1 Write a simple Java Program to print factorial of a given number using recursion.

```
public class FactorialCalculator {
    public static void main(String[] args) {
        int number = 5; // The number for which we want to calculate the factorial
        int factorial = calculateFactorial(number);
        System.out.println("Factorial of " + number + " is: " + factorial);
    }
    public static int calculateFactorial(int number) {
        if (number == 0 || number == 1) {
            return 1;
        } else {
            return number * calculateFactorial(number - 1);
        }
    }
}
```

////////////////////////////////

2 Write a Java program to implement student information in a file and perform the operations on it

```
import java.io.*;
import java.util.Scanner;

public class StudentInformationSystem {
    private static final String FILE_NAME = "student_info.txt";

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;

        do {
            System.out.println("********** Student Information System **********");
            System.out.println("1. Add Student");
            System.out.println("2. Display All Students");
            System.out.println("3. Search Student by ID");
            System.out.println("4. Delete Student by ID");
            System.out.println("0. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine(); // Consume the newline character

            switch (choice) {
                case 1:
                    addStudent();
                    break;
                case 2:
                    displayAllStudents();
                    break;
                case 3:
                    searchStudentByID();
                    break;
                case 4:
                    deleteStudentByID();
                    break;
                case 0:
                    System.out.println("Exiting the program... Goodbye!");
```

```
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
                break;
        }

        System.out.println();
    } while (choice != 0);
}

private static void addStudent() {
    try (PrintWriter writer = new PrintWriter(new FileWriter(FILE_NAME, true))) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("***** Add Student *****");
        System.out.print("Enter student ID: ");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        System.out.print("Enter student name: ");
        String name = scanner.nextLine();

        System.out.print("Enter student age: ");
        int age = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        writer.println(id + "," + name + "," + age);
        System.out.println("Student added successfully!");
    } catch (IOException e) {
        System.out.println("An error occurred while adding the student.");
        e.printStackTrace();
    }
}

private static void displayAllStudents() {
    try (BufferedReader reader = new BufferedReader(new FileReader(FILE_NAME))) {
        System.out.println("***** Displaying All Students *****");
        String line;

        while ((line = reader.readLine()) != null) {
            String[] parts = line.split(",");
            int id = Integer.parseInt(parts[0]);
            String name = parts[1];
            int age = Integer.parseInt(parts[2]);

            System.out.println("ID: " + id);
            System.out.println("Name: " + name);
            System.out.println("Age: " + age);
            System.out.println("----------------------");
        }
    } catch (IOException e) {
        System.out.println("An error occurred while displaying students.");
        e.printStackTrace();
    }
}
```

```
private static void searchStudentByID() {
    try (BufferedReader reader = new BufferedReader(new FileReader(FILE_NAME))) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("***** Search Student by ID *****");
        System.out.print("Enter student ID: ");
        int searchID = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        String line;

        while ((line = reader.readLine()) != null) {
            String[] parts = line.split(",");
            int id = Integer.parseInt(parts[0]);

            if (id == searchID) {
                String name = parts[1];
                int age = Integer.parseInt(parts[2]);

                System.out.println("ID: " + id);
                System.out.println("Name: " + name);
                System.out.println("Age: " + age);
                return;
            }
        }

        System.out.println("Student with ID " + searchID + " not found.");
    } catch (IOException e) {
        System.out.println("An error occurred while searching for the student.");
        e.printStackTrace();
    }
}

private static void deleteStudentByID() {
    try (BufferedReader reader = new BufferedReader(new FileReader(FILE_NAME));
         PrintWriter writer = new PrintWriter(new FileWriter("temp.txt"))) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("***** Delete Student by ID *****");
        System.out.print("Enter student ID: ");
        int deleteID = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        String line;

        while ((line = reader.readLine()) != null) {
            String[] parts = line.split(",");
            int id = Integer.parseInt(parts[0]);

            if (id != deleteID) {
                writer.println(line);
            }
        }
```

```
            File file = new File(FILE_NAME);
            File tempFile = new File("temp.txt");
            file.delete();
            tempFile.renameTo(file);

            System.out.println("Student with ID " + deleteID + " deleted successfully!");
        } catch (IOException e) {
            System.out.println("An error occurred while deleting the student.");
            e.printStackTrace();
        }
    }
}
```

//

.Q) Write a simple Java program to generate 5 random numbers.

```
import java.lang.Math;
public class RandomNumberExample1
{
public static void main(String args[])
{
// Generating random numbers
System.out.println("1st Random Number: " + Math.random());
System.out.println("2nd Random Number: " + Math.random());
System.out.println("3rd Random Number: " + Math.random());
System.out.println("4th Random Number: " + Math.random());
}
}
```

//

Q. 2. Write a program to design Registration process form using Applet and AWT components

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class RegistrationForm extends Frame
{
 TextField t,email1,nm1,enrollment_no1,address1,password1;
 Checkbox hobby1,hobby2,hobby3,hobby4,hobby5,male,female,Other,ce,me,ec;
 CheckboxGroup gender1,branch1;
 Button signup;
 Choice clg1;
 Label nm,enrollment_no,address,email,password,hobby,gender,branch,clg;
 Button b1;
 public RegistrationForm()
 {

  setTitle("RegsrationForm");
  setBackground(Color.cyan);
  setSize(353,350);
  setVisible(true);
```

```
setLayout(new FlowLayout());

nm=new Label("Enter Name");
add(nm);
nm1=new TextField(20);
add(nm1);

enrollment_no=new Label("Enter Your Enrollment No");
add(enrollment_no);
enrollment_no1=new TextField(18);
add(enrollment_no1);

address=new Label("Enter Your Address");
add(address);
address1=new TextField(20);
add(address1);

hobby=new Label("Select Your Hobby");
add(hobby);
hobby1=new Checkbox("Coding");
hobby2=new Checkbox("Study");
hobby3=new Checkbox("Programming");
hobby4=new Checkbox("photography");
hobby5=new Checkbox("Cricket");
add(hobby1);
add(hobby2);
add(hobby3);
add(hobby4);
add(hobby5);

gender=new Label("Select Your Gender");
add(gender);
gender1=new CheckboxGroup();
male=new Checkbox("Male",gender1,false);
female=new Checkbox("Female",gender1,false);
Other=new Checkbox("Other",gender1,false);
add(male);
add(female);

branch=new Label("Select Branch");
add(branch);
branch1=new CheckboxGroup();
ce=new Checkbox("CE.",branch1,false);
me=new Checkbox("Mech",branch1,false);
ec=new Checkbox("EC",branch1,false);
add(ce);
add(me);
add(ec);

clg=new Label("Select Your Collage");
add(clg);
clg1=new Choice();
clg1.add("GTU");
clg1.add("MSU");
clg1.add("LD Engg");
```

```
  clg1.add("NIRMA");
  add(clg1);

  email=new Label("Enter Your email");
  add(email);
  email1=new TextField(20);
  add(email1);

  password=new Label("Enter Your Password");
  add(password);
  password1=new TextField(20);
  password1.setEchoChar('*');
  add(password1);

  signup=new Button("Sign Up Now");
  add(signup);

 }
 public static void main(String s[])
 {
   RegistrationForm r1=new RegistrationForm();
 }

}
// step 1: Save the code in a file named RegistrationForm.java.
// javac RegistrationForm.java
// java RegistrationForm

// The registration form window will open, displaying the various input fields and //buttons. You can interac
t with the form by entering the required information and //clicking the "Sign Up Now" button.

// Note: The provided code uses the AWT library, which is a bit outdated. In modern //Java GUI developm
ent, it is recommended to use Swing or JavaFX for more robust and //feature-rich graphical interfaces

////////////////////////////////////////////////////////////////////////////////////////////////////
Q)   /Write a simple Java Program to implement stack using Queue interface
import java.util.Queue;
import java.util.LinkedList;
class Main {
  public static void main(String[] args) {
    // Creating Queue using the LinkedList class
    Queue<Integer> numbers = new LinkedList<>();

    // enqueue
    // insert element at the rear of the queue
    numbers.offer(1);
    numbers.offer(2);
    numbers.offer(3);
    System.out.println("Queue: " + numbers);

    // dequeue
    // delete element from the front of the queue
    int removedNumber = numbers.poll();
    System.out.println("Removed Element: " + removedNumber);
```

```
    System.out.println("Queue after deletion: " + numbers);
    }
}
```

//

Q. Write a JAVA Servlet Program to implement and demonstrate get() and Post methods(Using HTTP Servlet Class).

index.html

```
<html><head><title>Demonstration of Get and Post Method</title></head>
<body bgcolor="pink">
<center>
<form action="Prog3" method="post">
<a href="Prog3"><b>Click here to call get method</b></a><br><br>
<p><b>Press submit button to call Post method</b></p><br><B>color:</B>
<select name="color" size="1">
<option value="red">red</option>
<option value="green">green</option>
<option value="blue">blue</option>
</select><br><br>
<input type=submit value="submit">
</form>
</body>
</html>
```

Prog3.java (Servlet File)

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Prog3 extends HttpServlet
{
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException,IOException
{
String color=request.getParameter("color");
response.setContentType("text/html");
try (PrintWriter out = response.getWriter()) {
out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head>");
out.println("<title>Servlet Prog3</title>");
out.println("</head>");
out.println("<body bgcolor="+color+">");
out.println("<b>Hello from Post method</b><br><br>");
out.println("You have selected" + " " + color + " " + "color");
out.println("</body></html>");
}
}
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException,IOException
{
response.setContentType("text/html");
try (PrintWriter out = response.getWriter())
{
```

```
out.println("<b>Hello from Get method</b>");
out.println("<h1>Welcome to AIT</h1>");
}
}
```

//

2. Write a simple Java Program to Implement stack using Stack class

```
import java.util.Stack;

class Main {
  public static void main(String[] args) {

    // create an object of Stack class
    Stack<String> animals= new Stack<>();

    // push elements to top of stack
    animals.push("Dog");
    animals.push("Horse");
    animals.push("Cat");
    System.out.println("Stack: " + animals);

    // pop element from top of stack
    animals.pop();
    System.out.println("Stack after pop: " + animals);
  }
}
```

//

Q. 2 Write JSP Program to validate username and password

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/login")
public class LoginServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        if (username.equals("admin") && password.equals("admin123")) {
            response.sendRedirect("success.jsp");
        } else {
            response.sendRedirect("failure.jsp");
        }
    }
}
```

//

Q. 1 Write a java Program to implement thread using runnable interface

```
public class MyRunnable implements Runnable {
```

```
    public void run() {
        // Code to be executed in the thread
        System.out.println("Thread is running.");
    }

    public static void main(String[] args) {
        // Create an instance of the class implementing Runnable
        MyRunnable myRunnable = new MyRunnable();

        // Create a thread and pass the instance of the Runnable
        Thread thread = new Thread(myRunnable);

        // Start the thread
        thread.start();
    }
}
////////////////////////////////////////////////////////////////////////
 Write JSP program to print current date & time
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <title>Current Date and Time</title>
</head>
<body>
    <h1>Current Date and Time</h1>

    <%-- JSP code to print the current date and time --%>
    <%
        // Import required Java classes
        java.util.Date currentDate = new java.util.Date();

        // Display the current date and time
        out.println("<p>Current Date and Time: " + currentDate.toString() + "</p>");
    %>
</body>
</html>
////////////////////////////////////////////////////////////////////////////////
```