**SLIP :- 1**

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

**public class Factorial {**

**public static int factorial(int number) {**

**if (number == 0 || number == 1) {**

**return 1;**

**} else {**

**return number \* factorial(number - 1);**

**}**

**}**

**public static void main(String[] args) {**

**int number = 5; // Specify the number for which you want to calculate the factorial**

**int result = factorial(number);**

**System.out.println("Factorial of " + number + " is: " + result);**

**}**

**}**

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

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.io.\*;**

**public class GFG {**

**// Function to write a student**

**// information in JFrame and**

**// storing it in a file**

**public static void StudentInfo()**

**{**

**// Creating a new frame using JFrame**

**JFrame f**

**= new JFrame(**

**"Student Details Form");**

**// Creating the labels**

**JLabel l1, l2, l3, l4, l5;**

**// Creating three text fields.**

**// One for student name, one for**

**// college mail ID  and one**

**// for  Mobile No**

**JTextField t1, t2, t3;**

**// Creating two JComboboxes**

**// one for Branch and one**

**// for Section**

**JComboBox j1, j2;**

**// Creating  two buttons**

**JButton b1, b2;**

**// Naming the labels and setting**

**// the bounds for the labels**

**l1 = new JLabel("Student Name:");**

**l1.setBounds(50, 50, 100, 30);**

**l2 = new JLabel("College Email ID:");**

**l2.setBounds(50, 120, 120, 30);**

**l3 = new JLabel("Branch:");**

**l3.setBounds(50, 190, 50, 30);**

**l4 = new JLabel("Section:");**

**l4.setBounds(420, 50, 70, 30);**

**l5 = new JLabel("Mobile No:");**

**l5.setBounds(420, 120, 70, 30);**

**// Creating the textfields and**

**// setting the bounds for textfields**

**t1 = new JTextField();**

**t1.setBounds(150, 50, 130, 30);**

**t2 = new JTextField();**

**t2.setBounds(160, 120, 130, 30);**

**t3 = new JTextField();**

**t3.setBounds(490, 120, 130, 30);**

**// Creating two string arrays one for**

**// braches and other for sections**

**String s1[]**

**= { "  ", "CSE", "ECE", "EEE",**

**"CIVIL", "MEC", "Others" };**

**String s2[]**

**= { "  ", "Section-A", "Section-B",**

**"Section-C", "Section-D",**

**"Section-E" };**

**// Creating two JComboBoxes one for**

**// selecting branch and other for**

**// selecting the section**

**// and setting the bounds**

**j1 = new JComboBox(s1);**

**j1.setBounds(120, 190, 100, 30);**

**j2 = new JComboBox(s2);**

**j2.setBounds(470, 50, 140, 30);**

**// Creating one button for Saving**

**// and other button to close**

**// and setting the bounds**

**b1 = new JButton("Save");**

**b1.setBounds(150, 300, 70, 30);**

**b2 = new JButton("close");**

**b2.setBounds(420, 300, 70, 30);**

**// Adding action listener**

**b1.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e)**

**{**

**// Getting the text from text fields**

**// and JComboboxes**

**// and copying it to a strings**

**String s1 = t1.getText();**

**String s2 = t2.getText();**

**String s3 = j1.getSelectedItem() + "";**

**String s4 = j2.getSelectedItem() + "";**

**String s5 = t3.getText();**

**if (e.getSource() == b1) {**

**try {**

**// Creating a file and**

**// writing the data**

**// into a Textfile.**

**FileWriter w**

**= new FileWriter(**

**"GFG.txt", true);**

**w.write(s1 + "\n");**

**w.write(s2 + "\n");**

**w.write(s3 + "\n");**

**w.write(s4 + "\n");**

**w.write(s5 + "\n");**

**w.close();**

**}**

**catch (Exception ae) {**

**System.out.println(ae);**

**}**

**}**

**// Shows a Pop up Message when**

**// save button is clicked**

**JOptionPane**

**.showMessageDialog(**

**f,**

**"Successfully Saved"**

**+ " The Details");**

**}**

**});**

**// Action listener to close the form**

**b2.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e)**

**{**

**f.dispose();**

**}**

**});**

**// Default method for closing the frame**

**f.addWindowListener(new WindowAdapter() {**

**public void windowClosing(WindowEvent e)**

**{**

**System.exit(0);**

**}**

**});**

**// Adding the created objects**

**// to the frame**

**f.add(l1);**

**f.add(t1);**

**f.add(l2);**

**f.add(t2);**

**f.add(l3);**

**f.add(j1);**

**f.add(l4);**

**f.add(j2);**

**f.add(l5);**

**f.add(t3);**

**f.add(b1);**

**f.add(b2);**

**f.setLayout(null);**

**f.setSize(700, 600);**

**f.setVisible(true);**

**}**

**// Driver code**

**public static void main(String args[])**

**{**

**StudentInfo();**

**}**

**}**

**SLIP :-2**

**Q1. Write a simple Java program to generate 5 random numbers.**

**import java.util.Random;**

**public class RandomNumberGenerator {**

**public static void main(String[] args) {**

**Random random = new Random();**

**System.out.println("Generating 5 random numbers:");**

**for (int i = 0; i < 5; i++) {**

**int randomNumber = random.nextInt();**

**System.out.println(randomNumber);**

**}**

**}**

**}**

**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();**

**}**

**}**

**//<applet code="RegistrationForm.class" height=355 width=350></applet>**

**SLIP : 3**

**Q. 1. Write a 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. 2 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>");**

**}**

**}**

**SLIP : 4**

**Q. 1 Write a Java Program to Implement stack using Stack class**

**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**

**<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>Login Validation</title>**

**</head>**

**<body>**

**<h1>Login Validation</h1>**

**<form method="post" action="login.jsp">**

**<label for="username">Username:</label>**

**<input type="text" id="username" name="username" required><br>**

**<label for="password">Password:</label>**

**<input type="password" id="password" name="password" required><br>**

**<input type="submit" value="Login">**

**</form>**

**<%-- JSP code to validate the username and password --%>**

**<%@ page import="java.util.\*" %>**

**<%@ page import="java.io.\*" %>**

**<%!**

**private final String VALID\_USERNAME = "admin";**

**private final String VALID\_PASSWORD = "password";**

**private boolean validateLogin(String username, String password) {**

**return username.equals(VALID\_USERNAME) && password.equals(VALID\_PASSWORD);**

**}**

**%>**

**<%-- JSP code to process the form submission --%>**

**<% if (request.getMethod().equalsIgnoreCase("POST")) {**

**String username = request.getParameter("username");**

**String password = request.getParameter("password");**

**if (validateLogin(username, password)) {**

**out.println("<p>Login successful!</p>");**

**} else {**

**out.println("<p>Invalid username or password. Please try again.</p>");**

**}**

**} %>**

**</body>**

**</html>**

**SLIP : 5**

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

**public class MyRunnable implements Runnable {**

**@Override**

**public void run() {**

**// Code to be executed in the thread**

**for (int i = 0; i < 5; i++) {**

**System.out.println("Thread: " + Thread.currentThread().getId() + ", Count: " + i);**

**}**

**}**

**public static void main(String[] args) {**

**// Create an instance of the Runnable implementation**

**MyRunnable myRunnable = new MyRunnable();**

**// Create multiple threads using the same Runnable instance**

**Thread thread1 = new Thread(myRunnable);**

**Thread thread2 = new Thread(myRunnable);**

**// Start the threads**

**thread1.start();**

**thread2.start();**

**}**

**}**

**Q. 2 Write JSP program to print current date & time**

**<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>Current Date & Time</title>**

**</head>**

**<body>**

**<h1>Current Date & Time</h1>**

**<%-- JSP code to get the current date and time --%>**

**<%@ page import="java.util.Date" %>**

**<% Date currentDate = new Date(); %>**

**<%-- JSP code to print the current date and time --%>**

**<p>Current Date & Time: <%= currentDate %></p>**

**</body>**

**</html>**