Rai University - Rai School of Engineering

Department of Computer Science & Engineering / Information Technology

SYLLABUS FOR 3 Semester B.Tech. PROGRAMME

ACADEMIC WRITING (030003304)

Type of Course: B.Tech.

Prerequisite: Proficiency in the language of instruction (usually English) and a basic understanding of academic writing conventions.

Rationale: 1.Clarity and Precision : Develop the ability to express complex idea clearly and concisely . 2. Research Proficiency : Acquire skills in conducting and citing academic research effectively. 3. Critical Thinking : Enhance critical analysis and argumentation skills in writing. 4.Citation and Referencing : Master proper citation and referencing formats , such as APA or MLA. 5. Academic Integrity : Promote ethical writing practices and avoid plagiarism in academic work

Teaching and Examination Scheme:

Teaching Scheme			Credit	Examination Scheme					Total
				External		Internal			
Lecture Hrs/ Week	Tutorial Hrs/ Week	Lab Hrs/ Week		T	P	T	CE	P	
-	-	2	1	-	-	-	-	-	50

SEE - Semester End Examination, **T** - Internal Theory, **P** - Internal Practical

Contents:

Sr.	Topic	Weightage	Teaching Hrs.
1	**Academic & research writing**: Introduction; Importance of academic writing; Basic rules of academic writing, English in academic writing I & II; Elements of Writing Styles of research writing, Types of Academic Writing, Process of Academic Writing.	25%	11
2	**Plagiarism**: Introduction: Tools for the detection of plagiarism; Avoiding plagiarism,Literature review: Introduction, Source of literature; Process of literature review, Online literature databases; Literature management tools, referencing and citations	25%	12
3	**Report**: Report writing for an event, CV writing, Job Application, Types of letters- Business letters, Cover letter.	25%	11
4	**E-Mails**: Memo, Notice, Agenda, Minutes of Meeting, Business correspondence, How to write emails- do's and don'ts	25%	11

***Continuous Evaluation:**

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.

Reference Books:

1. Academic Writing, Anti- Plagiarism And Citations (TextBook)
By Vinod Kumar Kanvaria I Shipra Publications

Rai University - Rai School of Engineering

Department of Computer Science & Engineering / Information Technology

SYLLABUS FOR 3 Semester B.Tech. PROGRAMME

APTITUDE AND ANALYTICAL SKILLS (030003303)

Type of Course: B.Tech.

Prerequisite: Basic arithmetic, algebra, English grammar, logical reasoning, and data interpretation skills.

Rationale: ● Understand foundational concepts of quantitative aptitude and logical reasoning. ● Build problem-solving skills through real-world analytical reasoning exercises. ● Prepare students for competitive exams such as GATE, CAT, Campus Placements, and other aptitude tests. ● Apply mathematical reasoning and verbal skills to crack high-level aptitude questions. ● Strengthen data interpretation, visual reasoning, and logic formulation for technical interviews.

Teaching and Examination Scheme:

Teaching Scheme			Credit	Examination Scheme					Total
				External		Internal			
Lecture Hrs/ Week	Tutorial Hrs/ Week	Lab Hrs/ Week		T	P	T	CE	P	
3	-	-	3	70	-	-	-	-	100

SEE - Semester End Examination, **T** - Internal Theory, **P** - Internal Practical

Contents:

Sr.	Topic	Weightage	Teaching Hrs.
1	**Quantitative Aptitude – Arithmetic**: Topics: ● Number Systems ■ Integers, rational numbers, HCF & LCM ■ Remainder Theorem, unit digit, digital sum ● Percentages, Profit & Loss ■ Basic percentage concepts and successive percentage changes ■ Profit/loss, discount and mark-up based calculations ● Ratio, Proportion, and Mixtures ■ Simplified ratio problems ■ Mixture and alligation ● Averages, Time & Work ■ Weighted average, combined work problems ■ Pipes and cisterns ● Time, Speed, and Distance ■ Relative speed, trains, boats & streams, races	30%	12
2	**Simplified Algebra & Geometry for Placements**: Topics: ● Linear & Quadratic Equations ■ Solving linear and quadratic equations ■ Application in word problems ● Progressions ■ Arithmetic and Geometric Progressions ■ Number patterns and sequence-based problems ● Geometry and Mensuration ■ Area and perimeter of basic 2D shapes ■ Volume and surface area of 3D shapes ■ Basic coordinate geometry concepts	15%	10

3	**Logical and Analytical Reasoning**: Topics: ● Data Arrangements and Puzzles ■ Linear, circular, matrix-based puzzles ■ Blood relations, directions, rankings ● Syllogisms and Logical Deductions ■ Statements, conclusions using Venn diagrams ● Statements and Assumptions/Conclusions ■ Cause-effect, course of action problems ● Coding-Decoding & Series Completion ■ Letter and number-based series ■ Symbol/character substitution patterns ● Data Sufficiency ■ Logical decision-making using given statements	30%	15
4	**Data Interpretation & Visualization**: Topics: ● Tabular and Bar Graphs ■ Solving multi-data DI problems ■ Ratio, average, percentage-based DI ● Line Graphs & Pie Charts ■ Trend analysis and comparison ● Caselets and Mixed DI Sets ■ Logical DI involving multiple formats ■ Time-based data comparisons	20%	10
5	**Verbal Ability and Critical Thinking**: Topics: ● Reading Comprehension ■ Fact vs inference, tone of passage, main idea ● Vocabulary and Word Usage ■ Synonyms, antonyms, idioms, phrasal verbs, cloze tests ● Grammar and Sentence Correction ■ Subject-verb agreement, modifiers, prepositions ● Para Jumbles and Sentence Completion ■ Sentence sequencing, fill-in-the-blanks with logic	15%	10

***Continuous Evaluation:**

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.

Reference Books:

1. Quantitative Aptitude for Competitive Examinations
   By R.S. Aggarwal
2. A Modern Approach to Logical Reasoning
   By R.S. Aggarwal
3. How to Prepare for Quantitative Aptitude for CAT
   By by Arun Sharma
4. Verbal & Non-Verbal Reasoning
   By Dr. R.S. Aggarwal

Rai University - Rai School of Engineering

Department of Computer Science & Engineering / Information Technology

SYLLABUS FOR 3 Semester B.Tech. PROGRAMME

DATABASE MANAGEMENT SYSTEM (030703305)

Type of Course: B.Tech.

Prerequisite: 02080301-P - PROGRAMMING IN C++(P)

Rationale: 1.Basic understanding of Mathematics and Logic for data organization. 2.Familiarity with Programming Concepts, preferably in C, C++, or Python. 3.Fundamental knowledge of Data Structures like arrays, linked lists, and trees

Teaching and Examination Scheme:

Teaching Scheme			Credit	Examination Scheme					Total
				External		Internal			
Lecture Hrs/ Week	Tutorial Hrs/ Week	Lab Hrs/ Week		T	P	T	CE	P	
3	-	2	4	70	-	-	-	-	150

SEE - Semester End Examination, **T** - Internal Theory, **P** - Internal Practical

Contents:

Sr.	Topic	Weightage	Teaching Hrs.
1	**Introduction, Data Models**: What is database system, purpose of database system, view of data, Types of Databases, database architecture, transaction management Hierarchical data model, Network data model, Relational Data model	15%	8
2	**Relational Database Design and E-R Model, E-R Model, Normalization**: Structure of Relational databases, Domains, Relations, Relational algebra – fundamental operators and syntax, relational algebra queries, tuple relational calculus Basic concepts, Design process, constraints, Keys, Design issues, E-R diagrams, weak entity sets, extended E-R features – generalization, specialization, aggregation, reduction to E-R database schema, Data redundancy Normal forms 1NF, 2NF, 3NF, BCNF and 4NF	15%	12
3	**Structured Query Language, Constraints, Functions, Advanced Query**: Introduction to SQL, DDL, DML, DCL, TCL. Basic commands and Functions of SQL, Data Definition Language (DDL), Data Manipulation language (DML),Data Control Language (DCL), Transaction control Language (TCL) and all related commands, Use of Group by, Having, order by Primary key, foreign key, unique, not null, check, IN operator Aggregate functions, Built-in functions –numeric, date, string functions Set operations, Sub-queries and correlated sub-queries, Join and types of Join	30%	20
4	**Introduction to PL/SQL, Basics of PL/SQL**: The PL/SQL Syntax, The PL/SQL Block Structure, Fundamentals of PL/SQL, Advantages of PL/SQL data Types. Advanced SQL features such as updatable views, stored procedures, Triggers	30%	10
5	**Transaction Management and Concurrency Control**: Transaction concepts, ACID properties, Serializability and Concurrency Control, Lock based concurrency control (2PL, Deadlocks), Time stamping methods, optimistic methods, database recovery management.	10%	10

***Continuous Evaluation:**

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.

Reference Books:

1. Fundamentals of Database Systems
   By Ramez Elmasri and Shamkant B. Navathe l Fifth Edition, Pearson Education, 2008
2. An introduction to Database Systems (TextBook)
   By Desai Bipin C. l Pearson Education Asia l 7, Pub. Year 2001

List of Practical:

1. Perform the following : (a) View all databases, create a database of university, select that database and view all table in it. (b) Perform DDL commands (create, Alter, Truncate, Drop).
2. Perform DML (insert, update, delete) and DQL commands on student_info table.
3. Retrieve details from student_info table using distinct, order by clause and LIMIT clause.
4. Create customers table using Constraints with given Attributes: Customer_id – Primary key, Auto increment, Customer_name – Not Null, Contact_no – Unique key, City – Not Null.
5. Retrieve details from customers table using group by clause.
6. Create Product table with given attributes and Perform Aggregate functions (count, sum, avg, min, max) on product table. Product_id – Primary key, Product_name - Not Null, Quantity – Not Null, Price – Not Null.
7. Perform Numeric functions (sqrt, abs, floor, ceiling, round, square, power) on product table.
8. Perform String functions (ASCII, Char, Concat, Concat_ws, Left, Right, Lower, Upper, Ltrim, Rtrim, Trim, Reverse, substring, replace) on student_info table.
9. Perform Date functions (NOW, CURDATE, CURTIME, DATE, EXTRACT) on student_info table.
10. Apply check and default constraints on customers table.
11. Retrieve details from customers table using IN operator.
12. Perform join (inner, left, right, full outer) on tables.
13. Write a Subquery to transfer all the records from one table to another

Rai University - Rai School of Engineering

Department of Computer Science & Engineering / Information Technology

SYLLABUS FOR 3 Semester B.Tech. PROGRAMME

DATA AND FILE STRUCTURE USING C/C++ (030703304)

Type of Course: B.Tech.

Prerequisite: 030801101 - PROGRAMMING IN C

Rationale: 1. To organizing data for implementation of efficient algorithms and program development. 2. To develop the capability of selecting a particular data structure. 3.To develop application using array, structure, function, pointer and file

Teaching and Examination Scheme:

Teaching Scheme			Credit	Examination Scheme					Total
				External		Internal			
Lecture Hrs/ Week	Tutorial Hrs/ Week	Lab Hrs/ Week		T	P	T	CE	P	
3	-	2	4	70	-	-	-	-	150

SEE - Semester End Examination, **T** - Internal Theory, **P** - Internal Practical

Contents:

Sr.	Topic	Weightage	Teaching Hrs.
1	**Introduction to Data Structure, Types of Data Structures**: Introduction to Data Structure and Algorithm Data Types, primitive and non-primitive Performance Analysis and Measurement Linear & Non Linear Data Structures. String, Introduction, Operation performed on string Array, Introduction to Arrays, Linear array and its representation Binary Search and Linear Search with algorithm Pointers, Records and its representations., Recursion, Tower of Hanoi.	15%	10
2	**Link List, Stack, Queue (Linear data Structure)**: Introduction of stack, PUSH and POP operation along with algorithms, Application of Stack Expression Conversion: Prefix, Postfix ,Infix Link List Introduction, Types of Link List, Operations along with Algorithms- Insert, Delete, Search, Traverse, Application Queue types and Representation of Queue -Simple, Circular, Dequeue, Priority Operations Performed on Queue-Insert, delete, search, Application of Queue	20%	15
3	**Tree and Graphs-(Non Linear)**: Introduction of different trees and its representations, Types of Tree, Traversal algorithm, Binary search tree, Insert, Delete, Search in tree, Conversion of General Trees To Binary Trees, Applications Of Trees, Heap tree Graphs, Basic Terminology and Representation, Traversal of Graph, Sequential representation of graphs, Link list of graph, Graph-Matrix Representation of Graphs, Operations on graphs	20%	15
4	**Hashing and File Structure, File Organization**: Hashing, Hash Table, Hash Function, Hashing Technique, Collision, Collision Resolution Techniques File Organization, Introduction to File, Types of File Organization, Concepts of fields, records and files, Sequential, Indexed and Relative/Random File Organization, Indexing structure for index files, hashing for direct files, Multi-Key file organization and access methods.	20%	10
5	**Sorting and Searching**: Sorting types, Insertion, sort, Selection Sort, Quick Sort, Merge Sort, Radix sort, Searching types, Sequential Search and Binary Search	25%	10

***Continuous Evaluation:**

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.

Reference Books:

1. An Introduction to Data Structures with Applications (TextBook)
   By Jean-Paul Tremblay & Paul G. Sorenson I Tata McGraw Hill.
2. Data Structures using C & C++
   By Ten Baum I Prenctice-Hall International
3. Fundamentals of Computer Algorithms by
   By Horowitz, Sahni I Galgotia Pub. 2001 ed.

List of Practical:

1. Write a C program to display linear array elements.
2. Write a C program to calculate length of a given string.
3. Write a C program to perform index operation on a given String.
4. Write a C program to Concate two String.
5. Write a C program to find Sub string of given string.
6. Write a C program to implement PUSH and POP operation of STACK.
7. Write program to implement simple queue using C language.
8. Write a C program to search an element using linear search.
9. Write a C program to search an element using Binary search.
10. Write a C program to sort given list using Insertion sort.
11. Write a C program of matrix addition.
12. Write a C program of matrix multiplication.
13. Write a C program to traverse single linked list.
14. Write a C program to implement Bubble sort.
15. Write a C program to implement Radix sort.
16. Write a C program to implement Merge sort.
17. Write a C program to implement Selection sort.

Rai University - Rai School of Engineering

Department of Computer Science & Engineering / Information Technology

SYLLABUS FOR 3 Semester B.Tech. PROGRAMME

FULL STACK DEVELOPMENT (NEXT.JS) (030703306)

Type of Course: B.Tech.

Prerequisite: Basic knowledge of HTML, CSS, JavaScript, and React.js.

Rationale: Understand the core concepts of web rendering and the foundational features of the Next.js framework. Develop full-stack web applications using Next.js, API routes, authentication, and MongoDB integration. Design responsive UIs, optimize performance, and deploy applications using modern tools and best practices.

Teaching and Examination Scheme:

Teaching Scheme			Credit	Examination Scheme					Total
				External		Internal			
Lecture Hrs/ Week	Tutorial Hrs/ Week	Lab Hrs/ Week		T	P	T	CE	P	
3	-	2	4	70	-	-	-	-	150

SEE - Semester End Examination, **T** - Internal Theory, **P** - Internal Practical

Contents:

Sr.	Topic	Weightage	Teaching Hrs.
1	**Core Web Rendering Concepts & Next.js Fundamentals**: ● Fundamentals of Web Rendering ○ CSR (Client Side Rendering), SSR (Server Side Rendering), SSG (Static Site Generation), ISR (Incremental Static Regeneration) ○ Problems with CRA and benefits of SSR/SSG ○ SEO, performance, and UX considerations ● Introduction to Next.js ○ What is Next.js and why use it? ○ Core features: file-based routing, hybrid rendering, built-in API routes ○ When to choose Next.js for a project ● Project Setup and Configuration ○ Creating a new Next.js app ○ Project structure overview ○ Config files: next.config.js, .env.local, jsconfig.json ○ TypeScript and Tailwind CSS (optional integration) ● Basic Routing and Pages ○ File-based routing: static, dynamic, and nested routes ○ Custom 404 and _error.js pages ○ Linking between pages using <Link> and useRouter ● Metadata and SEO ○ Using next/head for title, meta tags, OpenGraph ○ Robots.txt, sitemap, canonical URLs	20%	10

2	**Data Fetching, API Routes & MongoDB Integration**: ● Data Fetching in Next.js ○ getStaticProps, getServerSideProps, getStaticPaths ○ Differences and real-world scenarios ○ ISR (Incremental Static Regeneration) concepts ○ Client-side fetching with SWR and React Query ● Full-Stack API Routes in Next.js ○ Creating API endpoints in /pages/api ○ Working with HTTP methods: GET, POST, PUT, DELETE ○ Request/response handling, middlewares, and custom headers ○ Input validation and error handling ● MongoDB Integration ○ Setting up MongoDB and Mongoose in a Next.js app ○ Connecting to the database using Mongoose models ○ CRUD operations in API routes ○ Schema design patterns and relational data with populate	20%	10
3	**Authentication, Authorization & State Management**: ● Authentication in Next.js ○ NextAuth.js setup (Google, GitHub, Credentials provider) ○ JWT-based custom authentication strategy ○ Login/logout, session management, token storage ○ Protected routes using middleware or HOCs ● Authorization ○ Role-based access control (RBAC) ○ Restricting pages and APIs to specific roles ● Global and Local State Management ○ useState, useContext, useReducer ○ Zustand for lightweight global state ○ Using SWR/React Query for data synchronization ○ Redux Toolkit integration	20%	10
4	**UI Design with Material UI, Forms & Validation**: ● UI Design with Material UI (MUI) ○ Theming and ThemeProvider setup ○ Responsive layouts with Grid, Container, Box, Card ○ Custom components and reusable UI patterns ○ Light/Dark mode switch ● Building Forms in Next.js ○ Handling inputs using useState or React Hook Form ○ Form validation with Yup or Zod ○ Form submission via API routes ○ Error messages, accessibility, and UX improvements ● Multi-step Forms and Dynamic Inputs ○ Conditional form fields ○ Navigation and progress indication between steps	20%	10
5	**Advanced Patterns, Optimization & Deployment**: ● Advanced Next.js Features ○ App Router (optional if using Next.js 13+) ○ Layouts, shared components across routes ○ Edge Functions and Middleware ○ Server Components and streaming (Next.js 13+) ● Real-time and External Integrations ○ Using Socket.io or Pusher for real-time features ○ Integrating Stripe or Razorpay for payments ○ Sending emails with NodeMailer ● Performance Optimization ○ Using next/image for image optimization ○ Dynamic imports and lazy loading ○ Lighthouse/Web Vitals for auditing and improvements ● Deployment and CI/CD ○ Deploying to Vercel (or alternatives: Render, Railway) ○ Environment variables and configuration ○ GitHub Actions for automated deployments ○ Monitoring with Sentry, LogRocket, or PostHog	20%	10

***Continuous Evaluation:**

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.

Reference Books:

1. Fullstack React with Next.js (TextBook)
   By Ari Lerner
2. Next.js Quick Start Guide
   By Kirill Konshin

List of Practical:

1. Create a static page using getStaticProps – Display your name on a static page in Next.js.
2. Create a dynamic route using [slug].js – Render different content for /blog/react and /blog/nextjs.
3. Navigate between pages using next/link – Add links to move between Home and About pages.
4. Connect to MongoDB and print "connected" in console – Use mongoose.connect() with a test MongoDB URI.
5. Create and insert one static product into MongoDB – Product: { name: 'Laptop', price: 50000 }
6. Fetch and display products using getServerSideProps – Show all products from MongoDB on /products.
7. Build a login form with email and password fields – Log form values in the console on submit.
8. Store and retrieve a JWT token in localStorage – Save "abc123" on login, and read it on page load.
9. Setup basic GitHub OAuth login using NextAuth – Show Sign In/Sign Out buttons using GitHub provider.
10. Create a form with one input using React Hook Form – Log the submitted value to console.
11. Validate a required field using Yup – Show “Name is required” if the input is empty.
12. Create a form with email and password + validation – Email format and min-length check for password.
13. Create a dashboard layout with sidebar and header – Use flex or grid to divide layout sections.
14. Build a form to add user names to a list – Use useState() and show the list below the form.
15. Show or hide buttons based on user role (RBAC) – If role is admin, show Edit/Delete buttons.

Rai University - Rai School of Engineering

Department of Computer Science & Engineering / Information Technology

SYLLABUS FOR 3 Semester B.Tech. PROGRAMME

CORE JAVA (030703302)

Type of Course: B.Tech.

Prerequisite: 030802201 - OBJECT ORIENTED PROGRAMMING WITH C++

Rationale: To working of the Java Virtual Machine (JVM), JDK, JRE, and IDEs. Develop Java programs using control structures, arrays, and methods. Implement object-oriented concepts like inheritance and polymorphism. Handle exceptions using try-catch-finally blocks and create user-defined exceptions. Create GUI-based applications using applets, layout managers, and Java AWT control

Teaching and Examination Scheme:

Teaching Scheme			Credit	Examination Scheme					Total
				External		Internal			
Lecture Hrs/ Week	Tutorial Hrs/ Week	Lab Hrs/ Week		T	P	T	CE	P	
3	-	2	4	70	-	-	-	-	150

SEE - Semester End Examination, **T** - Internal Theory, **P** - Internal Practical

Contents:

Sr.	Topic	Weightage	Teaching Hrs.
1	**Introduction to Java Programming**: Basics of Java programming, JDK,JRE, and IDE, Features of Java ,compiling and Executing a simple java program, Programming style, variables, Data Types (Numeric, Boolean, Character, String) its Operations , Adding comments to a java, Constants, Keywords, Identifiers, numeric type conversion, Arrays in Java, Advantages of Java, Applications of Java,	20%	12
2	**Methods, Object , Classes, Conditions & Loops in Java**: Basics of objects and classes in Java, Methods and objects. Method overloading, Constructors, Sending arguments to constructors, Constructors overloading. 'this' keyword, Static variable. Working with constants, if and if…..else, Nesting if… else, Using logical AND and OR operators, switch statement, Using the conditional AND not operators, Using the NOT operator, Understanding precedence. While loop, for loop, do…. while loop.	20%	12
3	**Inheritance and Polymorphism**: Inheritance in Java, Super and sub class. Overriding, Polymorphism, Dynamic binding. Generic Programming, Casting objects, Instance of operator, Abstract class, Interface in Java, Package in Java, Accessing super class methods. Constructor calling during inheritance, Extending classes, Final method, Final super class.	20%	12
4	**Exception Handling**: Learning about exceptions, Understanding the limitations of traditional error, and handling. Trying code and catching exceptions. Throwing and catching multiple exceptions. 'finally' block, Understanding the advantages of exception handling. Checked and unchecked exception, Creating own exceptions (custom exception).	20%	12

5	**Applet**: Introduction of Applet, Lifecycle of an Applet, Comparing Applets and Application, Creating Applets. Parameters passing in applet, Line, Rectangles, Ovals, Arcs, Polygons, Polyline methods. Event handling in Java, Event types, Mouse and key events. GUI Basics, Panels, Frames. Layout managers: Flow Layout, Border Layout, Grid Layout.GUI components like buttons, Check boxes, Radio buttons, Labels, Text fields, Text areas, Combo boxes, Lists, Scroll bars, Sliders, Windows, Menus, Dialog box.	20%	12

***Continuous Evaluation:**

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.

Reference Books:

1. Programming with Java (TextBook)
   By E. Balagurusamy I Sixth Edition, Tata Mc Graw Hill
2. Java Programming (TextBook)
   By Hari Mohan Pandey I Pearson
3. Java™: A Beginner's Guide (TextBook)
   By Herbert Schildt I 7th Edition

List of Practical:

1. Write a Java program to display "Hello World" and demonstrate basic syntax.
2. Write a program to find whether the number is prime or not.
3. Write a program to find a greater number among given three numbers using a) ternary operator and b)nested if.
4. Write a program to print the Fibonacci series.
5. Write a method to enter two integers and compute the gcd of two integers.
6. Write a test program that prompts the user to enter ten numbers, invoke a method to reverse the numbers, display the numbers.
7. Write a program that declares a class named Person. It should have instance variables to record name, age and salary. Use new operator to create a Person object. Set and display its instance variables. Add a constructor to the Person class developed above.
8. The employee list for a company contains employee code, name, designation and basic pay. The employee is given HRA of 10% of the basic and DA of 45% of the basic pay. The total pay of the employee is calculated as Basic pay+HRA+ DA. Write a class to define the details of the employee. Write a constructor to assign the required initial values. Add a method to calculate HRA, DA and Total pay and print them out. Write another class with a main method. Create objects for three different employees an
9. Write a Java Program to Count Number of Words in Given String.
10. Write a Java Program to Count the Occurrences of Each Character in String
11. Write a program to illustrate the concept of class with Method overloading.
12. Write a program to illustrate the concept of Single inheritance
13. Write a program to illustrate the concept of Multi level inheritance
14. Write an exception class for a time of day that can accept only 24 hour representation of clock hours. Write a java program to input various formats of timings and throw suitable error messages.
15. Write a program that illustrates interface inheritance. Interface A is extended by A1 and A2. Interface A12 inherits from both P1 and P2. Each interface declares one constant and one method. Class B implements A12.Instantiate B and invoke each of its methods. Each method displays one of the constants.
16. Developing an interactive Student Feedback Form using Java Applet and AWT GUI components. The form should collect the following information from the student: Name (Text Field),Roll Number (Text Field),Gender (Radio Buttons: Male, Female, Other),Course (Combo Box: BCA, BSc IT, MCA, MSc IT),Semester (List),Feedback on Facilities (Check Boxes: Library, Lab, Wi-Fi, Canteen, Hostel),Rating (Slider from 1 to 10),Submit and Reset Buttons
17. Develop a program using one class and object to calculate the area of a rectangle.

18. Demonstrate inheritance by creating a base class Person and a derived class Student.