDY: BOOK PUBLISHING COMPANY**

A publishing company produces scientific books on various subjects. The books are written by authors who specialize in one particular subject. The company employs editors who, not necessarily being specialists in a particular area, each take sole responsibility for editing one or more publications.

A publication covers essentially one of the specialist subjects and is normally written by a singleauthor. When writing a particular book, each author works with on editor, but may submit anotherwork for publication to be supervised by other editors. To improve their competitiveness, the company tries to employ a variety of authors, more than one author being a specialist in a particularsubject for the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams

### **Week-10: CASE STUDY GENERAL HOSPITAL**

A General Hospital consists of a number of specialized wards (such as Maternity, Pediatric, Oncology, etc.). Each ward hosts a number of patients, who were admitted on the recommendation of their own GP and confirmed by a consultant employed by the Hospital. On admission, the personaldetails of every patient are recorded. A separate register is to be held to store the information ofthe tests undertaken and the results of a prescribed treatment. A number of tests may be conductedfor each patient. Each patient is assigned to one leading consultant but may be examined byanother doctor, if required. Doctors are specialists in some branch of medicine and may be leading consultants for a number of patients, not necessarily from the same ward. For the above case study,do the following.

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams

### **Week-11: CASE STUDY: CAR RENTAL COMPANY**

A database is to be designed for a car rental company. The information required includes a description of cars, subcontractors (i.e. garages), company expenditures, company revenues and customers. Cars are to be described by such data as: make, model, year of production, engine size,fuel type, number of passengers, registration number, purchase price, purchase date, rent price andinsurance details. It is the company policy not to keep any car for a period exceeding one year. Allmajor repairs and maintenance are done by subcontractors (i.e. franchised garages), with whom CRC has long-term agreements. Therefore the data about garages to be kept in the database includes garage names, addresses, range of services and the like. Some garages require paymentsimmediately after a repair has been made; with others CRC has made arrangements for credit facilities. Company expenditures are to be registered for all outgoing connected with purchases,repairs, maintenance, insurance etc. Similarly the cash inflow coming from all sources: Car hire, car sales, insurance claims must be kept of file. CRC maintains a reasonably stable client base. For this privileged category of customers special credit card facilities are provided. These customers may also book in advance a particular car. These reservations can be made for any period of time up to one month. Casual customers must pay a deposit for an estimated time of rental, unless they wish to pay by credit card. All major credit cards are accepted. Personal detailssuch as name, address, telephone number, driving license, number about each customer are kept in the database. For the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams

### **Week-12: CASE STUDY: STUDENT PROGRESS MONITORING SYSTEM**

A database is to be designed for a college to monitor students' progress throughout their course ofstudy. The students are reading for a degree (such as BA, BA (Hons.) M.Sc., etc) within the framework of the modular system. The college provides a number of modules, each being characterized by its code, title, credit value, module leader, teaching staff and the department theycome from. A module is coordinated by a module leader who shares teaching duties with one or more



lecturers. A lecturer may teach (and be a module leader for) more than one module. Students are free to choose any module they wish but the following rules must be observed: Some modules require pre-requisites modules and some degree programs have compulsory modules. The database is also to contain some information about students including their numbers, names, addresses, degrees they read for, and their past performance i.e. modules taken and examination results. For the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.
3. Create the logical data model i.e., ER diagrams.
4. Comprehend the data given in the case study by creating respective tables with primary keys and foreign keys wherever required.
5. Insert values into the tables created (Be vigilant about Master- Slave tables).
6. Display the Students who have taken M.Sc course
7. Display the Module code and Number of Modules taught by each Lecturer.
8. Retrieve the Lecturer names who are not Module Leaders.
9. Display the Department name which offers 'English' module.
10. Retrieve the Prerequisite Courses offered by every Department (with Department names).
11. Present the Lecturer ID and Name who teaches 'Mathematics'.
12. Discover the number of years a Module is taught.
13. List out all the Faculties who work for 'Statistics' Department.
14. List out the number of Modules taught by each Module Leader.
15. List out the number of Modules taught by a particular Lecturer.
16. Create a view which contains the fields of both Department and Module tables. (Hint- The fields like Module code, title, credit, Department code and its name).

Update the credits of all the prerequisite courses to 5. Delete the Module 'History' from the Module table.

**References:**

1. Ramez Elmasri, Shamkant, B. Navathe, "Database Systems", Pearson Education, 6th Edition, 2013.
2. Peter Rob, Charles Coronel, "Database System Concepts", Cengage Learning, 7th Edition, 2008.

**Online Learning Resources/Virtual Labs:**

<http://www.scoopworld.in> <http://vlabs.iitb.ac.in/vlabs-dev/labs/dblab/index.php>

**Mapping of course outcomes with program outcomes**

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

(Levels of Correlation, viz., 1-Low, 2-Moderate, 3 High)

<b>Course Code</b>	<b>Basics of Python Programming Lab</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
20APC3605			<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>
<b>Pre-requisite</b>	<b>NIL</b>	<b>Semester</b>	<b>II-I</b>			

**Course Objectives:**

- To train the students in solving computational problems
- To elucidate solving mathematical problems using Python programming language
- To understand the fundamentals of Python programming concepts and its applications.
- To understand the object-oriented concepts using Python in problem solving.

**Course Outcomes (CO):**

- CO1:** Write, Test and Debug Python Programs  
**CO2:** Implement Conditionals and Loops for Python Programs  
**CO3:** Use functions and represent Compound data using Lists, Tuples and Dictionaries  
**CO4:** Read and write data from & to files in Python and develop Application using Python  
**CO5:** Implement the problem in terms of real world object using OOPs concepts

**List of Experiments:**

1. Install Python Interpreter and use it to perform different Mathematical Computations. Try to do all the operations present in a Scientific Calculator

2. Write a function that draws a grid like the following:

```

+ - - - + - - - +
|         |         |
|         |         |
+ - - - + - - - +
|         |         |
|         |         |
+ - - - + - - - +

```

3. Write a function that draws a Pyramid with # symbols

```

#
# #
# # #
# # # #
# # # # #

```

Up to 15 hashes at the bottom

4. Using turtles concept draw a wheel of your choice
5. Write a program that draws Archimedean Spiral
6. The letters of the alphabet can be constructed from a moderate number of basic elements, like vertical and horizontal lines and a few curves. Design an alphabet that can be drawn with a minimal number of basic elements and then write functions that draw the letters. The alphabet can belong to any Natural language excluding English. You should consider at least Ten letters of the alphabet.
7. The time module provides a function, also named time that returns the current Greenwich Mean Time in "the epoch", which is an arbitrary time used as a reference point. On UNIX systems, the epoch is 1 January 1970.

```

>>> import time
>>> time.time()
1437746094.5735958

```

Write a script that reads the current time and converts it to a time of day in hours, minutes, and seconds, plus the number of days since the epoch.

8. Given  $n+r+1 \leq 2r$ .  $n$  is the input and  $r$  is to be determined. Write a program which computes minimum value of  $r$  that satisfies the above.
9. Write a program that evaluates Ackermann function
10. The mathematician Srinivasa Ramanujan found an infinite series that can be used to generate a numerical approximation of  $1/\pi$ :  
Write a function called estimate\_pi that uses this formula to compute and return an estimate of  $\pi$ .

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

It should use a while loop to compute terms of the summation until the last term is smaller than  $1e-15$  (which is Python notation for  $10^{-15}$ ). You can check the result by comparing it to `math.pi`.

11. Choose any five built-in string functions of C language. Implement them on your own in Python. You should not

use string related Python built-in functions.

12. Given a text of characters, Write a program which counts number of vowels, consonants and special characters.
13. Given a word which is a string of characters. Given an integer say 'n', Rotate each character by 'n' positions and print it. Note that 'n' can be positive or negative.
14. Given rows of text, write it in the form of columns.
15. Given a page of text. Count the number of occurrences of each letter (Assume case insensitivity and don't consider special characters). Draw a histogram to represent the same
16. Write program which performs the following operations on list's. Don't use built-in functions
  - a) Updating elements of a list
  - b) Concatenation of list's
  - c) Check for member in the list
  - d) Insert into the list
  - e) Sum the elements of the list
  - f) Push and pop element of list
  - g) Sorting of list
  - h) Finding biggest and smallest elements in the list
  - i) Finding common elements in the list
17. Write a program to count the number of vowels in a word.
18. Write a program that reads a file, breaks each line into words, strips whitespace and punctuation from the words, and converts them to lowercase.
19. Go to Project Gutenberg (<http://gutenberg.org>) and download your favorite out-of-copyright book in plain text format. Read the book you downloaded, skip over the header information at the beginning of the file, and process the rest of the words as before. Then modify the program to count the total number of words in the book, and the number of times each word is used. Print the number of different words used in the book. Compare different books by different authors, written in different eras.
20. Go to Project Gutenberg (<http://gutenberg.org>) and download your favorite out-of-copyright book in plain text format. Write a program that allows you to replace words, insert words and delete words from the file.
21. Consider all the files on your PC. Write a program which checks for duplicate files in your PC and displays their location. Hint: If two files have the same checksum, they probably have the same contents.
22. Consider turtle object. Write functions to draw triangle, rectangle, polygon, circle and sphere. Use object oriented approach.
23. Write a program illustrating the object oriented features supported by Python.
24. Design a Python script using the Turtle graphics library to construct a turtle bar chart representing the grades obtained by N students read from a file categorizing them into distinction, first class, second class, third class and failed.
25. Design a Python script to determine the difference in date for given two dates in YYYY:MM:DD format(0 <= YYYY <= 9999, 1 <= MM <= 12, 1 <= DD <= 31) following the leap year rules.
26. Design a Python Script to determine the time difference between two given times in HH:MM:SS format.( 0 <= HH <= 23, 0 <= MM <= 59, 0 <= SS <= 59)

**References:**

1. Allen B. Downey , “ Think Python: How to Think Like a Computer Scientist”, Second Edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016.
2. Shroff “Learning Python: Powerful Object-Oriented Programming; Fifth edition, 2013.
3. David M.Baezly “Python Essential Reference”. Addison-Wesley Professional; Fourth edition, 2009.
4. David M. Baezly “Python Cookbook” O'Reilly Media; Third edition (June 1, 2013)

**Online Learning Resources/Virtual Labs:**

<http://www.edx.org>

**Mapping of course outcomes with program outcomes**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2
<b>CO1</b>	3	3	1	2	2								1	1
<b>CO2</b>	2	1	3		2								1	2
<b>CO3</b>	2	1	3		2								2	2
<b>CO4</b>	2	1	3		2								2	2
<b>CO5</b>	3	2	2		2				2			3	2	2

**(Levels of Correlation, viz., 1-Low, 2-Moderate, 3 High)**

<b>Course Code</b>	<b>Basics of Electrical &amp; Electronics Engineering</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>20AES0206</b>	<b>Lab</b>		<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>
<b>Pre-requisite</b>	<b>NIL</b>	<b>Semester</b>	<b>II-I</b>			

**Course Outcomes (CO):**

- CO1:** Verify Kirchoff's Laws & Superposition theorem for dc supply
- CO2:** Analyze the performance of AC and DC Machines by testing.
- CO3:** Study I – V Characteristics of PV Cell & Perform speed control of dc shunt motor
- CO4:** Ability to operate diodes for finding V-I Characteristics.
- CO5:** Ability to construct and operate rectifiers without & with filters
- CO6:** Ability to construct and operate BJT & FET Characteristics.

**List of Experiments:**

**Part A: Electrical Engineering Lab**

1. Verification of Kirchoff laws.
2. Verification of Superposition Theorem.
3. Open circuit characteristics of a DC Shunt Generator.
4. Speed control of DC Shunt Motor.
5. OC & SC test of 1 – Phase Transformer.
6. Brake test on 3 - Phase Induction Motor.
7. I – V Characteristics of Solar PV cell
8. Brake test on DC Shunt Motor.

**Part B: Electronics Engineering Lab**

1. Draw and study the characteristics of semi-conductor diode
2. Draw and study the characteristics of Zener diode
3. Construct half wave rectifier without filter and with filter and also find the ripple factor and plot the output waveforms.
4. Construct full wave rectifier without filter and with filter and also find the ripple factor and plot the output waveforms.
5. Draw and study the input and output characteristics of transistor in common emitter configuration
6. Draw and study the static and transfer characteristics of FET in common source configuration
7. Study of op-amp as an inverting amplifier, non-inverting amplifier, voltage follower, summer and subtractor.
8. Conduct an experiment on am modulation & de-modulation; plot the corresponding modulated and demodulated signals.
9. Conduct an experiment on fm modulation & demodulation, plot the corresponding modulated and demodulated signals.

**Mapping of course outcomes with program outcomes**

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

(Levels of Correlation, viz., 1-Low, 2-Moderate, 3 High)

Course Code	Client Side Scripting		L	T	P	C
20ASC3601			1	0	2	2
Pre-requisite	HTML	Semester	II-I			
<b>Course Objectives:</b>						
<ul style="list-style-type: none"> <li>To provide knowledge on basic concepts of web Programming</li> <li>To design Web Pages and form validation using java scripting.</li> <li>To learn the important concepts like CSS, DOM, DNS,AJAX and XML.</li> <li>To Demonstrate the functions of html in web communication.</li> <li>To quickly be able to understand the different parts of a web page</li> </ul>						
<b>Course Outcomes (CO):</b>						
<p><b>CO1:</b> Analyze and understand the basic concepts of web programming.  <b>CO2:</b> Implement Arrays, Functions and Strings  <b>CO3:</b> Apply techniques of form validation using Java Script.  <b>CO4:</b> Describe important concepts related to client side Web Security.  <b>CO5:</b> Save client information in cookie by server</p>						
<b>UNIT - I</b>	<b>Basics of JavaScript Programming</b>		3+6 Hrs			
<p>Features of JavaScript, Object Name, Property, Method, Dot Syntax, Main Event, Values and Variables, Operators and Expressions – Primary Expressions, Object and Array Initializers, Function Definition Expression, Property Access Expressions, Invocation Expressions, If Statement, if...else, if..elseif, Nested if Statement, Switch... Case Statement, Loop Statement – for Loop, for...in Loop, while Loop, do...while Loop, continue Statement, Querying and Setting Properties and Deleting Properties, Property Getters and Setters.</p> <ul style="list-style-type: none"> <li>WAP to print hello world</li> <li>WAP to use comments in JavaScript.</li> <li>WAP to add a noscript block.</li> <li>Write a Script in &lt;head&gt;...&lt;/head&gt; section.</li> <li>Write a Script in &lt;body&gt;...&lt;/body&gt; section.</li> <li>Write a Script in &lt;body&gt;...&lt;/body&gt; and &lt;head&gt;...&lt;/head&gt; sections.</li> <li>Write a Script using arithmetic, Comparison, Logical, Bitwise, and Assignment operators</li> <li>Write code to understand how the Conditional Operator and typeof operator works in JavaScript.</li> <li>Write code to understand the working of if statement, if...else statement, and if...else if... statement.</li> <li>Implement switch-case statement.</li> <li>Implement while loop, do-while and for loop in JavaScript.</li> <li>WAP to print the web browser's Navigator object using for loop.</li> <li>WAP To implement break, continue and label in JavaScript.</li> <li>Write code to call the function that displays the text message on clicking a button.</li> </ul>						
<b>UNIT - II</b>	<b>Array, Function and String</b>		3+6 Hrs			
<p>Array – Declaring an Array, Initializing an Array, Defining an Array Elements, Looping an Array, Adding an Array Element, Sorting an Array Element, Combining an Array Elements into a String, Changing Elements of an Array, Objects as Associative Arrays, Function – Defining a Function, Writing a Function, Adding an Arguments, Scope of Variable and Arguments, Calling a Function – Calling a Function With or Without an Argument, Calling Function from HTML, Function Calling another Function, Returning the Value from a Function, String – Manipulate a String, Joining a String, Retrieving a Character from given Position, Retrieving a Position of Character in a String, Dividing Text, Copying a Sub-string, Converting String to Number and Numbers to String, Changing the Case of String, Finding a Unicode of a Character – charCodeAt(), fromCharCode()).</p> <ul style="list-style-type: none"> <li>Write code to call the function that displays the text message on clicking a button.</li> <li>WAP to call a function that takes two parameters, name and age. Print the same.</li> <li>Define a function that takes two parameters and concatenates them before returning the resultant in the calling program.</li> </ul>						
<b>UNIT - III</b>	<b>Form and Event Handling</b>		3+6 Hrs			
<p>Building Blocks of a Form, Properties and Methods of Form, Button, Text, Text Area, Checkbox, Radio Button, Select Element, Form Events – Mouse Event, Key Events, Form Objects and Elements, Changing Attribute Value Dynamically, Changing Option List Dynamically, Evaluating Checkbox Selection, Changing a Label Dynamically, Manipulating Form Elements, Intrinsic JavaScript Functions, Disabling Elements, Read Only Elements.</p> <ul style="list-style-type: none"> <li>Write code to implement the following events – onclick, onsubmit, onmouseover and onmouseout.</li> <li>Design a Registration form (include email id and password) and perform validation to all its fields.</li> </ul>						

<b>UNIT - IV</b>	<b>Objects</b>	3+6 Hrs
Window Object, Math, Number, and Date Objects, Handling Strings Using Regular Expressions. Implement Number, Date, Math, Boolean, Strings, Arrays, RegEx, and HTML DOM objects with all its properties and methods.		

<b>UNIT - V</b>	<b>Cookies and Browser Data</b>	3+6 Hrs
<p>Cookies – Basic of Cookies, Reading a Cookie Value, Writing a Cookie Value, Creating a Cookies, Deleting a Cookies, Setting the Expiration Date of Cookie, Browser – Opening a Window, Giving the New Window Focus, Window Position, Changing the Content of Window, Closing a Window, Scrolling a Web Page, Multiple Windows at Once, Creating a Web Page in New Window, JavaScript in URLs, JavaScript Security, Timers, Browser Location and History.</p> <ul style="list-style-type: none"> <li>• Set a customer name in an input cookie.</li> <li>• WAP to get all the cookies.</li> <li>• Extend the expiry date of a cookie by 1 Month.</li> <li>• Delete a cookie by setting its expiry date to one month behind the current date.</li> <li>• Do a page redirect using JavaScript at client side.</li> <li>• Show an appropriate message to your site visitors before redirecting them to a new page. WAP with a time delay to load a new page.</li> <li>• Redirect your site visitors onto a different page based on their browsers.</li> <li>• Use an alert box to give a warning message.</li> <li>• Implement a confirmation dialog box to take user's consent on any option.</li> <li>• Use a prompt dialog box.</li> <li>• Use of void is to purposely generate the undefined value.</li> <li>• Demonstrates how to create an Object.</li> <li>• Create an object with a User-Defined Function.</li> <li>• Write code to add a function along with an object.</li> <li>• Demonstrate with keyword in JavaScript.</li> </ul>		

<b>Textbooks:</b>
<ol style="list-style-type: none"> <li>1. Javascript Beginners Guide, John Pollock, TMH, 4th Edition</li> <li>2. JavaScript. Demystified, JIM KEOGH , McGraw-Hill.</li> </ol>

<b>Reference Books:</b>
<ol style="list-style-type: none"> <li>1. JavaScript™ For Dummies,® 4th Edition, by Emily Vander Veer, Published by Wiley Publishing, Inc © 2005.</li> <li>2. JavaScript for impatient programmers (beta), by Dr. Axel Rauschmayer © 2019.</li> <li>3. Javascript: Beginners Guide on Javascript Programming, by Nick Goddard © 2016.</li> </ol>

<b>Online Learning Resources:</b>
W3Schools, <a href="https://www.tutorialspoint.com/javascript/index.htm">https://www.tutorialspoint.com/javascript/index.htm</a> , nptel Videos

**Mapping of course outcomes with program outcomes**

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

**(Levels of Correlation, viz., 1-Low, 2-Moderate, 3 High)**

<b>Course Code</b>	<b>Constitution Of India</b>				<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
20AMC9902	<b>(Common to : CSE, CIC, AIM, AID)</b>				<b>2</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>Pre-requisite</b>	<b>NIL</b>	<b>Semester</b>			<b>II-I</b>			
<b>Course Outcomes (CO):</b>								
Students will be able to:								
<b>CO1:</b> Discuss the growth of the demand for civil rights in India for the bulk of Indians before the arrival of Gandhi in Indian politics.								
<b>CO2:</b> Discuss the intellectual origins of the framework of argument that informed the conceptualization of social reforms leading to revolution in India.								
<b>CO3:</b> Discuss the circumstances surrounding the foundation of the Congress Socialist Party [CSP] under the leadership of Jawaharlal Nehru and the eventual failure of the proposal of direct elections through adult suffrage in the Indian Constitution.								
<b>CO4:</b> Discuss the Powers and functions of Governor, President, Judiciary.								
<b>CO5:</b> Discuss the functions of local administration bodies								
<b>UNIT - I</b>					8Hrs			
History of Making of the Indian Constitution - History Drafting Committee, (Composition & Working).								
<b>UNIT - II</b>					9Hrs			
Philosophy of the Indian Constitution - Preamble Salient Features								
<b>UNIT - III</b>					8Hrs			
Contours of Constitutional Rights & Duties - Fundamental Rights - Right to Equality- Right to Freedom - Right against Exploitation - Right to Freedom of Religion - Cultural and Educational Rights - Right to Constitutional Remedies - Directive Principles of State Policy - Fundamental Duties.								
<b>UNIT - IV</b>					8Hrs			
Organs of Governance - Parliament - Composition - Qualifications and Disqualifications - Powers and Functions - Executive - President - Governor - Council of Ministers - Judiciary, Appointment and Transfer of Judges, Qualifications - Powers and Functions								
<b>UNIT - V</b>					9 Hrs			
Local Administration - District's Administration head: Role and Importance - Municipalities: Introduction, Mayor and role of Elected Representative, CEO of Municipal Corporation- Pachayati raj: Introduction, PRI: ZillaPachayat - Elected officials and their roles, CEO Zilla Panchayat: Position and role - Block level: Organizational Hierarchy (Different departments) - Village level: Role of Elected and Appointed officials - Importance of grass root democracy.								
<b>Textbooks:</b>								
<ol style="list-style-type: none"> <li>1. The Constitution of India, 1950 (Bare Act), Government Publication.</li> <li>2. Dr. S. N. Busi, Dr. B. R. Ambedkar framing of Indian Constitution, 1st Edition, 2015.</li> <li>3. M. P. Jain, Indian Constitution Law, 7th Edn., Lexis Nexis, 2014.</li> <li>4. D.D. Basu, Introduction to the Constitution of India, Lexis Nexis, 2015.</li> </ol>								

**Mapping of course outcomes with program outcomes**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
<b>CO1</b>						1								
<b>CO2</b>						3								
<b>CO3</b>						3								
<b>CO4</b>						3								
<b>CO5</b>						3								

(Levels of Correlation, viz., 1-Low, 2-Moderate, 3 High)