65201

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

(Common to CSE & IT)

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 3	End Exam Duration	: 3 Hours

Course Objectives:

1. To understand and apply various object-oriented programming features like abstraction, encapsulation, inheritance and polymorphism to solve various computing problems using Java language.
2. To identify, define and implement exception handling and multi-threading mechanisms in application domains.
3. To design and develop GUI applications using AWT and Swings.

Unit I - Introduction to Java and Building Blocks of Java

Basics of Java- History/Background of Java, Java Buzzwords, Java Virtual Machine and Byte code, Java Environment setup, Java Program structure, Data Types, Variables- Scope and Life Time, Operators, Expressions, Type Conversions and Type casting, Conditional statements and Control statements, Simple Java Programs, javac and java command flags.

OOP Concepts –I: Encapsulation- Classes and Objects, Classes: Class structure, class components, Objects: Object declaration, Reference variables, Constructors - default Constructor, Parameterized Constructors, Constructor overloading, this keyword and its uses, arrays concept, static modifier, access modifiers, Wrapper classes.

Methods - Passing parameters to methods – Passing primitive types and Passing Objects, getters and setters, Method Overloading, Command-line arguments, garbage collection-java.lang.System.gc(), finalize(). **String Handling** - String class, String APIs, String Buffer and String Builder classes.

Unit II - OOP Concepts –II

Inheritance- Inheritance concept, super class and subclass relationship, Object class, principle of substitution, effect of access modifiers on inheritance. Usage of super (field, method, constructor) and final(field, class, method) keywords,

Polymorphism- method overriding, Dynamic method dispatch, Abstract classes and Interfaces - Abstract classes - concept, usage, Interfaces – declaration, implementation, components of an interface, extending interfaces.
Packages – package access, CLASSPATH, package access rules, Introduction to Java standard library and Java documentation.

Unit III - Dealing exceptions and I/O

Fundamentals of exception handling, benefits of exception handling, Exception types, Termination or resumptive models, Uncaught exceptions, using try and catch, multiple catch clauses, nested try statements, exception hierarchy, throw, throws and finally, built in Exceptions, Custom exceptions, Throwable Class
Java I/O– Byte streams, character streams, Scanner class, Console class, Serialization and Serializable interface, File class.

Unit IV – Multithreading and Introduction to Applets

Multithreading- Fundamentals, Thread Life Cycle, Ways of creating threads - Thread class and Runnable interface, Thread priorities, Creating multiple threads, core methods of Thread class, Thread Synchronization, inter thread communication.
Annotations- Annotation Basics, Specifying a Retention Policy, the Annotated Element Interface, Using Default Values, Marker Annotations, Single-Member Annotations.

Introduction to Applets- Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets

Unit V - GUI Development

AWT- Basics of GUI Programming, Event handling – Delegation event model, event sources, event listeners, event classes, adapter classes: nested classes and interfaces, anonymous inner classes handling keyboard and mouse events.

Swing I - MVC Architecture, Containers, components, layout managers, frames and windows, panels, buttons, checkboxes, radio buttons, combo boxes, lists, labels, color choosers, file choosers, text fields, text areas, tool tips

Swing II - menus, progress bars, tool bars, trees, editor and text panes, tables concurrency in Swing.

Course Outcomes:

At the end of the course, student should be able to

CO 1 : Design and implement object oriented concepts like encapsulation, abstraction and data hiding using programming constructs offered by java language.
CO 2 : Realize the power of inheritance, interfaces and packages.
CO 3 : Understand and demonstrate the concepts of exception handling and java io streams.
CO 4 : Demonstrate knowledge and understanding of multi threading, annotations and applets in java.
CO 5 : Design and develop java applications using AWT and Swings for providing solutions to real world problems.

Text Books:

1. Java: The Complete Reference, Herbert Schildt, 10th Edition, McGraw-Hill Education, Oracle Press, 2017.
2. Head First Java, Kathy Sierra and Bert Bates, 2nd Edition, O'Reilly Media, 2005.

References:

1. Core Java Volume I- Fundamentals, Cay S. Horstmann and Gary Cornell, 9th Edition, Prentice Hall, 2012.
2. Core Java Volume II- Advanced Features, Cay S. Horstmann and Gary Cornell, 9th Edition, Prentice Hall, 2013.

65202

DISCRETE STRUCTURES & GRAPH THEORY

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 3	End Exam Duration	: 3 Hours

Course Objectives:

1. To inculcate mathematical thinking and problem-solving skills associated with writing proofs.
2. To expose students to a wide variety of mathematical concepts that are used in the Computer Science discipline, which may include concepts drawn from the areas Number Theory, Graph Theory, Combinatorics and Probability.

Unit I - Mathematical Logic

Statements and notations, connectives, Well Formed Formulas, Truth tables, tautology, equivalence implication, Normal forms, Predicative logic, Quantifiers, universal quantifiers, Free & Bound variables, Rules of inference, Consistency, Proof by contradiction, Automatic Theorem proving and Applications.

Unit II - Relations

Properties of binary Relations, Equivalence, Transitive closure, Compatibility & Partial ordering Relations, Lattice and its properties, Hasse Diagram.

Functions: Inverse function, Composition of functions, Recursive functions and Applications.

Unit III - Algebraic structures

Algebraic systems Examples and general properties, semi groups and Monoids, Groups, sub groups, Homomorphism & Isomorphism and Applications

Unit IV - Elementary Combinatorics

Basics of counting, combinations & permutations, with repetitions, constrained repetitions, The principle of inclusion and exclusion, Binomial Coefficients, Binomial & Multinomial theorems, Pigeon hole principles and its applications.

Generating Functions- Generating Functions of sequences, Calculating coefficient of generating function and Applications.

Unit V - Graph Theory

Representation of Graph, Sub graphs and Multi graphs**,** Spanning Trees, DFS, BFS, Planar graphs, Isomorphism, Euler circuits and Hamiltonian graphs, Chromatic Numbers and Applications.

Course Outcomes:

CO 1 : Apply formal logic proofs and/or informal, but rigorous, logical reasoning to evolve theoretical proofs to real problems, such as predicting the behavior of software or solving problems such as puzzles.
CO 2 : Apply the logical notations to define and reason about fundamental mathematical concepts such as sets, relations, functions, and integers.
CO 3 : Understand and appreciate simple proofs of problems result in group theory.
CO 4 : Apply the concept of permutations and combinations to problem solving.
CO 5 : Demonstrate knowledge of fundamental concepts in graph theory.

Text Books:

1. Discrete Mathematical Structures with Applications to Computer Science, J.P. Tremblay and R. Manohar, Tata McGraw-Hill Publishing Company, 2008.
2. Discrete Mathematics for Computer Scientists & Mathematicians, J.L. Mott., A. Kandel and T.P. Baker, 2nd Edition, Prentice Hall, 2009.

References:

1. Discrete Mathematics and its Applications, Kenneth H.Rosen, 7th Edition, TMH, 2015.
2. Discrete and Combinatorial Mathematics- An Applied Introduction, Ralph P. Grimaldi, 5th Edition, Pearson Education, 2008.
3. Elements of Discrete Mathematics – A computer Oriented Approach, C L Liu, and D P Mohapatra, 3rd Edition, Tata McGraw-Hill, 2008.

67202

BOOLEAN ALGEBRA AND CIRCUIT DESIGN

(Common to CSE & IT)

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 3	End Exam Duration	: 3 Hours

Course Objectives:

1. To impart basic properties of Boolean algebra and to simplify Boolean functions.
2. To design fundamental components of a computer such as multiplexers and registers using combinatorial and sequential circuits.
3. To impart skillset to translate Boolean functions into modular programmable components.

Unit I - Binary Systems

Digital systems, Binary numbers, Number base conversions, Octal and Hexadecimal numbers, Complements, Signed binary numbers, Binary Codes, Binary logic.

Boolean Algebra and Logic Gates: Basic definitions, axiomatic definition of Boolean algebra, basic theorems and properties of Boolean algebra, Boolean functions, canonical and standard forms, Other logic operations, digital logic gates, Integrated Circuits.

Unit II - Gate Level Minimization

The map method, four-variable map, five-Variable map, product of sums simplification, don't-care conditions, NAND and NOR implementation, other Two- level implementations, Exclusive – OR function, Quine-McClusky method – Row and Column Dominance.

Unit III - Combinational Logic

Combinational circuits, analysis procedure, design procedure, binary adder- subtractor, decimal adder, binary multiplier, Magnitude comparator, decoder, encoders, multiplexers, Demultiplexer, Realization of combinational logic using Multiplexer and Decoder.

Unit IV - Synchronous Sequential Logic

Sequential circuits, latches, Flip-Flops, analysis of clocked sequential circuits, state reduction and assignment, Design Procedure

Registers and Counters: Registers, shift registers, Ripple counters, synchronous counters, other counters (counter with unused states, Ring counter, Johnson counter).

Unit V - Memory and Programmable Logic

Introduction, Random-access memory, memory decoding, error detection and correction, read only memory, Programmable Logic Array, Programmable Array Logic, sequential programmable devices.

Course Outcomes: At the end of the course, the student will be able to

CO 1 : Understand and master different number systems and realize the binary operations of Boolean algebra using logic gates.
CO 2 : Solve gate-level minimization problems using K-map and Quine-Mc Cluskey methods.
CO 3 : Analyze a given combinational circuit and Design a new optimized circuit for a given specification.
CO 4 : Analyze a given sequential circuit and Design an optimal circuit to implement a memory element or a counter.
CO 5 : Realize Programmable logic elements used in the design of processors and embedded systems.

Text Books:

1. Digital Design with an introduction to Verilog HDL, M. Morris Mano and Mikchael D. Ciletti, 5th Edition, Pearson Education/ PHI, 2013.
2. Fundamentals of Logic Design, Charles H. Roth, Larry L Kinney, 7th Edition, Cengage Learning India Private Limited, 2015.

References:

1. Switching and Finite Automata Theory, Zvi Kohavi, Tata McGraw-Hill.
2. Switching Theory and Logic Design, CVS Rao, Pearson Education, 2007.
3. Digital Principles and Design, Donaldo D. Givone, Tata McGraw- Hill.

68201

PROBABILITY AND STATISTICS

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 3	End Exam Duration	: 3 Hours

Course Objectives

1. To introduce the concepts of Probability and Statistics.
2. To learn how to apply Probability and Statistics to solve engineering problems.
3. To keep a balance between theory and methodology.
4. To show the applications of Probability and Statistics in engineering with examples.
5. To learn how to apply testing of hypothesis techniques to make decisions in real time problems.

Unit I

Basic Statistics and Probability: Measures of Central tendency, Dispersion, Moments, Skewness and Kurtosis.

Probability: Basic concepts in probability—Random experiments, Mathematical, Statistical Probability, Axiomatic Approach to Probability, Addition Theorem of Probability, Conditional Probability, Multiplication Theorem of Probability, Boole's Inequality, Independent Events, Bayes' theorem.

Unit II

Univariate Random variables and probability distributions: Random variables - Discrete and continuous. Probability distributions, mass function/ density function of probability distribution. Mathematical Expectation. Binomial, Poisson, Normal, Exponential distributions and their properties.

Multiple Random variables, Correlation& Regression: Joint probability distributions- Joint probability mass/ density function, Marginal probability mass / density functions. Co-variance of two random variables, Correlation Coefficient, Rank correlation. Regression- Regression Coefficient, the lines of regression, Multiple regression for three variables.

Unit III

Sampling Distributions and Testing of Hypothesis: Definitions of Population, Sample, Parameter, Statistic, Sampling distribution, Standard error. Types of Sampling, Expected values of Sample mean and variance, Sampling distribution of mean and variance.

Estimation – Point and Interval Estimation.

Testing of Hypothesis: Null hypothesis, Alternate hypothesis, Type I & Type II errors, critical region, Level of significance, Power of the test, one tailed and two tailed Tests, calculation of p-value.

Unit IV

Large sample tests: Test for one and two Population Means when Population Variances are Known or Un Known, difference of standard deviations, Confidence Intervals for Population Means.

Test for one and two Population Proportions, Confidence Intervals for Population Proportions

Unit V

Small sample tests: Student t-distribution, Snedecor's F-distribution, Chi-square distribution and their properties

Test for one and two Population Means, F – test for two population variances, Chi-square test for Population variance, goodness of fit and independence of Attributes.

Course Outcomes :

At the end of the course, the student is able to

CO 1 : Compute probabilities using theorems in probability and probability distributions.

CO 2 : Find estimates of parameters and test hypothesis about parameters.

CO 3 : Establish relationship between variables using correlation and regression.

CO 4 : Apply the tools in Probability and Statistics in engineering

CO 5 : Take decisions using testing of hypothesis techniques.

CO 6 : Analyse live data.

Text Books:

1. Fundamentals of Mathematical Statistics, S.C.Gupta and V.K.Kapoor, Sultan Chand, 2014.
2. Higher Engineering Mathematics by Dr. B.S Grewal, Khanna Publishers.

References:

1. Probability and Statistics T.K.V.Iyengar & B.Krishna Gandhi, S Chand, 2015.
2. Applied Statistics and Probability for Engineers, D.C.Montgomery and G.C.Runger, 5th Edition, Johnwiley, 2012.
3. Probability and Statistics for Engineers and Scientists by Sheldon M.Ross, Academic Press.
4. A first Course in Probability and Statistics, B.L.S. Prakasa Rao, world Scientific, 2009.

62204

BASIC ELECTRICAL AND ELECTRONICS ENGINEERING
(Common to CSE, IT & ME)

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 3	End Exam Duration	: 3 Hours

Course Objectives:

1. Learn basics of various fundamental laws, analysis of electrical circuits, and study nature of ac quantities.
2. To study the construction, functioning of different types of electrical machines and their performance.
3. To study working and applications of various electronic devices.

Unit I - Network analysis:

Ohm's law, basic circuit components, power and energy calculations, types of elements, Kirchhoff's laws. Resistive, inductive and capacitive networks, series and parallel circuits, star delta and delta star transformation. Mesh and Nodal Analysis. Principle of super position, Simple problems.

Alternating Quantities:
Basic definitions: frequency, average values and RMS values of alternating currents and voltage, form factor and peak factor.

Unit II - DC Machines:

Construction of dc machines, DC Generator- working and principle of operation, EMF equation, types of DC Generators. DC Motor - working and principle of operation, types of dc motor, torque equation, losses and efficiency calculations. Simple problems.

Unit III - Transformers:

Principles of operation, Constructional Details, Ideal Transformer and Practical Transformer, Losses, Transformer Test, Efficiency and Regulation Calculations

3-Phase Induction Machines:
Principle of operation of induction motor – slip – torque characteristics – applications.

Unit IV - Diode Applications

Rectifiers – Half wave, Full wave and Bridge rectifiers, introduction to filters, Capacitor filter, Zener Diode characteristics Voltage regulation using Zener Diode, Varactor Diode

Unit V - Transistor Biasing and Amplifiers
Need for Biasing, operating point, Bias stability, DC load line, Fixed Bias, Voltage divider Bias, Principal of operation of CE Amplifier.

Components of LT Switchgear:
Switch Fuse Unit (SFU), MCB, Earthing, Types of Batteries.

Course outcomes:

Upon successful completion of the course, student will able to

CO1 : Identify basic circuit components and solve basic electrical and electronic problems using different principles.
CO2 : Understand the construction and working of different types of DC machines and calculate the losses and efficiency.
CO3 : Understand the Construction and working principle of AC machines and their applications in real time.
CO4 : To Analyze and design different types of diodes and rectifiers.
CO5 : To study different transistor biasing methods.

Text Books :

1. Principles of Electrical and Electronics Engineering – V.K.Mehta, 3rd edition, S.Chand & Co, 2014.
2. Electronics Devices and Circuits, S Salivahanan & N Suresh Kumar, McGraw-Hill, 4th edition, 2017.

References :

1. Basic Electrical and Electronics Engineering- S. K. Bhattacharya, Pearson Education India, 2nd Edition 2017.
2. Basic Electrical Engineering 3/E, D Kothari, I Nagrath, McGraw-Hill Education, 2009.
3. Basic Electrical and Electronics Engineering, J.V. Gupta, S.K. Kataria and Sons, 2013.
4. Electronics Devices and Circuits, R.L. Boylestand and Louis Nashelsky, 9th Edition, Pearson/Prentice Hall, 2006.

65231

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

(Common to CSE and IT)

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 1.5	End Exam Duration	: 3 Hours

Course Objectives :

1. To implement the basic concepts of object-oriented programming.
2. To implement the practical aspects of exception handling, multithreading mechanisms and Java I/O.
3. To be able to design and implement applications using GUI components.
4. To set up the necessary environment for running java applications.

Lab Problems:

1. Write a simple program with a main() method to print messages to the console; the purpose is to learn how to invoke the compiler and virtual machine through a console window.
2. Write a program to implement the different types of operators, to perform the following tasks: comparison of values, simple arithmetic, bit-wise operations.
3. Learn to install JDK and set up the PATH variable. Understand the various folders that are part of JDK and learn their purpose after installation.
4. Learn the nature of javac and java commands. Try to figure out the purpose various flags and options.
5. Write a program to check and print the grade of a student when the score is given as an integer. Use a switch statement. Rewrite the program to use a sequence of if-else statements.
6. Write a program to demonstrate the task of overloading of constructors.
7. With a well-written program demonstrate the usage of this keyword and thereby understand the implications of using same identifier for fields and parameters.
8. Write a program to print the minimum and maximum values of integer and float types. Use the constants available in the wrapper classes.
9. Use an array of integers and find the sum and average of the elements of that array.
10. Write a program to understand the concept of type casting.

11. Write a program to check the difference in passing primitive values and object references as arguments to a method.
12. Write a program to understand method overloading.
13. Write a program to utilize both standard and custom packages. The program should reflect the usage of packages in a correct manner, along with the purpose of access modifiers.
14. Learn to understand the usage of Java SE API documentation. Bookmark the main page in your browser. Look at the available packages. Learn class path settings.
15. Write a program to use gc() method of both System and Runtime classes. Experiment with other methods of those classes.
16. Practice further programs on the usage of arrays.
17. To illustrate the concept of inheritance, write a program using the hierarchy of employees in a university.
18. Use the above program to illustrate the super and final keywords.
19. Learn the effect of access modifiers while using inheritance. Write programs to that effect.
20. Write a program to understand polymorphic invocation of methods, while overriding the methods. Use an employee base class and manager sub class; override the computeSalary() method to illustrate the concept.
21. Demonstrate the use of abstract classes. Write a Person abstract class and then subclass that into Student and Faculty classes. Use appropriate fields and methods.
22. Write a program to demonstrate the usage of interfaces.
23. Write a program which shows the concept of interface extension.
24. Use a program to show the advantages of inheriting from multiple interfaces.
25. Write a program to understand the full capability of String class. Implement as many methods as required. Consult API documentation to read through the methods.
26. Write programs using StringBuffer and StringBuilder library classes.
27. Write a program to demonstrate the command-line arguments.
28. Develop an application that uses inheritance. Use the class Account and then subclass it into different account types. Then making use of Customer and Employee classes to develop the application to reflect the nature of banking operations. Use minimum operational sequence.
29. Write a program to demonstrate the usage of try and associated keywords. Introduce bugs into the program to raise exceptions and then catch and process them.
30. Learn how to throw an exception from your method, when an exception arises.
31. Learn how to create and use custom exceptions.
32. Experiment on using various methods of Throwable and Exception classes.

33. Practice on chaining the exceptions.
35. Using byte streams, write a program to both read from and write to files.
36. Using FileReader and FileWriter, write a program to perform file copying and any other suitable operations.
37. Write a Java Program that displays the number of characters, lines and words in a text file.
38. Use the classes StringTokenizer, StringReader and StringWriter to write a program to find the capabilities of these classes.
39. Write a program using the object streams.
40. Write a program to show the power of Serialization.
41. Write a program to check the characteristics of a file after getting the filename form the user.
42. Write programs to find the usage of other stream classes. Consult API documentation.
43. Write a program to demonstrate enumerations.
44. Write a program to understand the usage of assertions.
45. Demonstrate assertions through simple programs.
46. Use jar command and understand the various command options.
47. Use an applet to demonstrate deployment of an application.
48. Demonstrate Applet life cycle
49. Explore the applet concept.
50. Write programs to illustrate the use of Thread class and Runnable interface.
51. Write a program to show the assignment of thread priorities.
52. Write a program to synchronize threads. Use any problem to illustrate the concept.
53. Use the core methods of Thread class to write a program to learn the nature of execution of threads.
54. Write a program to design a frame and control its various display properties.
55. Write a program to understand nested classes.
56. Write a program to understand key events and mouse events.
57. Write a program to understand adapter classes.
58. Write programs to understand the usage of swing widgets.
59. Write a program to understand the usage of tool tips, file choosers.
60. Write a program to demonstrate any layout manager. Use a suitable application.
61. Write a program to demonstrate other core swing widgets.
62. Explore using the other layout managers.
63. Write a program to attach menus to a window.
64. Write a program using Tables.
65. Practice on using a progress bar.
66. Explore on using Trees.
67. Develop a standalone application using a Banking (or any other) enterprise as a base concept. Use appropriate classes and interfaces and develop a swing-based GUI application to present

the activities of the organization. Use only simple activities to demonstrate the usage of IO streams, multithreading and exception handling concepts. Keeping in view this application goals, use appropriate practice programs in the previous weeks.

Course Outcomes:

At the end of the course a student should be able to

CO 1 : Implement object-oriented concepts like encapsulation, data hiding and abstraction using programming constructs offered by java language.
CO 2 : Develop java programs to realize the power of inheritance, interfaces and packages.
CO 3 : Develop java programs to demonstrate the concepts of exception handling and I/O streams.
CO 4 : Implement java applications using multithreading mechanism.
CO 5 : Use graphical user interfaces to create Applets & Swing applications for providing solutions to real world problems.

References:

1. Java: The Complete Reference, Herbert Schildt, 10th Edition, McGraw-Hill Education, Oracle Press, 2017.
2. Head First Java, Kathy Sierra and Bert Bates, 2nd Edition, O'Reilly Media, 2005.

65232

PHYTHON PROGRAMMING LAB

Instruction	: 2 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 1	End Exam Duration	: 3 Hours

Course Objective:

1. To train how to write, test, and debug simple Python programs.
2. To train solving problems through branching and iterative model.
3. To teach usage of functions for structuring Python programs.
4. To make student to handle compound data using Python lists, tuples, dictionaries.
5. To make students to handle data in file processing.

Lab Problems:

1. Write a program to implement all types of print options for different type data.
   Write a program to implement arithmetic operators on various data types. The data should be obtained from user.
2. Write a program to implement Boolean Expression Computation.
3. Write a program to compute sum, difference, multiplication, division and conjugate of complex numbers received from the user.
5. Write a python script to read an integer mark from the console and test the mark to determine the grade obtained by a student with various mark range using conditional statement 'if'.
6. Write a python scripts to calculate the cumulative sum of all integers between two given values using iterative method ('for' loop).
7. Write a python program to check given number is prime or composite using iterative (for loop) method.
8. Write a python program to compute GCD of two given positive numbers using iterative method (while loop).
9. Write a program to read a string and apply various string methods.
10. Write a program to find the maximum and minimum values given a list of numbers.
11. Create tuples to hold strings for each of the items to be substituted (i.e., a tuple of nouns, one of verbs, etc.). These can then be accessed with a simple random integer and printed

out to form the sentences of the curse. The random integer generation and print out can be wrapped inside of a loop to generate several different curses.

12. Write a program to compute the various set operations.
13. Create a dictionary and write the script to apply all the dictionary operations.
14. Write a program to apply the various methods from the math module.
15. Write a function program to illustrate the various scope levels.
16. Write a function program to read the data from a file and count the total number of lines and words in the file.
17. Write a program to read data from a CSV file.
18. Write a function to find a value in a given list using binary search.
19. Write a function that accepts an integer and then computes all the factors of that number.
20. Write a program that asks the user to enter a word. Rearrange all the letters of the word in alphabetical order and print out the resulting word.
21. Write a program to find all numbers between 1 and 1000 that are divisible by 7 and end in a 6.
22. Find the square root of an integer using Newton's method.
23. Write a function program that reads to values n and k and computes the binomial coefficient.
24. Write a module program compute Binomial distribution and plot the distribution model for a given data set.
25. Implement Poisson distribution and plot the distribution model for a given data set.
26. Compute standard deviation, variance and other statistical parameters for a given data set.
27. Write a program to visualize the relationships between two sets of values, as a Venn diagram.
28. Write a program to find the correlation coefficient and rank between two data sets.
29. Write a program to implement rolling the dice.
30. Write a Python script of the classic game "Hangman". The word to guess is represented by a row of dashes. If the player guess a letter which exists in the word, the script writes it in all its correct positions. The player has 10 turns to guess the word. You can easily customize the game by changing the variables.
31. Write a python program to simulate a Tic-tac-toe game. Use Tkinter.
32. Write a program to implement the concepts of modules for any application.
33. Write a python program to create a Student class with student_id, name and age data attributes and implement the get and set methods for those attributes.

34. Write a program to demonstrate the concept of inheritance. Use an Employee class and Manager class with required attributes and methods.
35. Write a factorial function using recursion and find nCr and nPr values for a given n and r values.
36. Write a program to check if a given string is palindrome using recursive function.
37. Write a program to implement comprehensions their usage.
38. Write a program to illustrate the concept of generators.
39. Write a program to show how to simulate a stack , using lists.
40. Write a program to show how to simulate a queue, using lists
41. Write a program to show how to simulate a binary tree, using lists
42. Create a package with multiple modules and demonstrate how to use the package.
43. Write a program to use the datetime and time modules.
44. Write a program to compute matrix addition and multiplication using Numpy Package.
45. Write a program to use various collection objects using collections module.
46. Write a program, using the Ordered Dict and default dict objects.
47. Write a program to use various objects of the string module.
48. Write a program to generate N random data between two given data. Calculate the statistical measurement Mean, Median, Mode and other measurements. Use random package
49. Write a program to illustrate the various methods of the statistics module for simulated data.
50. Write a program to implement a miniature Monte Carlo simulator for a two-resistor voltage divider. It is suggested that you save this as Monte.py. This circuit will consist of a voltage source E and two resistors in series, first R1 and then R2. For simplicity, we shall assume that the voltage source is perfectly stable. The two resistors, however, will have a stated tolerance. We would like our program to simulate the action of picking two resistors from bins, which is, randomizing their values, and then determine the voltage across R2.

Course Outcomes:

At the end of the course, a student will be able to:

CO 1 : Design, code and test Python modules.
CO 2 : Understand and use standard library modules.
CO 3 : Apply object-oriented concepts to solve problems.
C0 4 : Write Python solutions to solve statistical problems.
C0 5 : Use advanced Python programming concepts.

References:

1. Think Python, Allen B Downey, Second Edition, O'Reilly, 2016.
2. Programming in Python 3, Mark Summerfield, Second Edition, Pearson Education, 2010.
3. Fundamentals of Programming Python, Richard L. Halterman, Southern Adventist University, January 18, 2018.

62233

ELECTRICAL AND ELECTRONICS ENGINEERING LAB

(Common to CSE, IT & ME)

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 1.5	End Exam Duration	: 3 Hours

Course Objectives:

1. Students analyze various electrical parameters and different theorems.
2. Students can understand working and operation of different electrical machines.
3. Students learn working and applications of various semiconductor devices.

List of the Experiments

Part-A

(Any 5 experiments from below given list)

1. Verification of Superposition with Resistive load
2. Verification of KCL & KVL principles
3. Magnetization characteristics of D. C Shunt generator.
4. Swinburne's Test on DC shunt machine
   (Predetermination of efficiency of a given DC shunt machine working as motor and generator).
5. O. C & S. C tests on Single-phase transformer.
   (Predetermination of efficiency and regulation at given power factors and determination of Equivalent circuit)
6. Brake test on 3-phase Induction motor (Performance Characteristics).

Part-B

(Any 5 experiments from below given list)

1. Zener diode characteristics
2. Rectifier without filters (HWR and FWR)
3. Rectifier with filters (HWR and FWR)
4. Bridge rectifier
5. Design of Voltage divider bias Circuit
6. Frequency response Common Emitter Amplifier

Course outcomes:

At the end of the course, students will demonstrate the ability to

CO 1 : Do simplification and Verification of various electrical circuits.
CO 2 : Conduct investigation on the experimental setup for various electrical circuits.
CO 3 : Study working and applications of various electrical machines.
CO 4 : To Analyze and Design Rectifier circuits.
CO 5 : To Design the basic transistor biasing techniques.

68232

VERBAL ABILITY LAB

(common to all branches)

Instruction	: 2 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 1	End Exam Duration	: 3 Hours

Course Objectives:

1. Students will be trained to become proficient in word formation, spellings, and vocabulary.
2. Students will develop linguistic competence through appropriate use of Idioms and Phrasal verbs.
3. Students will develop verbal reasoning through Verbal Analogy.
4. Students will be trained to identify the common errors in English and write grammatically correct sentences.
5. Students will develop professional writing skills through business letters.

The students will be given exercises covering the following topics:

1. Word Formation
2. Spelling
3. Synonyms and Antonyms
4. Homonyms, Homophones and Homographs
5. Collocations
6. One word substitutes
7. Idiomatic expressions
8. Phrasal Verbs
9. Verbal Analogy
10. Sentences
    a. Rearranging jumbled words to make meaningful sentences
    b. Identifying errors in sentences
    c. Correction of sentences
    d. Improvement of sentences
11. Writing Skills
    a. Paragraph writing (different kinds of paragraph writing)
    b. Essay writing
    c. Business Letter writing

The students will be tested on the use and application of all the topics mentioned in the syllabus in addition to their basic conceptual understanding.

Course Outcomes:

CO 1 : Students will be empowered in English language skills and meet the demands of the global work environment.
CO 2 : Students will have enriched vocabulary
CO 3 : Students will be proficient in answering reasoning based questions.
CO 4 : Students will develop the ability to write grammatically correct sentences.
CO 5 : Students will enhance their professional writing skills through business letters.

References:

1. Objective English, Edgar Thorpe & Showick Thorpe, S.Chand & Co., 2011.
2. A Modern Approach to Verbal Reasoning, R. S. Aggarwal, S.Chand & Co., 2011.
3. Barron's Essential Words for GRE, Philip Geer, Barron's Educational Series, 2011.
4. How to prepare for Verbal Ability and Reading Comprehension for the CAT, Arun Sharma and Meenakshi Upadhyay, Tata McGraw-Hill, 2011.
5. Word Power Made Easy, Norman Lewis, Goyal publishers & Distributors, 2011.
6. English Vocabulary in Use Advanced, Michael McCarthy, Cambridge University Press, 2008.

65251

ADVANCED DATA STRUCTURES THROUGH JAVA

(Common to CSE and IT)

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 3	End Exam Duration	: 3 Hours

Course Objectives:

1. To understand the importance of generic programming and Java's collection framework.
2. To implement various basic data structure like stacks, queues, linked lists etc using user defined generic classes and java's collection classes.
3. To learn various data structures for implementing dictionaries.
4. To understand pattern matching algorithms and tries.

Unit I - Generics:

Introduction to Generics, simple Generics examples, Generic Types, Generic methods, Bounded Type Parameters and Wild cards, Inheritance & Sub Types, Generic super class and sub class, Type Inference, Restrictions on Generics.

Unit II - 1D and 2D Collections:

1D Collection:1D Collection Interfaces: Collection, Set, List, NavigableSet, SortedSet, Queue, Deque. 1D Collection Classes-Hash Set, Linked HashSet, TreeSet, ArrayList, LinkedList.

2D Collection:2D Collection Interfaces-Map, NavigableMap, SortedMap, 2D Collection Classes-HashMap, LinkedHashMap, TreeMap.

Unit III - Dictionaries:

Introduction: Dictionary definition, Dictionary ADT.

Dictionaries implementation-I :

Linear List Representation: Basics of linear list, implementation of sorted list using user defined generic classes and, LinkedList Collections class.

Hashing: basics, closed hashing – linear probing, quadratic probing, double hashing, rehashing, extendible hashing and their implementation, open hashing-separate chaining and its implementation using user defined generic classes.

Binary Search Trees: definition and basics, implementation of operations-searching, non-recursive traversals, insertion and deletion using user defined generic classes.

Unit IV - **Dictionaries implementation-II:**

AVL Tree: definition, height of an AVL tree, representation, operations-rotations, insertion, and searching, deletion and their implementation using Java's Collection framework.

Red Black Binary search trees: definition, insertion, deletion and searching operations.

B-Trees: B-Tree of order m, height of a B-Tree, searching, insertion, deletion operations.

Unit V - Priority Queues and Pattern Matching:

Priority Queue: definition, max and min heaps, realizing priority queues using heaps, operations-insertion, deletion and their implementation using user defined generic classes, heap sort and its implementation using user defined generic classes.

Pattern Matching: Introduction, Brute Force algorithm, Boyer Moore algorithm,
Knuth-Morris-Pratt algorithm and their implementation.

Tries: Standard Tries, Compressed Tries, Suffix tries.

Course Outcomes:
At the end of the course, student should be able to

CO1 : Realize the power of generics in java.
CO2 : Understand Java's Collection class hierarchy and implement stacks, queues, dictionaries and trees using them.
CO3 : Implement dictionaries using linear lists and hashing and compare their performances.
CO4 : Implement dictionaries using various height balanced trees and also analyze the advantages and disadvantages of height balanced trees.

CO5 : Evaluate various pattern and word matching algorithms in terms of their complexity and efficiency.

Text Books:

1. Java: The Complete Reference, Herbert Schildt, 10th Edition, McGraw-Hill Education, Oracle Press, 2017. (Units I and II)
2. Data Structures and Problem Solving Using Java, Mark A. Weiss, 4th Edition, Pearson Education, 2009. (Units III and IV)
3. Data Structures and Algorithms in Java, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, 6th Edition, Wiley Publications, 2014.(Unit V)

References:

1. Data Structures, Algorithms, And Applications In Java, SartajSahni, 2nd Edition, Universities Press, 2005
2. Data Structures: Abstraction and Design Using Java, Elliot B. Koffman, Second Edition, Wiley publications, January 2010.
3. Head First Java, Kathy Sierra and Bert Bates, 2nd Edition, OREILLY publications., 2005

67251

COMPUTER ORGANIZATION

(Common to CSE, IT)

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 3	End Exam Duration	: 3 Hours

Course Objectives:

1. To develop thorough analyzing capabilities of complexities in implementation of multiplication and division algorithms and their realization through hardware design.
2. To make students adapt the basic notations of Register Transfer Language and mnemonics suitable for simplification of assembly language instructions.
3. To transfer design and implementation knowledge of basic functional units of CPU based on ISA.
4. To design an efficient computer system through cache, pipeline and multiprocessor machines.

UNIT I - Basic Structure of Computers

Computer Types, Functional unit, Basic Operational concepts, Bus structures, Software, Performance, Reduced Instruction set computer, Data Representation. Fixed Point Representation. Floating – Point Representation, Computer Arithmetic: Multiplication and Division Algorithms.

UNIT II – Register Transfer Language and Micro-operations

Register Transfer language. Register Transfer Bus and memory transfers, Arithmetic Micro operations, shift micro operations, Arithmetic logic shift unit, Instruction codes. Computer Registers, Computer instructions – Instruction cycle, I/P-O/P and Interrupt.

Instruction Sets: STACK organization, Instruction formats, Addressing modes, DATA Transfer and manipulation, Program control, IA-32 Architecture and instruction set.

UNIT III –Mcro programmed Control

Control memory, Address sequencing, microprogram example, design of control unit, Microprogrammed control.

The Memory System: Basic concepts of semiconductor RAM memories, Read – only memories, Cache memories, performance considerations, virtual memories, secondary storage.

UNIT IV – Input-Output Organization

Peripheral Devices, DMA, Input – Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt, PCI Bus.
Pipeline and Parallel Processing: Parallel processing, Flynn's classification, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline.

UNIT V – Multiprocessors

Characteristics of Multiprocessors, Vector Processing, Array Processors, Interconnection Structures, Inter processor Arbitration, Inter Processor Communication and Synchronization ,Cache Coherence, Shared Memory Multiprocessors.

Course Outcomes:

At the end of the course, the student should be able to

CO1 : Represent floating point number according to IEEE standard and realize booths algorithm for multiplication and division operations on integers.
CO2 : Figure out functional units of the processor such as registers and arithmetic – logical unit, instruction execution timing, bus operation, addressing modes, instruction formats and have basic understanding of assembly language programming.
CO3 : Attain the knowledge of micro programming and understand the concepts of memory.
CO4 : Demonstrate the basics of the system topics: Multi stage instruction pipeline and hazards, super scalar and Parallel architectures.
CO5 : Discuss the vector processing and parallelism in terms of a single processor and multi-processors.

Text Books:

1. Computer Systems Architecture, M.Morris Mano, 3rd Edition, Pearson/ PHI, 2007.
2. Computer Organization, CarlHamacher, ZvonksVranesic, SafeaZaky, V Edition, McGraw-Hill, 2017.

References:

1. Computer Organization and Architecture , William Stallings 6th Edition, Pearson/PHI.
2. Structured Computer Organization, Andrew S.Tanenbaum, 4th Edition Pearson/PHI.
3. Fundamentals or Computer Organization and Design, Sivaraama Dandamudi, 1st Edition Springer, 2003.

67252

OPERATING SYSTEMS

(Common to CSE and IT)

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 3	End Exam Duration	: 3 Hours

Course Objectives:

1. To acquire knowledge on basic concepts, principles, functions of operating systems, scheduling and synchronization.
2. To learn different memory management techniques and their implementations.
3. To analyze and understand process synchronization and deadlock algorithms.
4. To gain knowledge about secondary storage structures, file systems and their implementations.
5. To understand different protection and security threats.

Unit I - Operating System Overview, Process Management

Operating System Overview: Operating system functions and services, Overview of computer operating systems, distributed and special purpose systems, System calls and system programs, Operating system structure.
Process Management: Process concepts, Threads, scheduling-criteria, Scheduling algorithms (FCFS, SJF, Priority), Scheduling algorithms (RR, Multilevel queue, Multilevel feedback queue).

Unit II – Synchronization, Deadlocks

Synchronization: The critical- section problem and Peterson's solution, Synchronization hardware, Semaphores, Classic problems of synchronization, Monitors.
Deadlocks: Deadlock characterization, Deadlock prevention, Deadlock avoidance (Banker's algorithm), Deadlock detection and recovery

Unit III - Memory-Management Strategies, Virtual-Memory Management

Memory-Management Strategies: Contiguous memory allocation, Paging, Structure of the page table, Segmentation.
Virtual-Memory Management: Virtual memory and demand paging, Introduction to page replacement& page replacement algorithms (FIFO, Optimal). (LRU, LRU variations, Counting based), Allocation of frames and thrashing.

Unit IV - File System, System Implementation, Mass-Storage Structure

File System: The concept of a file and access methods, Directory structure, File sharing and protection.

File System Implementation: File system structure, file system implementation and directory implementation. Allocation methods, Free-space management, efficiency and performance

Mass-Storage Structure: Overview of mass-storage structure, Disk scheduling, RAID structure

Unit V - I/O Systems, Protection, Security

I/O Systems: I/O hardware, Application I/O interface, kernel I/O subsystem.

Protection: Goals and principles of protection, domain of protection access matrix. Implementation of access matrix, revocation of access rights.

Security: The Security problem, program threats, system and network threats. Cryptography as a security tool. User authentication, implementing security defenses. Firewalling to protect systems and networks, computer – security classifications.

Note: Case studies: Linux and Windows.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Acquire basic knowledge about different functions, structures and design features of contemporary operating systems, to aid in applying and solving complex engineering problems.

CO 2 : Analyze and formulate the problems, design the solutions, implement and demonstrate different process scheduling, Synchronization, and deadlock related algorithms.

CO 3 : Design algorithmic experiments, analyse and interpret the data for different memory management techniques and their implementations.

CO 4 : Differentiate and Demonstrate secondary storage structures, file systems, directory structures and their implementations. Select and apply appropriate disk scheduling algorithms to predict the performance.

CO 5 : Understand the impact of the different protection and security threats in the context of social, safety, ethical, legal issues and design solutions and mechanisms including responsibilities to overcome them.

Text Books:

1. Operating System Concepts, Abraham Silberchatz, Peter B. Galvin and Greg Gagne, 8th Edition, John Wiley, 2008.
2. Operating Systems- A Concept Based Approach, D.M.Dhamdhere, 3rd Edition, TMH, 2009.

References:

1. Operating Systems, Internals and Design Principles, William Stallings, 6th Edition, Pearson Education, 2009.
2. Modern Operating Systems, Andrew S Tanenbaum, 2nd Edition, PHI, 2008.
3. Operating Systems, A.S. Godbole, 2nd Edition, TMH, 2008.

65252

DATABASE MANAGEMENT SYSTEMS

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 3	End Exam Duration	: 3 Hours

Course Objectives:

1. To introduce the role of database management system in an organization.
2. To represent real world scenario using E-R diagrams.
3. To modal the database using relations avoiding redundancies.
4. To learn transaction management and concurrency protocols to ensure data consistency.
5. To understand query optimization and PL/SQL concepts.

Unit I

Introduction to DBMS
History of DBMS, Concepts and overview of DBMS, Data models - ER model, Relational model, Levels of Abstraction in DBMS, Data base Languages, Architecture of DBMS, Data Base Users and Administrators.

ER-Model
Data base design and ER model, ER modeling Constructs, Additional features of ER Model, Class Hierarchies, Aggregation, Conceptual Design with ER model, Case study: ER design for Large Enterprises.

Unit II

Relational Algebra and Calculus
Introduction to relational model, Logical Database Design- ER to Relational, Relational Algebra - Selection and Projection, Set operations, Renaming, joins, Examples of Relational Algebra Relational Calculus- Tuple relational Calculus, Domain relational calculus.

Introduction to Structured Query Language
Form of Basic SQL Query, Examples of Basic SQL Queries, Introduction to Nested Queries, Correlated Nested Queries, Set Comparison Operator-Aggregate Operators, NULL values and Comparison using Null values ,Logical connectivity's – AND, OR and NOT, OUTER Joins, Disallowing NULL Values.

Unit III

Schema Refinement
Introduction to schema refinement, Problems caused by decomposition, Functional dependencies (FDs) and reasoning about FDs, Normal Forms (NF), Properties of Decomposition, Schema Refinement in Data Base Design, Case studies using Normal Forms

Unit IV

Transaction Management- Transaction concept & state, Implementation of atomicity and durability, Concurrent executions of transaction, Serializability and Recoverability, Implementation of Isolation, Testing for serializability, Lock-Based Protocols, Graph Based Protocol, Timestamp-Based Protocols, Validation-Based, Protocols, Multiple Granularity.

Unit V – Database Recovery System and PL/SQL

Recovery and Atomicity, Log based Recovery, Recovery with concurrent transaction.

Query Optimization
Cost based query optimization, Estimation of plan cost.

PL/SQL
Basics of PL/SQL, Cursors, Procedures, Functions and Triggers.

Course Outcomes:

At the end of the course, student should be able to

CO 1 : Demonstrate an understanding of database management system components and features. Design E-R Model to represent real-world database application scenarios.
CO 2 : Demonstrate mathematical approach towards querying database using relational algebra and relational calculus, and implement using SQL.
CO 3 : Convert E-R Model to relational Model and design proper relational database while eliminating anomalies.
CO 4 : Demonstrate the role of transaction management and concurrency control protocols.
CO 5 : Demonstrate an understanding of query optimization techniques and apply PL/SQL concepts for database manipulations with constraints.

Text Books:

1. Database System Concepts, A.Silberschatz, H.F. Korth, S.Sudarshan, 6th edition, McGraw-Hill, 2006.
2. Database Management Systems, Raghu Ramakrishnan, Johannes Gehrke, 3rd Edition, TMH,2003.

References:

1. Fundamentals of Database Systems, Ramez Elmasri, Shamkant B.Navathe, 7th edition, Pearson Education,2008.
2. Database Systems: The Complete Book by Hector Garcia-Molina, Jeffery D.Ullman, Jennifer Widom, 2nd Edition, Pearson Education, 2008.
3. Database Management System Oracle SQL and PL/SQL, P.K.Das Gupta, 2nd edition, PHI, 2013.

65253

AUTOMATA AND FORMAL LANGUAGES

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 3	End Exam Duration	: 3 Hours

Course Objectives:

1. To introduce central ideas of theoretical computer science from the perspective of formal languages.
2. To classify theoretical machines by their power to recognize languages.
3. To understand the computational complexity of deterministic and non-deterministic machines.
4. To distinguish between decidability and undecidability.

Unit – I

Introduction to Finite Automata, Central Concepts of Automata Theory – Alphabets, Strings, Languages, Sample Problems.

Deterministic Finite Automata, Nondeterministic Finite Automata, An application: Text Search, Finite Automata with Epsilon-Transitions.

Unit – II

Equivalence between NFA with and without Epsilon–transitions, NFA to DFA conversion, Minimization of FSM's, Equivalence between two FSM's, Finite Automata with Output: Moore and Melay machines

Regular Expressions, Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Properties of Regular Languages, Pumping Lemma for Regular Languages, Applications of the Pumping Lemma.

Unit – III

Regular Grammar: Right Linear grammar and Left Linear grammar, Equivalence between Regular Grammar and FA.

Context-Free Grammars: Definition of Context-Free Grammar, Derivations Using a Grammar, Leftmost and Rightmost Derivations, The Language of a Grammar, Sentential Forms, Parse Tress, Applications of Context-Free Grammars, Ambiguity in Context- Free Grammars, Minimization of Context- Free Grammar.

Normal Forms for Context- Free Grammars(Chomsky Normal Form, Griebach Normal Form), The Pumping Lemma for Context-Free Languages, Closure Properties of Context-Free Languages.

Unit – IV

Push Down Automata: Definition, The Language of a PDA, Deterministic Pushdown Automata, Equivalence between PDA and CFL.
Introduction to Turing Machines, Problems that Computers Cannot Solve, The Turing Machine definition, Model and Design, Programming Techniques for Turing Machines, Types of Turing Machine.

Unit – V

Chruch's Hypothesis, Counter machine. Chomsky Hierarchy of Languages.

Undecidability: A Language that is Not Recursively Enumerable, An Undecidable Problem that is RE, Undecidable Problems about Turing Machines, Post's Correspondence Problem, Other Undecidable Problems, Intractable Problems, The Classes: P, NP, NP-Complete and NP- Hard Problem.

Course Outcomes:

Upon completion of the course, a student should be able to

CO 1 : Model and implement the NFA and DFA for recognizing the regular languages.

CO 2 : Realize the equivalence of recognition capabilities of DFA and NFA.

CO 3 : Model the context free language generation using CFG and characterize them using Normal forms.

CO 4 : Solve the context free language recognition problem through constructing PDA.

CO 5 : Characterize the formal languages based on their expressive power and the complexity involved in developing recognizers.

Text Books:

1. Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E.Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education, 2011.

2. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Schand, 2014.

References:

1. Introduction to Languages and The Theory of Computation, John C Martin, 3rd Edition, TMH 2007.
2. Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R, 1st Edition, Pearson 2009.
3. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekaran, 3rd, edition, PHI, 2012.

65281

ADVANCED DATA STRUCTURES THROUGH JAVA LAB

(Common to CSE and IT)

Instruction	:	3 Periods/week	Sessional Marks		: 30
Tutorial	:	-	End Examination Marks		: 70
Credits	:	1.5	End Exam Duration		: 3 Hours

Course Objectives:

1. To implement generic programming and Java's collection framework.
2. To apply Java's collection framework for implementing basic data structures like stacks, queues, linked list ,etc.
3. To implement dictionaries using advanced data structures like Binary search trees, and AVL trees.
4. To implement pattern matching algorithms.

Lab problems:

1. Write a java program to demonstrate the use of bounded type parameters and wild card arguments.

2. Write a java program to implement iterators on Array List and LinkedList.

3. a) Implement a Generic stack to deal with Integer, Double and String data using user defined arrays and linked lists.
   b) Implement a Generic queue to deal with Integer, Double and String datauser defined arrays and linked lists.

4. a) Write a Java program to implement Generic stack using Array List Collection class.
   b) Write a Java program to implement Generic stack using LinkedList Collection class.

5. a) Write a Java program to implement Generic queue using ArrayList Collection class.
   b) Write a Java program to implement Generic queue using LinkedList Collection class.

6. Write a Java program to demonstrate the use of following Collection classes.
   a. HashSet b. LinkedHashSet c. TreeSet

7. Write a java program to create a class called Person with income, age and name as its members. Read set A of persons from a user and compute the following sets:

   i) Set B of persons whose age > 60
   ii) Set C of persons whose income < 10000 and iii) B ∩ C

8. Write a Java program to demonstrate the use of following Collection classes.
   a. HashMap b. LinkedHashMap c. TreeMap

9. Write a Java program to implement Sorted Chain.
10. Write a Java program to implement Separate Chaining
11. Write a Java program to implement Linear Probing.
12. Implement BST using Collection API, use recursive procedures to implement inOrder, preOrder and postOrder traversals.
13. Implement AVL tree using Collection API.
14. Implement priority queues with max Heap tree using Collection API.
15. Implement heap sort with max Heap tree using Collection API.
16. Implement Boyer Moor algorithm.
17. Implement Knuth Morris Pratt algorithm.

Course Outcomes:

At the end of the course, student should be able to

CO 1 : Implement stacks and queues using user defined generic classes.
CO 2 : Implement hashing, sets, stacks and queues using collection classes in java.util package.
CO 3 : Implement dictionaries using various data structures like sorted list, and hashing
CO 4 : Implement dictionaries using various height balanced trees and also analyze the advantages and disadvantages of height balanced trees.
CO5 : Implement Pattern Matching Algorithms

References:

1. Java: The Complete Reference, Herbert Schildt, 10th Edition, McGraw-Hill Education, Oracle Press, 2017. (Units I and II)
2. Data Structures and Problem Solving Using Java, Mark A. Weiss, 4th Edition, Pearson Education, 2009. (Units III and IV)
3. Data Structures and Algorithms in Java,Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, 6th Edition, Wiley Publications, 2014.(Unit V)

65282

MOBILE APPLICATION DEVELOPMENT LAB

Instruction	: 3 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 1.5	End Exam Duration	: 3 Hours

Course Objectives:

1. To install Android OS on to the device.
2. To develop mobile applications using ANDROID.
3. To publish the android applications on to Google play store.

1. **Introduction**
   a. Hold a discussion on the architectural features of Android. Address the different components of Android.
   b. Use of Android Studio for the development of android application.
   c. Discuss the complete anatomy of the android application and the gradle tool.
   d. Sample Hello World! Application demonstration both in AVD (Android Virtual Device) and on a real device (Mobile or Tab).

2. **Flash and XML**
   a. Demonstrate the installation of custom ROM on to the Android device.
   b. XML – eXtensible Markup Language introduction.
      i. Sample XML document creation.
      ii. Attributes and name spaces explanation
      iii. Sample UI Demonstration using XML for Hello World! Application.

3. **Activity and Intent**
   a. Life cycle of an Activity and Fragment
   b. Intent discussion.
   c. Lifecycle demonstration with multiple activities (use Toast and Logcat)
   d. Create a splash screen to welcome a user to your application.

4. **Views & View Groups**
   Develop an android application to illustrate the use of various View Groups in android.

5. **Dialog Windows Android application to demonstrate the following dialog windows.**
   Alert Dialog Progress Dialog Date Picker and Time Picker Calendar

6. **Adapter Views**

Demonstrate the use of the following Adapter Views

1 – ListView 2 – GridView 3 – Gallery 4 – Spinner

5 – Auto Complete TextView

7. **Menus**

Create an android application to demonstrate the use of Option and Context Menu in android.

8. **Persistent Storage – 1**

a. Create a Registration form using various views. Store the details of the user using Shared Preferences.
b. Create a login screen. Use the data stored with Shared Preferences to authenticate the user.

9. **Persistent Storage – 2**

a. Demonstrate the use of File Storage (Internal) by an application that pushes the contents of different views to a file and then read them back when required.
b. Demonstrate the above application with the help of External Storage.

10. **Persistent Storage – 3**

Create an application to demonstrate the use of SQLite database. Perform the CURD operations on the contents of the database.

Use necessary screens to display the information.

11. **Deployment / Publishing**

Demonstrate the deployment process of .apk file to android device and to the android market / play store.

12. **Advanced Topics – Theoretical Explanation**

a. Content Providers
b. SMS Manager using Broadcast Receiver
c. Inclusion of Maps in the application
d. Using REST API and JSON object.

13. **Case Study**

Divide the students into a group of 6 each and assign each group with a small application to develop.

Develop an end – to – end application using the various features of android discussed in the earlier lab sessions.

Make use of the necessary screens to model the application

Use the required type of data storage mechanism for the application.

Course Outcomes:

At the end of the course a student should be able to:

CO 1 : Flash the device and upgrade to the latest version that is well supported.
CO 2 : Design and develop User Interfaces for Android applications.
CO 3 : Demonstrate their knowledge in developing Mobile Application for various domains.
CO 4 : Deploy and publish the applications to the device and also to the android market.
CO 5 : Understand and apply the features like maps api, messaging and content providers in developing various android apps.

References:

1. Head First Android Development, David Griffiths, Dawn Griffiths, 2nd Edition, O'Reilly Media, Inc., 2017.
2. Beginning Android 4 Application Development, Wei – Meng Lee, Wrox Publications, 2012.
3. Android Wireless Application Development, Shae Conder and Lauren Darcey, 2nd Edition, Addison – Wesley Professional, 2012.

65283

DATABASE MANAGEMENT SYSTEMS LAB

Instruction	: 2 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 1	End Exam Duration	: 3 Hours

Course Objectives:

1. To understand the relational model.
2. Analyze database requirements and determine the entities involved in the system and their relationship to each other.
3. Understand logical design of the database modeling concepts such as E-R diagrams.
4. Demonstrated SQL DML/DDL commands to insert and manipulate the database.
5. Understand procedures, functions and triggers in PL/SQL.

Database Description: This lab enables the students to practice the concepts learnt in the subject DBMS by developing a database for an example "**Boat reservation by the sailor**"and "**employee data maintenance in an organization**" whose description is as given below. The student is expected to practice the designing, developing and querying a database in the context of reserving a boat and employee data maintenance. Students are expected to use "MySql" database.

"Boat reservation by the sailor" is a schema with several boats which could be reserved depending on color and availability on a particular day. The sailor reserves the boat on a particular day y registering himself with a rating. The sailor is identified by sailor id, boats are identified by boat id and reservation is uniquely identified by sailor id, boat id and day.

"Employee data maintenance in an organization": In any organization, we need to maintain the data of employees categorized into department as per the salary. The scheme contains employee, department and sal grade tables which are identified by employee id, department id and range of salary respectively.

1. E-R Model

Analyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc. Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Boat reservation by the sailor:

Entities:

1. SAILORS
2. BOATS
3. RESERVES

PRIMARY KEY ATTRIBUTES:

1. SID (SAILOR ENTITY)
2. BID (BOATS Entity)
3. SID,BID,DAY (RESERVES ENTITY)

Employee data maintenance in an organization

Entities:

1. EMPLOYEE
2. DEPT
3. SALGRADE

PRIMARY KEY ATTRIBUTES:

1. EID (EMPLOYEE ENTITY)
2. DID (DEPT Entity)
3. LOWSAL AND HIGHSAL (SALGRADE ENTITY)

2. Concept design with E-R Model

Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc wherever required for

1) Boat reservation by the sailor

2) Employee data maintenance in an organization

3. Relational Model

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality. Represent attributes as columns in tables or as tables based on the requirement. Different types of attributes (Composite, Multi valued, and Derived) have different way of representation.

SAILORS

SID	SNAME	RATING	AGE

EMPLOYEE

EID	ENAME	DID	SAL	DESIGNATION	MGRNUM	DOJ	AGE

4. Normalization

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

Perform do the second and third normal forms for sailors and Employee databases if required.

5. Installation of Mysql and practicing DDL commands

Installation of MySql. In this week student will learn Creating databases, How to create tables, altering the database, dropping tables and databases If not required. Students will also try truncate, rename commands etc.

6. Practicing DML commands

DML commands are used to for managing data within schema objects. Some examples:

1.SELECT - retrieve data from the a database

2.INSERT - insert data into a table

3.UPDATE - updates existing data within a table

4. DELETE - deletes all records from a table, the space for the records remain

7. Querying - I

In this week students are going to practice queries (along with sub queries) using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

8. Querying - II

Students are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

9. Triggers

In this week students are going to work on Triggers. Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database.

10. Procedures

In this session students will learn Creation of stored procedure, Execution of procedure and modification of procedure. Practice procedures using the above database.

11. Cursors

In this week students will learn to declare a cursor that defines a result set. Open the cursor to establish the result set. Fetch the data into local variables as needed from the cursor, one row at a time. Close the cursor when done

Course Outcomes:

At the end of the course, student should be able to

C0 1 : Analyze database requirements and determine the entities involved in the system and their relationship to each other.
C0 2 : Design E-R Model to represent database application scenarios.
CO 3 : Convert/transform the E-R Model to relational tables, populate relational database and formulate SQL queries on data.
C0 4 : Improve the database design by normalization.
C0 5 : Implement PL/SQL procedures, function, triggers and cursors.

References:

1. Database Management Systems, Raghu Ramakrishnan, Johannes Gehrke, 3rd Edition, TMH, 2003.
2. Introduction to SQL, Rick F.VanderLans, 4th edition, Pearson education, 2007.
3. Oracle PL/SQL, B.Rosenzweig and E.Silvestrova, 2nd edition, Pearson education, 2003.

68283

REASONING AND DATA INTERPRETATION LAB

Instruction	: 2 Periods/week	Sessional Marks	: 30
Tutorial	: -	End Examination Marks	: 70
Credits	: 1	End Exam Duration	: 3 Hours

Course Objectives:

1. To train the students to face the questions that require reasoning and interpretation of data with greater facility and help them face requirement tests and entrance examinations for all courses of higher education successfully.
2. To develop the use of analytical, reasoning and logical skills in formal and informal situations.
3. To introduce Graphs, Charts, problem solving with Data, Puzzles and logical Questions.
4. To train the students towards preparation for placement, Competitive examinations like GATE, CAT, GRE etc.

CONTENTS:

Exercises/Experiments on the following topics will be done during the course with necessary illustrations.

1. Classification of data
2. Coding and Decoding
    a. Letter coding
    b. Number coding
    b. Split by Half coding
    d. Coded Inequality
3. Series
    a. Number
    b. Letter
    c. Alpha Numeric
    d. Alphabet Test
4. Direction Sense
    a. Fictious Symbol
    b. Direction Test
5. Cubes
6. Blood Relations
7. Ratios, Percentages and Averages
8. Syllogism
9. Data Sufficiency
    a. Analytical Decision Making
    b. Input-Output
10. Data Visualization / Interpretation
    a. Bar Charts
    b. Line Graphs
    c. X-Y Charts

d. Pie charts
e. Tables

11. Analogy
    a. Number
    b. Letter
    c. Word
12. Puzzeles
    a. Simple Table &Comparisions Problems
    b. Circle sitting & Row/Column sitting Problems

Course Outcomes: At the end of the course, student should be able to

CO 1 : Understand the concepts of Statement-Argument, Assumption and Course of Action and use reasoning as a tool to match statements with arguments etc.
CO 2 : Look at data and find links and patterns, link data with conclusions and study data logically.
CO 3 : Study problem situations and use reasoning as a tool to find solutions.
CO 4 : Nurture the ability to use reasoning as a skill in real time problems solving.
CO 5 : Analyze and infer the data with respect to trend and case based.

Text Books:

1. How to prepare for Data Interpretation for CAT, Arun Sharma, McGraw-Hill.
2. Modern Approach to Verbal and Nonverbal Reasoning, R.S. Aggarwal, S Chand.

References:

1. Quantitative Aptitude, R.S Aggarwal, S. Chand.
2. A Modern Approach to Logical Reasoning, R.S Aggarwal, S.Chand.
3. Reasoning & Aptitude for GATE & ESE, MADE EASY Publications.

68282

GENDER SENSITIZATION

(Common to All Branches)

Instruction: 1 Periods/week Sessional Marks :100
Credits :0

Course Objectives:

1. To develop student's sensibility with regard to issue of gender in contemporary India.
2. To provide a critical perspective on the socialization of men and women.
3. To introduce students to information about some key biological aspects of genders.
4. To expose the students to debates on the politics and economics of work.
5. To help students reflect critically on gender violence.
6. To expose students to more egalitarian interactions between men and women.

Unit I – Understanding Gender:

Gender: Why Should We Study It? **(Towards a World of Equals: Unit -1)**
Socialization: Making Women, Making Men **(Towards a World of Equals: Unit -2)**
Introduction. Preparing for Womanhood. Growing up Male. First lessons in Caste. Different Masculinities.
Just Relationships: Being Together as Equals **(Towards a World of Equals: Unit-12)**
Mary Kom and Onler. Love and Acid just do not Mix. Love Letters.
Mothers and Fathers. Further Reading: Rosa Parks- The Brave Heart.

Unit II - Gender and Biology:

Mission Women: Sex Selection and Its Consequences **(Towards a World of Equals: Unit-4)**
Declining Sex Ratio. Demographic Consequences.
Gender Spectrum: Beyond the Binary **(Towards a World of Equals: Unit-10)**
Two or Many? Struggles with Discrimination.
Additional Reading: Our Bodies, Our Health **(Towards a World of Equals: Unit-13)**

Unit III - Gender and Labour:
Housework: the Invisible Labour**(Towards a World of Equals: Unit-3)**
"My Mother doesn't Work." "Share the Load".
Women's Work: Its Politics and Economics **(Towards a World of Equals: Unit-7)**
Fact and Fiction. Unrecognized and Unaccounted work. Further Reading: Wages and Conditions of Work.

Unit IV - Issues of Violence:

Sexual Harassment: Say No! **(Towards a World of Equals: Unit-6**)
Sexual Harassment, not Eve-teasing-Coping with Everyday Harassment-Further Reading: " Chupulu"
Domestic Violence: SpeakingOut (Towards a World of Equals: Unit-8)
Is Home a Safe Place? When Women Unite (Film). Rebuilding Lives. Further Reading: New Forums for justice.
Thinking about Sexual Violence (Towards a World of Equals: Unit-11)
Blaming the Victim – "Fought for my life…"-Further Reading: The Caste Face of Violence.

Unit V - Gender Studies:

Knowledge: Through the Lens of Gender (Towards a World of Equals: Unit-5)
Point of View. Gender and the Structure of Knowledge. Further Reading: Unacknowledged Women Artists of Telangana.
Whose History? Questions for Historians and Others (Towards a World of Equals: Unit-9)
Reclaiming a Past. Writing other Histories. Further Reading: Missing Pages from Modern Telangana History.

Essential Reading: All the Units in the Textbook, "Towards a World of Equals: A Bilingual Textbook on Gender" written by A.Suneetha, Uma Bhrugubanada, DuggiralaVasanta, Rama Melkote, Vasudha Nagaraj, Asma Rasheed, GoguShyamala, Deepa Sreenivas and Susie Tharu.

Note: Since it is interdisciplinary Course, Resource Persons can be drawn from the fields of English Literature or Sociology or Political Science or any other qualified faculty who has expertise in this field.

Course Outcomes: At the end of the Course student will be able to:

CO 1: Students will have developed a better understanding of important issues related to gender in contemporary India.
CO 2: Students will be sensitized to basic dimensions of the biological, sociological, psychological and legal aspects of gender. This will be achieved through discussion of materials derived from research, facts, everyday life, literature and film.
CO 3: Students will attain a finer grasp of how gender discrimination works in our society and how to counter it.
CO 4: Students will acquire insight into the gendered division of labour and its relation to politics and economics.
CO 5: Men and women students and professionals will be better equipped to work and live together as equals.
CO 6: Students will develop a sense of appreciation of women in all walks of life
CO 7: Through providing accounts of studies and movements as well as the new laws that provide protection and relief to women, the textbook will empower students to understand and respond to gender violence.

Text Books:

1. Sen, Amartya. "More than One Million Women are Missing. "New York Review of Books 37.20 (20 December 1990). Print "We Were Making History……." Life Stories of Women in the Telangana People's Struggle. New Delhi: Kali for Women, 1989
2. TriptiLahiri. "By the Numbers: Where Indian Women Work." Women's Studies Journal (14 November 2012) Available online at: http://blogs.wsj.com/India real time/2012/11/14by-thenumbers- where-Indian-women-work
3. K. Satyanarayana and Susie Tharu (Ed.) Steel Nibs Are Sprouting: New Dalit Writing From South India, Dossier 2: Telugu and Kannada http://harpercollins.co.in/ BookDetail.asp?Book Code=3732 4. Vimala. "Vantillu (The Kitchen)". Women Writing in India: 600 BC to the Present. Volume II: The 20th Century. Ed. Susie Tharu and K. Lalitha. Delhi: Oxford University Press, 1995. 599- 601.
4. Shatrughna, Veena et al. Women's Work and its Impact on Child Helath and Nutrition, Hyderabad, National Institute of Nutrition, Indian Council of Medical Research. 1993.
5. Stree Shakti Sanghatana. "We Were Making Histroy…", Life Stories of Women in the Telangana People's Struggle. New Delhi: Kali for Women, 1989.

6. Menon, Nivedita. Seeing like a Feminist. New Delhi: Zubaan-Penguin Books, 2012
7. Jayaprabha, A. "Chupulu (stares)". Women Writing in India: 600 BC to the Present. Volume II: The 20th Century Ed. Susie Tharu and K. Lalitha. Delhi: Oxford University Press, 1995. 596-597.
8. Javeed, Shayan and Anupam Manuhaar. "Women and Wage Discrimination in India: A Critical Analysis." International Journal of Humanities and Social Science Invention 2.4 (2013)
9. Gautam, Liela and Gita Ramaswamy. "A'conversation' between a Daughter and a Mother." Broadsheet on Contemporary Politics. Special Issue on Sexuality and Harassment: Gender Politics on Campus Today. Ed.Madhumeeta Sinha and Asma Rasheed. Hyderabad: Anveshi Research Center for Women's Studies, 2014.
10. Abdulali Sohaila. "I Fought For My Life… and Won." Available online at: http://www.thealternative.in/lifestyle/i-fought-formy-lifeand-wonsohaila-abhulal/
11. Jeganathan Pradeep, Partha Chatterjee (Ed). "Community, Gender and Violence Subaltern Studies XI". Permanent Black and Ravi Dayal Publishers, New Delhi, 2000.
12. K. Kapadia. The Violence of Development: The Politics of Identity, Gender and Social Inequalities in India. London: Zed Books, 2002.
13. S. Benhabib. Situating the Self: Gender, Community, and Postmodernism in Contemporary Ethics, London: Rout ledge, 1992.
14. Virginia Woolf. A Room of One's Own. Oxford: Black Swan. 1992.
15. T. Banuri and M. Mahmood, Just Development: Beyond Adjustment with a Human Face, Karachi: Oxford University Press, 1997.