Practical 1

1. Write a program to display first n prime numbers.

public class Prime {

public static void main(String[] args) {

int n=20;

int min = 0;

for (int i = 2; i <= n; i++) {

int count = 0;

for (int j = 1; j <= n; j++) {

if(i%j == 0){

count++;

}

}

if (count==2) {

min++;

if(min<=10){

System.out.println(i+" is prime");

}

}

}

}

}

Output:

2 is prime

3 is prime

5 is prime

7 is prime

11 is prime

13 is prime

17 is prime

19 is prime

1. Write a program to implement basic calculator

import java.util.Scanner;

class Calculator{

public static void main(String args[]) {

System.out.println("enter Value=");

Scanner sc= new Scanner(System.in);

int v = sc.nextInt();

int a=10;

int b=20;

int sum;

switch (v) {

case 1 -> {

sum=a+b;

System.out.println("add = "+sum);

}

case 2 -> {

sum=a-b;

System.out.println("Sub = "+sum);

}

case 3 -> {

sum=a\*b;

System.out.println("Mul = "+sum);

}

case 4 -> {

sum=a/b;

System.out.println("Div = "+sum);

}

default -> System.out.println("Error");

}

sc.close();

}

}

Output:

enter Value=

1

add = 30

enter Value=

2

Sub = -10

enter Value=

3

Mul = 200

enter Value=

4

Div = 0

1. Write a program to list a number divisible by n between n1 and n2 range

public class Cal2 {

public static void main(String[] args) {

int n=2;

int n1=1;

int n2=10;

for (int i = n1; i < n2; i++) {

if(i%n == 0){

System.out.println(n+" is divided by "+i);

}

}

}

}

OUTPUT

2 is divided by 2

2 is divided by 4

2 is divided by 6

2 is divided by 8

Practical 2

1. Write a program to validate email address

import java.util.Scanner;

public class P2P1 {

public static void main(String[] args) {

Scanner x = new Scanner(System.in);

String mail;

System.out.println("Enter Mail: ");

mail = x.nextLine();

if (mail.length() <= 30){

if (mail.contains("@") && mail.contains(".com") && !mail.contains(" ")) {

System.out.println("Valid");

}

else{

System.out.println("Invalid");

}

}

else{

System.out.println("Invalid");

}

x.close();

}

}

OUTPUT

Enter Mail:

dhyeypithadia@gmail.com

Valid

1. Write a program to validate GTU enrollno

import java.util.Scanner;

public class Validate\_GtuEnroll {

public static void main(String[] args){

Scanner x = new Scanner(System.in);

int flag = 0;

String Enroll;

System.out.print("Enter Enrollment: ");

Enroll = x.next();

if (Enroll.length() == 12 && Enroll != null && !Enroll.contains(" ")){

for(char ch : Enroll.toCharArray()){

if(Character.isDigit(ch)){

flag = 0;

break;

}

else{

flag = 1;

}

}

if(flag==0){

System.out.println("Valid");

}

else{

System.out.println("Invalid");

}

}

else{

System.out.println("InValid");

}

x.close();

}

}

OUTPUT

Enter Enrollment: 230470107140

Valid

1. Write a program to check given Strings are anagram or not

import java.util.Arrays;

import java.util.Scanner;

public class anagram {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter String 1 = ");

        String str = sc.next();

        System.out.println("Enter String 2 = ");

        String str1 = sc.next();

        if (str.length() == str1.length()){

            str = str.toLowerCase();

            str1 = str1.toLowerCase();

            char s1[] = str.toCharArray();

            char s2[] = str1.toCharArray();

            Arrays.sort(s1);

            Arrays.sort(s2);

            if(Arrays.equals(s1,s2)){

                System.out.println("String is anagram");

            }else{

                System.out.println("String is not anagram");

            }

        }

        else {

            System.out.println("String is not anagram");

        }

        sc.close();

    }

}

OUTPUT

Enter String 1 =

ana

Enter String 2 =

ana

String is anagram

Practical 3

1. Write a program to implement push and pop operations

import java.util.Scanner;

public class P1\_part2 {

static int top = -1;

static int size = 5;

static int stack[] = new int[5];

public static void push(){

Scanner sc = new Scanner(System.in);

if(top == size-1){

System.out.println("Stack overflow");

}

else{

top++;

System.out.println("Enter Element");

stack[top] = sc.nextInt();

}

}

public static void pop(){

if(top == -1){

System.out.println("Stack underflow");

}

else{

top--;

System.out.println("Element Popped");

}

}

public static void display(){

if(top == -1){

System.out.println("Stack underflow");

}

else{

for(int i=top;i>=0;i--){

System.out.println(stack[i]);

}

}

}

public static void main(String[] args) {

int v;

Scanner ch = new Scanner(System.in);

do{

System.out.println("Enter Choice");

System.out.println("1.push 2.pop 3.display 4.exit");

v = ch.nextInt();

switch(v) {

case 1:

push();

break;

case 2:

pop();

break;

case 3:

display();

break;

case 4:

System.out.println("Exiting ...");

v=0;

break;

default:

System.out.println("Not valid choice");

}

}while(v != 0);

ch.close();

}

}

OUTPUT

Enter Choice

1.push 2.pop 3.display 4.exit

1

Enter Element

2

Enter Choice

1.push 2.pop 3.display 4.exit

1

Enter Element

3

Enter Choice

1.push 2.pop 3.display 4.exit

1

Enter Element

4

Enter Choice

1.push 2.pop 3.display 4.exit

2

Element Popped

Enter Choice

1.push 2.pop 3.display 4.exit

3

3

2

Enter Choice

1.push 2.pop 3.display 4.exit

4

Exiting ...

1. Write a program to do matrix multiplication (3\*3)

import java.util.Scanner;

public class P2 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int martrixA[][] = new int[3][3];

int martrixB[][] = new int[3][3];

int result[][] = new int[3][3];

System.out.println("Enter Matrix A:");

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

martrixA[i][j] = sc.nextInt();

}

}

System.out.println("Enter Matrix B:");

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

martrixB[i][j] = sc.nextInt();

}

}

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

result[i][j] = 0;

for(int k=0;k<3;k++){

result[i][j] += martrixA[i][k] \* martrixB[k][j];

}

}

}

System.out.println("result is");

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

System.out.print(result[i][j]+" ");

}

System.out.println(" ");

}

sc.close();

}

}

OUTPUT

Enter Matrix A:

1 2 3

4 5 6

7 8 9

result is

30 24 18

84 69 54

138 114 90

Enter Matrix B:

9 8 7

6 5 4

3 2 1

Practical 4

1. Write a program to calculate area of circle, triangle and square using the concept of method overloading.

public class P1{

public static void area(double r)

{

double area = 3.14 \* r \* r;

System.out.println("Area of circle is: "+area);

}

public static void area(double h,double b)

{

double area = 0.5 \* b \* h;

System.out.println("Area of triangle is: "+area);

}

public static void area(float s)

{

double area = s\*s;

System.out.println("Area or Square is: "+area);

}

public static void main(String[] args) {

area(0.5f);

}

}

OUTPUT

Area or Square is: 0.25

Area or Square is: 16.0

Area of triangle is: 9.0

1. Write a method with following method header.

public static int gcd(int num1, int num2) Write a program that prompts the user to enter two integers and compute the gcd of two integers.

public class P2 {

    static int gcd = 1;

    public static int GCD(int num1,int num2)

    {

        for(int i=1; i<=num1 && i<=num2; i++)

        {

            if(num1%i == 0 && num2%i == 0)

            {

                gcd = i;

            }

        }

        return gcd;

    }

    public static void main(String[] args) {

        int ans = GCD(50,10);

        System.out.println(ans);

    }

}

OUTPUT

10

1. Define a class Student with name of student, department, and 3 subject marks as data members. Define methods to enter detials of 5 students, print details of the entered students and sort details of the students on the base of their percentage.

import java.util.Scanner;

import java.util.Arrays;

class Student {

String name;

String department;

int[] marks = new int[3];

double percentage;

Student(String name, String department, int m1, int m2, int m3) {

this.name = name;

this.department = department;

this.marks[0] = m1;

this.marks[1] = m2;

this.marks[2] = m3;

this.percentage = (m1 + m2 + m3) / 3.0;

}

void displayDetails() {

System.out.println("Name: " + name);

System.out.println("Department: " + department);

System.out.println("Marks: " + marks[0] + ", " + marks[1] + ", " + marks[2]);

System.out.println("Percentage: " + percentage + "%");

System.out.println("---------------------------------");

}

}

public class P3 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Student[] students = new Student[5];

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

System.out.println("Enter details for Student " + (i + 1) + ":");

System.out.print("Name: ");

String name = scanner.nextLine();

System.out.print("Department: ");

String department = scanner.nextLine();

System.out.print("Enter marks for 3 subjects: ");

int m1 = scanner.nextInt();

int m2 = scanner.nextInt();

int m3 = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

students[i] = new Student(name, department, m1, m2, m3);

}

Arrays.sort(students, (s1, s2) -> Double.compare(s2.percentage, s1.percentage));

System.out.println("\nStudent Details (Sorted by Percentage in Descending Order):");

for (Student student : students) {

student.displayDetails();

}

scanner.close();

}

}

OUTPUT

Enter details for Student 1:

Name: dhyey

Department: ce

Enter marks for 3 subjects: 80

80

80

Enter details for Student 2:

Name: yug

Department: ce

Enter marks for 3 subjects: 80

80

80

Enter details for Student 3:

Name: vismay

Department: ce

Enter marks for 3 subjects: 80

80

80

Enter details for Student 4:

Name: madhav

Department: it

Enter marks for 3 subjects: 90

90

90

Enter details for Student 5:

Name: rudra

Department: it

Enter marks for 3 subjects: 80

80

80

Student Details (Sorted by Percentage in Descending Order):

Name: madhav

Department: it

Marks: 90, 90, 90

Percentage: 90.0%

---------------------------------

Name: dhyey

Department: ce

Marks: 80, 80, 80

Percentage: 80.0%

---------------------------------

Name: yug

Department: ce

Marks: 80, 80, 80

Percentage: 80.0%

---------------------------------

Name: vismay

Department: ce

Marks: 80, 80, 80

Percentage: 80.0%

---------------------------------

Name: rudra

Department: it

Marks: 80, 80, 80

Percentage: 80.0%

1. Create a class named 'Member' having the following members:

1 – Name

2 - Age

3 - Phone number

4 - Address

5 - Salary

It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. No, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same along with specialization and department respectively.

class Member {

    String name;

    int age;

    String phone;

    String address;

    double salary;

    public Member(String name, int age, String phone, String address, double salary) {

        this.name = name;

        this.age = age;

        this.phone = phone;

        this.address = address;

        this.salary = salary;

    }

    public void printSalary() {

        System.out.println("Salary: " + salary);

    }

}

class Employee extends Member {

    String specialization;

    public Employee(String name, int age, String phone, String address, double salary, String specialization) {

        super(name, age, phone, address, salary);

        this.specialization = specialization;

    }

    public void displayDetails() {

        System.out.println("Employee Details:");

        System.out.println("Name: " + name);

        System.out.println("Age: " + age);

        System.out.println("Phone: " + phone);

        System.out.println("Address: " + address);

        printSalary();

        System.out.println("Specialization: " + specialization);

        System.out.println(" ");

    }

}

class Manager extends Member {

    String department;

    public Manager(String name, int age, String phone, String address, double salary, String department) {

        super(name, age, phone, address, salary);

        this.department = department;

    }

    public void displayDetails() {

        System.out.println("Manager Details:");

        System.out.println("Name: " + name);

        System.out.println("Age: " + age);

        System.out.println("Phone: " + phone);

        System.out.println("Address: " + address);

        printSalary();

        System.out.println("Department: " + department);

        System.out.println(" ");

    }

}

public class P4 {

    public static void main(String[] args) {

        // Creating Employee Object

        Employee emp = new Employee("Rahul Sharma", 30, "9876543210", "Ahmedabad, India", 50000, "Software Development");

        // Creating Manager Object

        Manager mgr = new Manager("Priya Patel", 40, "9123456789", "Surat, India", 80000, "IT Department");

        // Displaying Employee and Manager details

        emp.displayDetails();

        mgr.displayDetails();

    }

}

OUTPUT

Employee Details:

Name: Rahul Sharma

Age: 30

Phone: 9876543210

Address: Ahmedabad, India

Salary: 50000.0

Specialization: Software Development

Manager Details:

Name: Priya Patel

Age: 40

Phone: 9123456789

Address: Surat, India

Salary: 80000.0

Department: IT Department

Practical 5

1. Describe abstract class called Shape which has three subclasses say Triangle, Rectangle and Circle. Define one method area() in the abstract class and override this area() in these three subclasses to calculate for specific object.

abstract class Shape {

abstract void area();

}

class Triangle extends Shape {

double base, height;

void area() {

base = 10; // Assigning values directly

height = 5;

double area = 0.5 \* base \* height;

System.out.println("Area of Triangle: " + area);

}

}

class Circle extends Shape {

double radius;

void area() {

radius = 7; // Assigning value directly

double area = 3.14 \* radius \* radius;

System.out.println("Area of Circle: " + area);

}

}

class Rectangle extends Shape {

double length, breadth;

void area() {

length = 8; // Assigning values directly

breadth = 4;

double area = length \* breadth;

System.out.println("Area of Rectangle: " + area);

}

}

public class pro1 {

public static void main(String[] args) {

Triangle t = new Triangle();

Circle c = new Circle();

Rectangle r = new Rectangle();

t.area();

c.area();

r.area();

}

}

OUTPUT

Area of Triangle: 25.0

Area of Circle: 153.86

Area of Rectangle: 32.0

1. Write a program that illustrates interface inheritance. Interface P is extended by P1 and P2. Interface P12 inherits from both P1 and P2.Each interface declares one constant and one method. class Q implements P12.Instantiate Q and invoke each of its methods. Each method displays one of the constants.

interface P {

int var1=10;

void m();

}

interface P1 extends P {

int var2=20;

void m1();

}

interface P2 extends P{

int var3=30;

void m2();

}

interface P12 extends P1,P2 {

int var4=10;

void m3();

}

class Q implements P12{

public void m(){

System.out.println(var1);

}

public void m1() {

System.out.println(var2);

}

public void m2(){

System.out.println(var3);

}

public void m3(){

System.out.println(var4);

}

}

public class Prog2 {

public static void main(String[] args) {

Q q = new Q();

q.m();

q.m1();

q.m2();

q.m3();

}

}

OUTPUT

10

20

30

10

1. Write a program to create interface and declare one mehtod interest(p,r,n) then create two class SimpleInterest and CompoundInterest and calculate simple and compound interest.

import java.util.Scanner;

interface A {

void findInterest(double p, double r, double n);

}

class SimpleInterest implements A {

public void findInterest(double p, double r, double n) {

double SI = (p \* r \* n) / 100;

System.out.println("Simple Interest: " + SI);

}

}

class CompoundInterest implements A {

public void findInterest(double p, double r, double n) {

double CI = p \* (Math.pow((1 + (r / 100)), n) - 1);

System.out.println("Compound Interest: " + CI);

}

}

public class Prog3 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Principal Amount (P): ");

double p = sc.nextDouble();

System.out.println("Enter Rate of Interest (%): ");

double r = sc.nextDouble();

System.out.println("Enter Number of Years (N): ");

double n = sc.nextDouble();

System.out.println("Choose: 1 for Simple Interest, 2 for Compound Interest");

int choice = sc.nextInt();

A interestCalculator;

switch (choice) {

case 1:

interestCalculator = new SimpleInterest();

break;

case 2:

interestCalculator = new CompoundInterest();

break;

default:

System.out.println("Invalid choice! Please enter 1 or 2.");

sc.close();

return;

}

interestCalculator.findInterest(p, r, n);

sc.close();

}

}

OUTPUT

Enter Principal Amount (P):

1000

Enter Rate of Interest (%):

2

Enter Number of Years (N):

4

Choose: 1 for Simple Interest, 2 for Compound Interest

1

Simple Interest: 80.0

Enter Principal Amount (P):

1000

Enter Rate of Interest (%):

2

Enter Number of Years (N):

4

Choose: 1 for Simple Interest, 2 for Compound Interest

2

Compound Interest: 82.43215999999998

Practical 6

1. Create ArrayList mover. In which selected element of one list moved to another. i.e.

Subject Selected Subject

Java

DAA

Maths-I

EME

Database

Let’s move Java and DAA to selected subject

Subject Selected Subject

Maths-I Java

EME DAA

Database

import java.util.ArrayList;

public class prog1 {

    public static void main(String[] args) {

        ArrayList<String> list = new ArrayList<String>();

        list.add("Java");

        list.add("DAA");

        list.add("Maths-1");

        list.add("EME");

        list.add("Database");

        System.out.println("Available: "+list);

        String[] moveSubject = {"Java", "DAA"};

        ArrayList<String> list2 = new ArrayList<String>();

        for (String s : moveSubject){

            if(list.contains(s)){

                list2.add(s);

                list.remove(s);

            }

        }

        System.out.println("Selected: "+list2);

    }

}

OUTPUT

Available: [Java, DAA, Maths-1, EME, Database]

Selected: [Java, DAA]

1. You are given a phone book that consists of people's names and their phone number. After that you will be given some person's name as query. For each query, print the phone number of that person.

import java.util.HashMap;

import java.util.Scanner;

public class prog2 {

public static void main(String[] args) {

HashMap<String, Integer> hashMap = new HashMap<>();

hashMap.put("dhyey",25);

hashMap.put("vismay",45);

hashMap.put("yug",35);

int a;

Scanner v =new Scanner(System.in);

do{

System.out.println("Enter 1 continue search or 0 for exiting");

a = v.nextInt();

switch (a) {

case 1:

Scanner sc =new Scanner(System.in);

System.out.println("Enter name for search to get number:");

String name = sc.nextLine();

if(hashMap.containsKey(name)){

System.out.println(hashMap.get(name));

}

else{

System.out.println("not present");

}

break;

default:

System.out.println("Exiting....");

break;

}

}

while(a!=0);

v.close();

}

}

OUTPUT

Enter 1 continue search or 0 for exiting

1

Enter name for search to get number:

dhyey

25

Enter 1 continue search or 0 for exiting

1

Enter name for search to get number:

vismay

45

Enter 1 continue search or 0 for exiting

0

Exiting....

Practical 7

1. Design program to print a string with TypeWriter effect.

public class typewriter {

public static void main(String[] args) {

String text = "Hello world";

System.out.println(text);

int delay = 200;

for (char c : text.toCharArray()){

System.out.print(c);

try {

Thread.sleep(delay);

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

}

OUTPUT

Hello world

Hello world

1. Design PrettyPrinter in nonpremtive manner using Thread.

public class Prettier extends Thread {

public static void main(String[] args) {

Prettier Thread = new Prettier();

Prettier Thread1 = new Prettier();

Prettier Thread2 = new Prettier();

Thread.start();

Thread1.start();

Thread2.start();

}

static Object lock = new Object();

@Override

public void run() {

synchronized(lock){

for(int i = 1 ; i<=2 ; i++){

System.out.println(Thread.currentThread().getName()+" Threat is running");

}

}

}

}

OUTPUT

Thread-1 Threat is running

Thread-1 Threat is running

Thread-2 Threat is running

Thread-2 Threat is running

Thread-0 Threat is running

Thread-0 Threat is running

Practical 8

1. Write a program to export book data in csv format.

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

class Book {

    private String title;

    private String author;

    private double price;

    public Book(String title, String author, double price) {

        this.title = title;

        this.author = author;

        this.price = price;

    }

    public String toCSV() {

        return String.format("%s,%s,%.2f", title, author, price);

    }

}

public class BookCSVExporter {

    public static void main(String[] args) {

        ArrayList<Book> books = new ArrayList<>();

        books.add(new Book("The Great Gatsby", "F. Scott Fitzgerald", 10.99));

        books.add(new Book("1984", "George Orwell", 8.99));

        books.add(new Book("To Kill a Mockingbird", "Harper Lee", 12.49));

        String filePath = "books.csv";

        exportToCSV(books, filePath);

    }

    public static void exportToCSV(List<Book> books, String filePath) {

        try (FileWriter writer = new FileWriter(filePath)) {

            writer.append("Title,Author,Price\n"); // CSV Header

            for (Book book : books) {

                writer.append(book.toCSV()).append("\n");

            }

            System.out.println("Book data exported successfully to: " + filePath);

        } catch (IOException e) {

            System.err.println("Error writing to CSV file: " + e.getMessage());

        }

    }

}

OUTPUT

Book data exported successfully to: books.csv

1. Implement file read, write and append operation

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Paths;

public class FIle\_rwa {

private static final String FILE\_NAME = "sample.txt";

public static void main(String[] args) {

// Writing to the file

writeToFile("Hello, this is the first line.\n");

// Appending to the file

appendToFile("This is an appended line.\n");

// Reading from the file

readFromFile();

// Deleting from the file

// deleteFile();

}

// Method to write to a file (overwrites existing content)

public static void writeToFile(String content) {

try (FileWriter writer = new FileWriter(FILE\_NAME)) {

writer.write(content);

System.out.println("File written successfully.");

} catch (IOException e) {

System.out.println("Error writing file: " + e.getMessage());

}

}

// Method to append data to the file

public static void appendToFile(String content) {

try (FileWriter writer = new FileWriter(FILE\_NAME, true)) { // 'true' enables appending

writer.write(content);

System.out.println("Data appended successfully.");

} catch (IOException e) {

System.err.println("Error appending to file: " + e.getMessage());

}

}

// Method to read from the file

public static void readFromFile() {

try (BufferedReader reader = new BufferedReader(new FileReader(FILE\_NAME))) {

String line;

System.out.println("\nReading file content:");

while ((line = reader.readLine()) != null) {

System.out.println(line);

}

} catch (IOException e) {

System.err.println("Error reading file: " + e.getMessage());

}

}

// Method to delete from the file

/\*

public static void deleteFile() {

try {

File file = new File("example.txt");

file.delete();

Files.delete(Paths.get("sample.txt"));

System.out.println("File deleted successfully.");

} catch (IOException e) {

e.printStackTrace();

}

}

\*/

}

OUTPUT

File written successfully.

Data appended successfully.

Reading file content:

Hello, this is the first line.

This is an appended line.

Practical 9

1. Implement a utility to get content from input url.

// importing package required

import java.net.URL;

class URLexample {

    public static void main(String[] args) {

        String url = "https://www.tutorialspoint.com/java/index.htm";

        // Calling method to get URL info

        getUrlInfo(url);

    }

    static void getUrlInfo(String url\_string) {

             try {

            URL url = new URL(url\_string);

            System.out.println("Hostname: " + url.getHost());

            System.out.println("Port Number: " + url.getPort());

            System.out.println("File name: " + url.getFile());

            System.out.println("Protocol: " + url.getProtocol());

        } catch (Exception e) {

            // Exception handling left empty intentionally

        }

    }

}

OUTPUT

Note: URLexample.java uses or overrides a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

Hostname: www.tutorialspoint.com

Port Number: -1

File name: /java/index.htm

Protocol: https

Practical 10

1. Write a javafx form for user signup which includes username(Textfield), password(passwordfield), gender(radiobutton),branch(combobox),address(textarea), hobbies(checkbox). Create two buttons display value of user data into a label. signup(button), clear(button)

import javafx.application.Application;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.\*;

import javafx.stage.Stage;

public class SignupForm extends Application {

@Override

public void start(Stage primaryStage) {

// Username

Label lblUser = new Label("Username:");

TextField txtUser = new TextField();

// Password

Label lblPass = new Label("Password:");

PasswordField txtPass = new PasswordField();

// Gender (Radio Buttons)

Label lblGender = new Label("Gender:");

ToggleGroup genderGroup = new ToggleGroup();

RadioButton rbMale = new RadioButton("Male");

RadioButton rbFemale = new RadioButton("Female");

rbMale.setToggleGroup(genderGroup);

rbFemale.setToggleGroup(genderGroup);

// Branch (ComboBox)

Label lblBranch = new Label("Branch:");

ComboBox<String> cmbBranch = new ComboBox<>();

cmbBranch.getItems().addAll("Computer Science", "Mechanical", "Electrical", "Civil", "Electronics");

// Address (TextArea)

Label lblAddress = new Label("Address:");

TextArea txtAddress = new TextArea();

// Hobbies (CheckBox)

Label lblHobbies = new Label("Hobbies:");

CheckBox chkReading = new CheckBox("Reading");

CheckBox chkSports = new CheckBox("Sports");

CheckBox chkMusic = new CheckBox("Music");

// Signup and Clear buttons

Button btnSignup = new Button("Signup");

Button btnClear = new Button("Clear");

// Label to display user data

Label lblResult = new Label();

// Signup button action

btnSignup.setOnAction(e -> {

String username = txtUser.getText();

String password = txtPass.getText();

String gender = rbMale.isSelected() ? "Male" : rbFemale.isSelected() ? "Female" : "Not Selected";

String branch = cmbBranch.getValue() != null ? cmbBranch.getValue() : "Not Selected";

String address = txtAddress.getText();

String hobbies = "";

if (chkReading.isSelected()) hobbies += "Reading ";

if (chkSports.isSelected()) hobbies += "Sports ";

if (chkMusic.isSelected()) hobbies += "Music ";

lblResult.setText("User: " + username + "\nGender: " + gender + "\nBranch: " + branch +

"\nAddress: " + address + "\nHobbies: " + hobbies);

});

// Clear button action

btnClear.setOnAction(e -> {

txtUser.clear();

txtPass.clear();

genderGroup.selectToggle(null);

cmbBranch.setValue(null);

txtAddress.clear();

chkReading.setSelected(false);

chkSports.setSelected(false);

chkMusic.setSelected(false);

lblResult.setText("");

});

// Layout (GridPane)

GridPane grid = new GridPane();

grid.setVgap(10);

grid.setHgap(10);

grid.setAlignment(Pos.CENTER);

grid.add(lblUser, 0, 0);

grid.add(txtUser, 1, 0);

grid.add(lblPass, 0, 1);

grid.add(txtPass, 1, 1);

grid.add(lblGender, 0, 2);

grid.add(rbMale, 1, 2);

grid.add(rbFemale, 2, 2);

grid.add(lblBranch, 0, 3);

grid.add(cmbBranch, 1, 3);

grid.add(lblAddress, 0, 4);

grid.add(txtAddress, 1, 4, 2, 1);

grid.add(lblHobbies, 0, 5);

grid.add(chkReading, 1, 5);

grid.add(chkSports, 2, 5);

grid.add(chkMusic, 3, 5);

grid.add(btnSignup, 1, 6);

grid.add(btnClear, 2, 6);

grid.add(lblResult, 0, 7, 3, 1);

// Scene

Scene scene = new Scene(grid, 400, 400);

primaryStage.setTitle("Signup Form");

primaryStage.setScene(scene);

primaryStage.show();

}

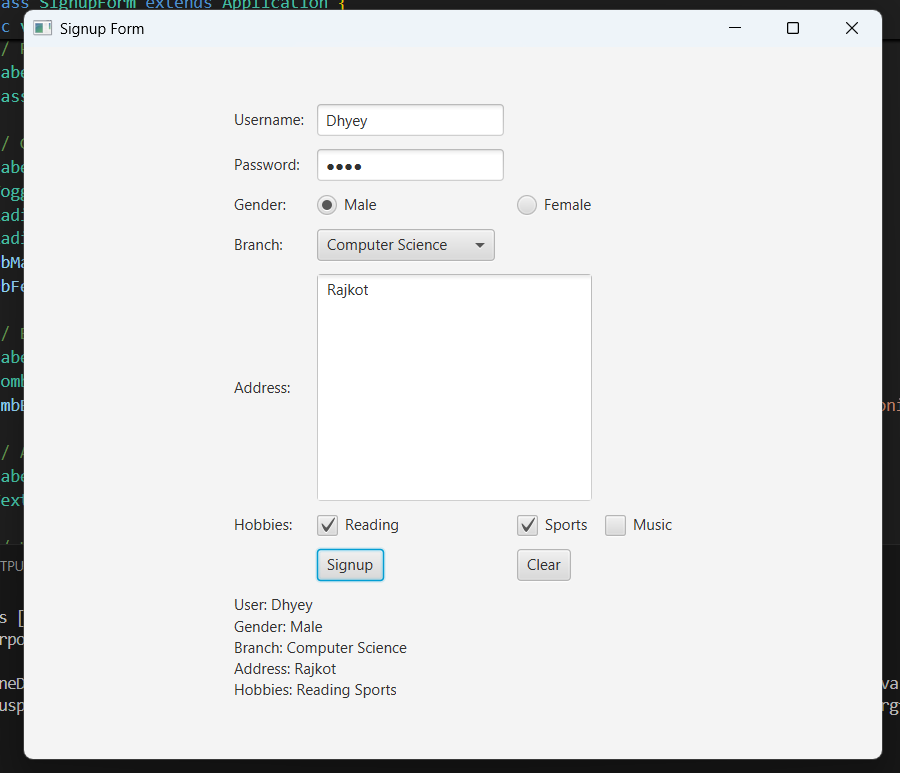
public static void main(String[] args) {

launch(args);

}

}

OUTPUT



1. Write a javafx program to move a circle from left to right when navigation keys been clicked. when mouse hovered over the circle then it change the color of circle.

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.input.KeyCode;

import javafx.scene.layout.Pane;

import javafx.scene.paint.Color;

import javafx.scene.shape.Circle;

import javafx.stage.Stage;

public class CircleMovement extends Application {

    @Override

    public void start(Stage primaryStage) {

        Pane pane = new Pane();

        // Create a circle

        Circle circle = new Circle(50, 100, 30);

        circle.setFill(Color.BLUE);

        // Add mouse hover events to change color

        circle.setOnMouseEntered(e -> circle.setFill(Color.RED));

        circle.setOnMouseExited(e -> circle.setFill(Color.BLUE));

        pane.getChildren().add(circle);

        // Create the scene and set key event handler

        Scene scene = new Scene(pane, 400, 200);

        scene.setOnKeyPressed(event -> {

            double step = 10;

            if (event.getCode() == KeyCode.RIGHT) {

                circle.setCenterX(circle.getCenterX() + step);

            } else if (event.getCode() == KeyCode.LEFT) {

                circle.setCenterX(circle.getCenterX() - step);

            }

        });

        primaryStage.setTitle("Move Circle with Keys and Change Color on Hover");

        primaryStage.setScene(scene);

        primaryStage.show();

        // To make sure the pane is focused and can receive key events

        pane.requestFocus();

    }

    public static void main(String[] args) {

        launch(args);

    }

}

OUTPUT

