**Java Practicals**

Slip 1

Q1. Write a program to accept a number from user and generate multiplication table of a number.

Solution:

import java.util.\*;

class Mult

{

public static void main(String args[])

{

int num;

Scanner sc = new Scanner(System.in);

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

num=sc.nextInt();

System.out.println("Multipication Table");

for(int i=1;i<=10;i++)

{

System.out.println(num+"\*"+i+"="+num\*i);

}

}

}

Q.2 Write a Java program to create a super class Vehicle having members Company and Price. Derive two different classes LightMotorVehicle (mileage) and HeavyMotorVehicle (capacity\_in\_tons). Accept the information for “n” vehicles and display the information in appropriate form. While taking data, ask user about the type of vehicle first.

Solution:

import java.util.Scanner;

abstract class Vehicle {

String company;

double price;

Vehicle(String company, double price) {

this.company = company;

this.price = price;

}

abstract void displayInfo();

}

class LightMotorVehicle extends Vehicle {

int mileage;

LightMotorVehicle(String company, double price, int mileage) {

super(company, price);

this.mileage = mileage;

}

@Override

void displayInfo() {

System.out.println("Company: " + company);

System.out.println("Price: " + price);

System.out.println("Mileage: " + mileage);

}

}

class HeavyMotorVehicle extends Vehicle {

int capacity\_in\_tons;

HeavyMotorVehicle(String company, double price, int capacity\_in\_tons) {

super(company, price);

this.capacity\_in\_tons = capacity\_in\_tons;

}

@Override

void displayInfo() {

System.out.println("Company: " + company);

System.out.println("Price: " + price);

System.out.println("Capacity: " + capacity\_in\_tons + " tons");

}

}

public class LightMotor {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = scanner.nextInt();

scanner.nextLine();

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

String type = scanner.nextLine();

String company = scanner.nextLine();

double price = scanner.nextDouble();

scanner.nextLine();

if (type.equalsIgnoreCase("LightMotorVehicle")) {

int mileage = scanner.nextInt();

Vehicle vehicle = new LightMotorVehicle(company, price, mileage);

vehicle.displayInfo();

} else if (type.equalsIgnoreCase("HeavyMotorVehicle")) {

int capacity\_in\_tons = scanner.nextInt();

Vehicle vehicle = new HeavyMotorVehicle(company, price, capacity\_in\_tons);

vehicle.displayInfo();

}

System.out.println();

}

}

}

OR

Q.2 Write a class Student with attributes roll no, name, marks and. Initialize values through parameterized constructor. If marks of student is not in between 15 and 20 then generate userdefined exception ―”Marks below average”. If name contains numbers or special symbols raise exception ―Name not valid.

Solution:

import java.util.\*;

import java.util.regex.Pattern;

class Student {

private int rollNo;

private String name;

private int marks;

public Student(int rollNo, String name, int marks) {

if (marks < 15 || marks > 20) {

throw new MarksBelowAverageException("Marks below average");

}

if (Pattern.matches("[0-9a-zA-Z\\s]+", name) == false) {

throw new NameNotValidException("Name not valid");

}

this.rollNo = rollNo;

this.name = name;

this.marks = marks;

}

public int getRollNo() {

return rollNo;

}

public String getName() {

return name;

}

public int getMarks() {

return marks;

}

}

class MarksBelowAverageException extends Exception {

public MarksBelowAverageException(String s) {

super(s);

}

}

class NameNotValidException extends Exception {

public NameNotValidException(String s) {

super(s);

}

}

public class StudentMarksException {

public static void main(String[] args) {

try {

Student student = new Student(1, "John Doe", 17);

System.out.println("Roll No: " + student.getRollNo());

System.out.println("Name: " + student.getName());

System.out.println("Marks: " + student.getMarks());

} catch (MarksBelowAverageException e) {

System.out.println(e.getMessage());

} catch (NameNotValidException e) {

System.out.println(e.getMessage());

}

}

}

Slip2

Q1. Write a program to accept ‘n’ integers from the user & store them in an Array List collection.

Display the elements of Array List.

Solution:

import java.util.\*;

public class DisplayArray

{

public static void main(String[] args)

{

ArrayList l1 = new ArrayList<>();

Scanner sc = new Scanner(System.in);

System.out.println("Enter no.of elements in list: ");

int n=sc.nextInt();

System.out.println("Enter elements in list: ");

for(int i=0;i<n;i++)

{

String elmt=sc.next();

l1.add(elmt);

}

System.out.println(l1);

}

}

Q2. Define a class MyNumber having one private integer data member. Write a default constructor

initialize it to 0 and another constructor to initialize it to a value. Write methods isNegative,

isPositive, isOdd, iseven. Use command line argument to pass a value to the object and perform the

above operations.

OR

Q.2 Write a program which define class Employee with data member as id, name and salary Store

the information of n; employees and Display the name of employee having maximum salary (Use

array of object).

Solution:

import java.util.\*;

class Employee

{

int id;

String name;

double salary;

public Employee(int id,String name,double salary)

{

this.id=id;

this.name=name;

this.salary=salary;

}

}

public class MainEmpSalary

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter number of employees: ");

int n=sc.nextInt();

Employee[] e = new Employee[n];

for(int i = 0;i<n;i++)

{

System.out.println("Enter employee ID: ");

int id=sc.nextInt();

System.out.println("Enter employee name: ");

String name=sc.next();

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

double salary=sc.nextDouble();

Employee emp=new Employee(id,name,salary);

e[i]=emp;

}

double maxSalary=e[0].salary;

Employee maxSalaryEmployee=e[0];

for(Employee emp: e)

{

if(emp.salary>maxSalary)

{

maxSalary=emp.salary;

maxSalaryEmployee=emp;

}

}

System.out.println("Employee with maximum salary is: "+maxSalaryEmployee.name);

}

}

Slip 3

Q.1 Write a Java Program to Reverse a Number. Accept number using command line argument.

Solution:

public class ReverseNumber

{

public static void main(String[] args)

{

if (args.length != 1)

{

System.out.println("Usage: java ReverseNumber <number>");

return;

}

int number = Integer.parseInt(args[0]);

int rev = 0;

while (number != 0)

{

int digit = number % 10;

rev = rev \* 10 + digit;

number = number / 10;

}

System.out.println("Reversed number: " + rev);

}

}

Q.2 Write a program which define class Employee with data member as id, name and salary Store the information of 'n' employees and display the name of employee having maximum salary (Use array of object).

Solution:

import java.util.\*;

class Employee

{

int id;

String name;

double salary;

public Employee(int id,String name,double salary)

{

this.id=id;

this.name=name;

this.salary=salary;

}

}

public class MainEmpSalary

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter number of employees: ");

int n=sc.nextInt();

Employee[] e = new Employee[n];

for(int i = 0;i<n;i++)

{

System.out.println("Enter employee ID: ");

int id=sc.nextInt();

System.out.println("Enter employee name: ");

String name=sc.next();

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

double salary=sc.nextDouble();

Employee emp=new Employee(id,name,salary);

e[i]=emp;

}

double maxSalary=e[0].salary;

Employee maxSalaryEmployee=e[0];

for(Employee emp: e)

{

if(emp.salary>maxSalary)

{

maxSalary=emp.salary;

maxSalaryEmployee=emp;

}

}

System.out.println("Employee with maximum salary is: "+maxSalaryEmployee.name);

}

}

OR

Q.2 Create a java application to store city names and their STD codes using an appropriate collection. i. Add a new city and its code (No duplicates) ii. Remove a city from the collection iii. Search for a cityname and display the code.

Solution:

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class CitySTD {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Map<String, String> citySTD = new HashMap<>();

citySTD.put("City1", "STD001");

citySTD.put("City2", "STD002");

citySTD.put("City3", "STD003");

while (true) {

System.out.println("\nMenu:");

System.out.println("1. Add a new city and its code");

System.out.println("2. Remove a city from the collection");

System.out.println("3. Search for a city and display the code");

System.out.println("4. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

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

switch (choice) {

case 1:

System.out.print("Enter city name: ");

String cityName = scanner.nextLine();

System.out.print("Enter STD code: ");

String stdCode = scanner.nextLine();

if (!citySTD.containsKey(cityName)) {

citySTD.put(cityName, stdCode);

System.out.println("City added successfully!");

} else {

System.out.println("City already exists!");

}

break;

case 2:

System.out.print("Enter city name to remove: ");

String cityToRemove = scanner.nextLine();

if (citySTD.containsKey(cityToRemove)) {

citySTD.remove(cityToRemove);

System.out.println("City removed successfully!");

} else {

System.out.println("City not found!");

}

break;

case 3:

System.out.print("Enter city name to search: ");

String cityToSearch = scanner.nextLine();

if (citySTD.containsKey(cityToSearch)) {

System.out.println("City: " + cityToSearch + ", STD code: " + citySTD.get(cityToSearch));

} else {

System.out.println("City not found!");

}

break;

case 4:

scanner.close();

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

System.exit(0);

default:

System.out.println("Invalid choice!");

break;

}

}

}

}

Slip 4

Q.1 Write java program to check whether number is Perfect or not.

Solution:

//import java.util.Scanner;

public class PerfectNumber

{

public static void main(String args[])

{

long n=7, sum=0;

//Scanner sc=new Scanner(System.in);

//System.out.print("Enter the number: ");

//n=sc.nextLong();

int i=1;

while(i <= n/2)

{

if(n % i == 0)

{

sum = sum + i;

}

i++;

}

if(sum==n)

{

System.out.println(n+" is a perfect number.");

}

else

System.out.println(n+" is not a perfect number.");

}

}

Q.2 Define a class student having rollno, name and percentage. Define Default and parameterized constructor. Accept the 5 student details and display it. (Use this keyword)

Solution:

import java.util.\*;

public class StudentDetails

{

int rollno;

String name;

float percentage;

public StudentDetails(int rollno,String name,float percentage)

{

this.rollno=rollno;

this.name=name;

this.percentage=percentage;

}

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter number of student: ");

int n=sc.nextInt();

StudentDetails s[] = new StudentDetails[n];

for(int i=0; i<n; i++)

{

System.out.println("Student Roll No.: ");

int id=sc.nextInt();

System.out.println("Student name: ");

String name=sc.next();

System.out.println("Student percentage: ");

float percentage=sc.nextFloat();

}

}

}

OR

Slip 5

Q.1 . Write a program to accept ‘n’ integers from the user & store them in an Array List collection. Display the elements of Array List.

**Same as slip1 Q1**

Q.2 . Write a program create class as MyDate with dd,mm,yy as data members. Write parameterized constructor. Display the date in dd-mm-yy format. (Use this keyword).

Solution:

public class MyDateSlip5 {

int dd;

int mm;

int yy;

public MyDateSlip5(int dd, int mm, int yy) {

this.dd = dd;

this.mm = mm;

this.yy = yy;

}

public void displayDate() {

System.out.println(this.dd + "-" + this.mm + "-" + this.yy);

}

public static void main(String[] args) {

MyDateSlip5 myDate = new MyDateSlip5(1, 1, 2023);

myDate.displayDate();

}

}

OR

Q.2 Write a program to design following GUI using JTextArea. Write a code to display number of words and characters of text in JLabel. Use JScrollPane to get scrollbars for JTextArea

Slip 6

Q.1Write a Java program to print the sum of elements of the array.

Solution:

public class SumOfArray {

public static void main(String[] args) {

int [] arr = new int [] {1, 2, 3, 4, 5};

int sum = 0;

for (int i = 0; i < arr.length; i++) {

sum = sum + arr[i];

}

System.out.println("Sum of all the elements of an array: " + sum);

}

}

Q.2 . Define a “Point” class having members – x,y(coordinates). Define default constructor and parameterized constructors. Define two subclasses “ColorPoint” with member as color and subclass “Point3D” with member as z (coordinate). Write display method to display the details of different types of Points .

OR

Q.2 Write a java program to accept Doctor Name from the user and check whether it is valid or not. (It should not contain digits and special symbol) If it is not valid then throw user defined Exception - Name is Invalid -- otherwise display it.

Solution:

import java.util.Scanner;

class InvalidNameException extends Exception {

public InvalidNameException(String message) {

super(message);

}

}

public class DoctorNameValidation {

static boolean isValidName(String name) {

return name.matches("[a-zA-Z]+");

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Doctor's Name: ");

try {

String doctorName = scanner.nextLine();

if (!isValidName(doctorName)) {

throw new InvalidNameException("Name is Invalid");

} else {

System.out.println("Doctor's Name: " + doctorName);

}

} catch (InvalidNameException e) {

System.out.println(e.getMessage());

} finally {

scanner.close();

}

}

}

Slip 7

Q.1Write a Java program to print the factors of a given number. (Use Scanner class).

Solution:

import java.util.Scanner;

public class Factors {

public static void main(String[] args) {

int n, i;

Scanner my\_scanner = new Scanner(System.in);

System.out.print("Enter the number : ");

n = my\_scanner.nextInt();

System.out.print("The factors of " + n + " are: ");

for (i = 1; i <= n; ++i) {

if (n % i == 0) {

System.out.print(i + " ");

}

}

}

}

Q.2 Write a Java program to create a super class Employee (members – name, salary). Derive a subclass as Developer (member – projectname). Derive a sub-class Programmer (member – proglanguage) from Developer. Create object of Programmer and display the details of it. Implement this multilevel inheritance with appropriate constructor and methods.

OR

Q. 2 Create an abstract class Myshapes with methods area & volume. Derive a class Cylinder (radius, height). Calculate area and volume.

Solution:

import java.util.Scanner;

abstract class Myshapes {

abstract double area();

abstract double volume();

}

class Cylinder extends Myshapes {

double radius;

double height;

Cylinder(double radius, double height) {

this.radius = radius;

this.height = height;

}

double area() {

return 2 \* Math.PI \* radius \* (radius + height);

}

double volume() {

return Math.PI \* radius \* radius \* height;

}

}

public class MainCylinder {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the radius of the cylinder: ");

double radius = scanner.nextDouble();

System.out.println("Enter the height of the cylinder: ");

double height = scanner.nextDouble();

Cylinder cylinder = new Cylinder(radius, height);

System.out.println("The area of the cylinder is: " + cylinder.area());

System.out.println("The volume of the cylinder is: " + cylinder.volume());

}

}

Slip 8

Q.1Write java program to check whether number is Palindrome or not.

Solution:

class PalindromeExample{

public static void main(String args[]){

int r,sum=0,temp;

int n=454;

temp=n;

while(n>0){

r=n%10; //getting remainder

sum=(sum\*10)+r;

n=n/10;

}

if(temp==sum)

System.out.println("palindrome number ");

else

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

}

}

Q.2 Create an abstract class Shape with methods area & volume. Derive a class Cylinder (radius, height). Calculate area and volume.

Solution:

import java.util.Scanner;

abstract class Myshapes {

abstract double area();

abstract double volume();

}

class Cylinder extends Myshapes {

double radius;

double height;

Cylinder(double radius, double height) {

this.radius = radius;

this.height = height;

}

double area() {

return 2 \* Math.PI \* radius \* (radius + height);

}

double volume() {

return Math.PI \* radius \* radius \* height;

}

}

public class MainCylinder {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the radius of the cylinder: ");

double radius = scanner.nextDouble();

System.out.println("Enter the height of the cylinder: ");

double height = scanner.nextDouble();

Cylinder cylinder = new Cylinder(radius, height);

System.out.println("The area of the cylinder is: " + cylinder.area());

System.out.println("The volume of the cylinder is: " + cylinder.volume());

}

}

OR

Q.2 Write a java program to accept Doctor Name from the user and check whether it is valid or not. (It should not contain digits and special symbol) If it is not valid then throw user defined Exception - Name is Invalid -- otherwise display it.

Solution:

import java.util.Scanner;

class InvalidNameException extends Exception {

public InvalidNameException(String message) {

super(message);

}

}

public class DoctorNameValidation {

static boolean isValidName(String name) {

return name.matches("[a-zA-Z]+");

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Doctor's Name: ");

try {

String doctorName = scanner.nextLine();

if (!isValidName(doctorName)) {

throw new InvalidNameException("Name is Invalid");

} else {

System.out.println("Doctor's Name: " + doctorName);

}

} catch (InvalidNameException e) {

System.out.println(e.getMessage());

} finally {

scanner.close();

}

}

}

Slip 9

Q.1Write Java program to find multiplication of two matrix. Accept matrix from user.

Solution:

import java.util.Scanner;

public class MatrixMultiplication {

public static void main(String[] args) {

int i, j, k, rowF, rowS, colF, colS;

int first[][] = new int[10][10];

int second[][] = new int[10][10];

int product[][] = new int[10][10];

Scanner scanner = new Scanner(System.in);

System.out.println("Enter Rows and Cols of First Matrix");

rowF = scanner.nextInt();

colF = scanner.nextInt();

System.out.println("Enter Elements of First Matrix");

for (i = 0; i < rowF; i++) {

for (j = 0; j < colF; j++) {

first[i][j] = scanner.nextInt();

}

}

System.out.println("Enter Rows and Cols of Second Matrix");

rowS = scanner.nextInt();

colS = scanner.nextInt();

System.out.println("Enter Elements of Second Matrix");

for (i = 0; i < rowS; i++) {

for (j = 0; j < colS; j++) {

second[i][j] = scanner.nextInt();

}

}

for (i = 0; i < rowF; i++) {

for (j = 0; j < colF; j++) {

for (k = 0; k < colS; k++) {

product[i][j] += first[i][k] \* second[k][j];

}

}

}

System.out.println("Product Matrix");

for (i = 0; i < rowF; i++) {

for (j = 0; j < colS; j++) {

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

}

System.out.print("\n");

}

}

}

Q.2 . Define an interface “Operation” which has methods area (), volume (). Define a constant PI having a value 3.142. Create a class circle (member – radius), cylinder (members – radius, height) which implements this interface. Calculate and display the area and volume

OR

Q.2 Write a program which define class Student with data member as rno, name and score. Store the information of 'n' students and Display the name of student having maximum score (Use array of object).

Solution:

import java.util.\*;

class Student

{

int id;

String name;

double marks;

public Student(int id,String name,double marks)

{

this.id=id;

this.name=name;

this.marks=marks;

}

}

public class StudMaxMarks

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter number of students: ");

int n=sc.nextInt();

Student[] s=new Student[n];

for(int i=0;i<n;i++)

{

System.out.println("Enter student ID: ");

int id=sc.nextInt();

System.out.println("Enter student name: ");

String name=sc.next();

System.out.println("Enter student marks: ");

double marks=sc.nextDouble();

Student stud=new Student(id,name,marks);

s[i]=stud;

}

double maxMarks=s[0].marks;

Student maxStudMarks=s[0];

for(Student stud:s)

{

if(stud.marks>maxMarks)

{

maxMarks=stud.marks;

maxStudMarks=stud;

}

}

System.out.println("Student with maximum marks is: "+maxStudMarks.name);

}

}

Slip 10

Q.1 Write a Java Program to Display Armstrong Numbers Between range. Accept range from user.

Solution:

import java.util.Scanner;

public class ArmstrongNumberWithinRange

{

public static void main(String args[])

{

Scanner s=new Scanner(System.in);

System.out.print("Enter the Range :");

int range=s.nextInt();

int a,b,c,sum,count=0;

b=1;

while(b<=range)

{

sum=0;

c=b;

while(c>0)

{

a=c%10;

sum=sum+(a\*a\*a);

c=c/10;

}

if(sum==b)

{

System.out.println(b+" is a Armstrong Number");

count=count+1;

}

b++;

}

System.out.println("Total Armstrong Number Present With in that Range is "+count);

}

}

Q.2 Accept 'n' integers from the user and store them in a collection. Display them in the sorted order. The collection should not accept duplicate elements. (Use a suitable collection). Search for a particular element using predefined search method in the Collection framework.

Solution:

import java.util.TreeSet;

import java.util.Scanner;

public class collection3 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

TreeSet<Object> ts = new TreeSet<>();

System.out.println("Enter how many Numbers: ");

int n = sc.nextInt();

System.out.println("Enter the " + n + " Numbers: ");

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

int num = sc.nextInt();

ts.add(num);

}

System.out.println("Numbers in Sorted Order and without Duplication :" + ts);

System.out.println("Enter element to be search: ");

int num = sc.nextInt();

if(ts.contains(num))

{

System.out.println("Element is found");

}

else

{

System.out.println("Element is not found");

}

}

}

OR

Q.2 Define a class MyData having one private integer data member. Write a default constructor initialize it to 0 and another constructor to initialize it to a value. Write methods isNegative, isPositive, isOdd, iseven. Use command line argument to pass a value to the object and perform the above operations.

Slip 11

Q.1 Write a java program to find factorial of a number.Accept number from user.

Solution:

import java.util.\*;

class FactorialExample

{

public static void main(String args[])

{

int n, i,fact=1;

Scanner sc=new Scanner(System.in);

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

n=sc.nextInt();

for(i=1;i<=n;i++)

{

fact=fact\*i;

}

System.out.println("Factorial of "+n+" is: "+fact);

}

}

Q.2 Write a java program to accept a number from the user, if number is zero then throw user defined exception ―Number is 0, otherwise check whether no is prime or not.

Solution:

import java.util.Scanner;

class CustomException extends Exception {

public CustomException(String message) {

super(message);

}

}

public class PrimeCheck {

static boolean isPrime(int num) {

if (num <= 1) {

return false;

}

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) {

return false;

}

}

return true;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

try {

int number = scanner.nextInt();

if (number == 0) {

throw new CustomException("Number is 0");

} else {

if (isPrime(number)) {

System.out.println(number + " is a prime number.");

} else {

System.out.println(number + " is not a prime number.");

}

}

} catch (CustomException e) {

System.out.println(e.getMessage());

} catch (Exception e) {

System.out.println("Invalid input. Please enter a valid number.");

} finally {

scanner.close();

}

}

}

OR

Q.2Write a java program to design a following GUI. Use appropriate Layout and Components.

Slip 12

Q.1Write a program to accept the 'n' different numbers from user and store it in array. Display maximum number from an array.

Solution:

public class MaxofArray{

public static void main(String[] args) {

int [] arr = new int [] {25, 11, 7, 75, 56};

int max = arr[0];

for (int i = 0; i < arr.length; i++) {

if(arr[i] > max)

max = arr[i];

}

System.out.println("Largest element present in given array: " + max);

}

}

Q.2 Write a java program to accept Doctor Name from the user and check whether it is valid or not. (It should not contain digits and special symbol) If it is not valid then throw user defined Exception - Name is Invalid -- otherwise display it.

Solution:

import java.util.Scanner;

class InvalidNameException extends Exception {

public InvalidNameException(String message) {

super(message);

}

}

public class DoctorNameValidation {

// Function to check if the name is valid

static boolean isValidName(String name) {

return name.matches("[a-zA-Z]+");

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Doctor's Name: ");

try {

String doctorName = scanner.nextLine();

if (!isValidName(doctorName)) {

throw new InvalidNameException("Name is Invalid");

} else {

System.out.println("Doctor's Name: " + doctorName);

}

} catch (InvalidNameException e) {

System.out.println(e.getMessage());

} finally {

scanner.close();

}

}

}

OR

Q.2 Write a program which define class Student with data member as id, name and marks Store the information of 'n' students and Display the name of student having maximum marks (Use array of object).

Solution:

import java.util.\*;

class Student

{

int id;

String name;

double marks;

public Student(int id,String name,double marks)

{

this.id=id;

this.name=name;

this.marks=marks;

}

}

public class StudMaxMarks

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter number of students: ");

int n=sc.nextInt();

Student[] s=new Student[n];

for(int i=0;i<n;i++)

{

System.out.println("Enter student ID: ");

int id=sc.nextInt();

System.out.println("Enter student name: ");

String name=sc.next();

System.out.println("Enter student marks: ");

double marks=sc.nextDouble();

Student stud=new Student(id,name,marks);

s[i]=stud;

}

double maxMarks=s[0].marks;

Student maxStudMarks=s[0];

for(Student stud:s)

{

if(stud.marks>maxMarks)

{

maxMarks=stud.marks;

maxStudMarks=stud;

}

}

System.out.println("Student with maximum marks is: "+maxStudMarks.name);

}

}

Slip 13

Q1. Define a class MyNumber having one private integer data member. Write a parameterized constructor to initialize to a value. Write isEven() to check given number is even or not. Use command line argument to pass a value to the object.

Solution:

public class MyNumber13 {

private int num;

public MyNumber13(int num) {

this.num = num;

}

public boolean isEven() {

return num % 2 == 0;

}

public static void main(String[] args) {

if (args.length == 0) {

System.out.println("No argument provided.");

return;

}

int value = Integer.parseInt(args[0]);

MyNumber13 myNumber = new MyNumber13(value);

if (myNumber.isEven()) {

System.out.println(value + " is even.");

} else {

System.out.println(value + " is not even.");

}

}

}

Q.2 Write a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user-defined exception ―Age Not Within The Range. If name contains numbers or special symbols raise exception ―Name not valid.

Solution:

class AgeNotWithinRangeException extends Exception {

public AgeNotWithinRangeException(String message) {

super(message);

}

}

class InvalidNameException extends Exception {

public InvalidNameException(String message) {

super(message);

}

}

class Student {

private int rollNo;

private String name;

private int age;

private String course;

public Student(int rollNo, String name, int age, String course) throws AgeNotWithinRangeException, InvalidNameException {

this.rollNo = rollNo;

if (age < 15 || age > 21) {

throw new AgeNotWithinRangeException("Age Not Within The Range");

} else {

this.age = age;

}

if (!isValidName(name)) {

throw new InvalidNameException("Name not valid");

} else {

this.name = name;

}

this.course = course;

}

private boolean isValidName(String name) {

return name.matches("[a-zA-Z]+");

}

public void displayStudentDetails() {

System.out.println("Roll Number: " + rollNo);

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

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

System.out.println("Course: " + course);

}

}

public class StudentAge {

public static void main(String[] args) {

try {

Student student = new Student(101, "JohnDoe", 18, "Computer Science");

student.displayStudentDetails();

} catch (AgeNotWithinRangeException e) {

System.out.println(e.getMessage());

} catch (InvalidNameException e) {

System.out.println(e.getMessage());

}

}

}

OR

Q.2 Write a program that displays the number of characters, lines and words of a file.

Solution:

import java.io.\*;

public class File {

public static void main(String[] args)

throws IOException

{

File file = new File("C:\\Users\\VE\\Desktop\\FY\\Assig\\Java Slips\\Test.txt");

FileInputStream fileInputStream = new FileInputStream(file);

InputStreamReader inputStreamReader = new InputStreamReader(fileInputStream);

BufferedReader bufferedReader = new BufferedReader(inputStreamReader);

String line;

int wordCount = 0;

int characterCount = 0;

int paraCount = 0;

int whiteSpaceCount = 0;

int sentenceCount = 0;

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

if (line.equals("")) {

paraCount += 1;

}

else {

characterCount += line.length();

String words[] = line.split("\\s+");

wordCount += words.length;

whiteSpaceCount += wordCount - 1;

String sentence[] = line.split("[!?.:]+");

sentenceCount += sentence.length;

}

}

if (sentenceCount >= 1) {

paraCount++;

}

System.out.println("Total word count = "+ wordCount);

System.out.println("Total number of sentences = "+ sentenceCount);

System.out.println("Total number of characters = "+ characterCount);

System.out.println("Number of paragraphs = "+ paraCount);

System.out.println("Total number of whitespaces = "+ whiteSpaceCount);

}

}

Slip 14

Q1. Define a class MyNumber having one private integer data member. Write a parameterized constructor to initialize to a value. Write isOdd() to check given number is odd or not. Use command line argument to pass a value to the object.

Solution:

public class MyNumberisOdd {

private int num;

public MyNumberisOdd(int num) {

this.num = num;

}

public boolean isOdd() {

return num % 2 != 0;

}

public static void main(String[] args) {

if (args.length == 0) {

System.out.println("No argument provided.");

return;

}

int value = Integer.parseInt(args[0]);

MyNumberisOdd myNumber = new MyNumberisOdd(value);

if (myNumber.isOdd()) {

System.out.println(value + " is odd.");

} else {

System.out.println(value + " is not odd.");

}

}

}

Q.2 Write a java program to design a following GUI. Use appropriate Layout and Components.

OR

Q.2 Write a java program to accept a number from the user, if number is zero then throw user defined exception ―Number is 0, otherwise check whether no is divisible by 2.

Solution:

import java.util.Scanner;

class NumberIsZeroException extends Exception {

public NumberIsZeroException(String errorMessage) {

super(errorMessage);

}

}

public class MainDivBy2 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a number: ");

int number = scanner.nextInt();

try {

if (number == 0) {

throw new NumberIsZeroException("Number is 0");

} else if (number % 2 == 0) {

System.out.println("Number is divisible by 2");

} else {

System.out.println("Number is not divisible by 2");

}

} catch (NumberIsZeroException e) {

System.out.println(e.getMessage());

}

}

}

Slip 15

Q.1 Construct a Linked List having names of Fruits: Apple, Banana, Guava and Orange. Display the contents of the List using an Iterator.

Solution:

import java.util.Iterator;

import java.util.LinkedList;

public class Fruits{

public static void main(String[] args) {

LinkedList<String> fruits = new LinkedList<>();

fruits.add("Apple");

fruits.add("Banana");

fruits.add("Guava");

fruits.add("Orange");

System.out.println("Contents of the Linked List:");

Iterator<String> iterator = fruits.iterator();

while (iterator.hasNext()) {

System.out.println(iterator.next());

}

}

}

Q.2 Write a Program to design following GUI by using swing component JComboBox. On click of show button display the selected language on JLabel.

OR

Q.2 Write a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate userdefined exception ―Age Not Within The Range.

Solution:

class AgeNotWithinRangeException extends Exception {

public AgeNotWithinRangeException(String message) {

super(message);

}

}

class InvalidNameException extends Exception {

public InvalidNameException(String message) {

super(message);

}

}

class Student {

private int rollNo;

private String name;

private int age;

private String course;

public Student(int rollNo, String name, int age, String course) throws AgeNotWithinRangeException, InvalidNameException {

this.rollNo = rollNo;

if (age < 15 || age > 21) {

throw new AgeNotWithinRangeException("Age Not Within The Range");

} else {

this.age = age;

}

if (!isValidName(name)) {

throw new InvalidNameException("Name not valid");

} else {

this.name = name;

}

this.course = course;

}

private boolean isValidName(String name) {

return name.matches("[a-zA-Z]+");

}

public void displayStudentDetails() {

System.out.println("Roll Number: " + rollNo);

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

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

System.out.println("Course: " + course);

}

}

public class StudentAge {

public static void main(String[] args) {

try {

Student student = new Student(101, "JohnDoe", 18, "Computer Science");

student.displayStudentDetails();

} catch (AgeNotWithinRangeException e) {

System.out.println(e.getMessage());

} catch (InvalidNameException e) {

System.out.println(e.getMessage());

}

}

}

Slip 16

Q.1 Construct a Linked List having names of Fruits: Apple, Banana, Guava and Orange. Display the contents of the List in reverse order using ListIterator.

Solution:

import java.util.Iterator;

import java.util.LinkedList;

public class Fruits{

public static void main(String[] args) {

LinkedList<String> fruits = new LinkedList<>();

fruits.add("Apple");

fruits.add("Banana");

fruits.add("Guava");

fruits.add("Orange");

System.out.println("Contents of the Linked List:");

Iterator<String> iterator = fruits.iterator();

while (iterator.hasNext()) {

System.out.println(iterator.next());

}

}

}

Q.2 Write a program to design following GUI using JTextArea. Write a code to display number of words and characters of text in JLabel. Use JScrollPane to get scrollbars for JTextArea.

OR

Q.2 Write a program that displays the number of characters, lines and words of a file.

Solution:

import java.io.\*;

public class File {

public static void main(String[] args)

throws IOException

{

File file = new File("C:\\Users\\VE\\Desktop\\FY\\Assig\\Java Slips\\Test.txt");

FileInputStream fileInputStream = new FileInputStream(file);

InputStreamReader inputStreamReader = new InputStreamReader(fileInputStream);

BufferedReader bufferedReader = new BufferedReader(inputStreamReader);

String line;

int wordCount = 0;

int characterCount = 0;

int paraCount = 0;

int whiteSpaceCount = 0;

int sentenceCount = 0;

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

if (line.equals("")) {

paraCount += 1;

}

else {

characterCount += line.length();

String words[] = line.split("\\s+");

wordCount += words.length;

whiteSpaceCount += wordCount - 1;

String sentence[] = line.split("[!?.:]+");

sentenceCount += sentence.length;

}

}

if (sentenceCount >= 1) {

paraCount++;

}

System.out.println("Total word count = "+ wordCount);

System.out.println("Total number of sentences = "+ sentenceCount);

System.out.println("Total number of characters = "+ characterCount);

System.out.println("Number of paragraphs = "+ paraCount);

System.out.println("Total number of whitespaces = "+ whiteSpaceCount);

}

}

Slip 17

Q1. Write a program create class as MyDate with dd,mm,yy as data members. Write parameterized constructor. Display the date in dd-mm-yy format. (Use this keyword).

Solution:

public class MyDateSlip5 {

int dd;

int mm;

int yy;

public MyDateSlip5(int dd, int mm, int yy) {

this.dd = dd;

this.mm = mm;

this.yy = yy;

}

public void displayDate() {

System.out.println(this.dd + "-" + this.mm + "-" + this.yy);

}

public static void main(String[] args) {

MyDateSlip5 myDate = new MyDateSlip5(1, 1, 2023);

myDate.displayDate();

}

}

Q.2 . Write a program to copy the contents from one file into another file in upper case.

OR

Q.2 Create an abstract class Shape with methods area & volume. Derive a class Sphere (radius). Calculate and display area and volume.

Solution:

import java.util.Scanner;

public class MainShape {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the radius of the sphere: ");

double radius = scanner.nextDouble();

Sphere sphere = new Sphere(radius);

System.out.println("Surface Area of the sphere is: " + sphere.getSurfaceArea());

System.out.println("Volume of the sphere is: " + sphere.getVolume());

}

}

abstract class Shape {

abstract double getSurfaceArea();

abstract double getVolume();

}

class Sphere extends Shape {

final double PI = 3.142;

private double radius;

public Sphere(double radius) {

this.radius = radius;

}

public double getSurfaceArea() {

return 4 \* PI \* Math.pow(radius, 2);

}

public double getVolume() {

return (4 / 3) \* PI \* Math.pow(radius, 3);

}

}

Slip 18

Q1. Write a program to accept a number from user. Check whether number is prime or not.

Solution:

import java.util.\*;

public class PrimeNumber {

public static void main(String[] args) {

int count = 0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter a number to check prime number");

int num = sc.nextInt();

if(num == 2) {

System.out.println(num + " is a prime number");

}

else {

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

if(num % i == 0) {

count++;

}

}

if(count == 2) {

System.out.println(num + " is a prime number");

} else {

System.out.println(num + " is not a prime number");

}

}

}

}

Q.2 Write a program which define class Product with data member as id, name and price. Store the information of 5 products and Display the name of product having minimum price (Use array of object).

Solution:

import java.util.\*;

class Product

{

int id;

String name;

double price;

public Product(int id,String name,double price)

{

this.id=id;

this.name=name;

this.price=price;

}

}

public class MainProdPrice

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("enter number of products: ");

int n=sc.nextInt();

Product[] p = new Product[n];

for(int i = 0;i<n;i++)

{

System.out.println("Enter product ID: ");

int id=sc.nextInt();

System.out.println("Enter product name: ");

String name=sc.next();

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

double price=sc.nextDouble();

Product prod=new Product(id,name,price);

p[i]=prod;

}

double minPrice=p[0].price;

Product minPriceProduct=p[0];

for(Product prod: p)

{

if(prod.price<minPrice)

{

minPrice=prod.price;

minPriceProduct=prod;

}

}

System.out.println("Product with minimum price is: "+minPriceProduct.name);

}

}

OR

Q.2 Write a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate userdefined exception ―Age Not Within The Range.

Solution:

class AgeNotWithinRangeException extends Exception {

public AgeNotWithinRangeException(String message) {

super(message);

}

}

class InvalidNameException extends Exception {

public InvalidNameException(String message) {

super(message);

}

}

class Student {

private int rollNo;

private String name;

private int age;

private String course;

public Student(int rollNo, String name, int age, String course) throws AgeNotWithinRangeException, InvalidNameException {

this.rollNo = rollNo;

if (age < 15 || age > 21) {

throw new AgeNotWithinRangeException("Age Not Within The Range");

} else {

this.age = age;

}

if (!isValidName(name)) {

throw new InvalidNameException("Name not valid");

} else {

this.name = name;

}

this.course = course;

}

private boolean isValidName(String name) {

return name.matches("[a-zA-Z]+");

}

public void displayStudentDetails() {

System.out.println("Roll Number: " + rollNo);

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

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

System.out.println("Course: " + course);

}

}

public class StudentAge {

public static void main(String[] args) {

try {

Student student = new Student(101, "JohnDoe", 18, "Computer Science");

student.displayStudentDetails();

} catch (AgeNotWithinRangeException e) {

System.out.println(e.getMessage());

} catch (InvalidNameException e) {

System.out.println(e.getMessage());

}

}

}

Slip 19

Q.1Write a program to accept a number from user.Find square and cube of that number.Define method numberSquare() to find square and numberCube() to find cube of a number.

Solution:

import java.util.Scanner;

public class NumSqrCube {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a number: ");

int num = scanner.nextInt();

System.out.println("Square of " + num + " is: " + numberSquare(num));

System.out.println("Cube of " + num + " is: " + numberCube(num));

}

public static int numberSquare(int num) {

return num \* num;

}

public static int numberCube(int num) {

return num \* num \* num;

}

}

Q.2 Write a program which define class Employee with data member as id, name and salary Store the information of 'n' employees and Display the name of employee having maximum salary (Use array of object).

Solution:

import java.util.\*;

class Employee

{

int id;

String name;

double salary;

public Employee(int id,String name,double salary)

{

this.id=id;

this.name=name;

this.salary=salary;

}

}

public class MainEmpSalary

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter number of employees: ");

int n=sc.nextInt();

Employee[] e = new Employee[n];

for(int i = 0;i<n;i++)

{

System.out.println("Enter employee ID: ");

int id=sc.nextInt();

System.out.println("Enter employee name: ");

String name=sc.next();

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

double salary=sc.nextDouble();

Employee emp=new Employee(id,name,salary);

e[i]=emp;

}

double maxSalary=e[0].salary;

Employee maxSalaryEmployee=e[0];

for(Employee emp: e)

{

if(emp.salary>maxSalary)

{

maxSalary=emp.salary;

maxSalaryEmployee=emp;

}

}

System.out.println("Employee with maximum salary is: "+maxSalaryEmployee.name);

}

}

OR

Q.2 Write a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate userdefined exception ―Age Not Within The Range.

Solution:

class AgeNotWithinRangeException extends Exception {

public AgeNotWithinRangeException(String message) {

super(message);

}

}

class InvalidNameException extends Exception {

public InvalidNameException(String message) {

super(message);

}

}

class Student {

private int rollNo;

private String name;

private int age;

private String course;

public Student(int rollNo, String name, int age, String course) throws AgeNotWithinRangeException, InvalidNameException {

this.rollNo = rollNo;

if (age < 15 || age > 21) {

throw new AgeNotWithinRangeException("Age Not Within The Range");

} else {

this.age = age;

}

if (!isValidName(name)) {

throw new InvalidNameException("Name not valid");

} else {

this.name = name;

}

this.course = course;

}

private boolean isValidName(String name) {

return name.matches("[a-zA-Z]+");

}

public void displayStudentDetails() {

System.out.println("Roll Number: " + rollNo);

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

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

System.out.println("Course: " + course);

}

}

public class StudentAge {

public static void main(String[] args) {

try {

Student student = new Student(101, "JohnDoe", 18, "Computer Science");

student.displayStudentDetails();

} catch (AgeNotWithinRangeException e) {

System.out.println(e.getMessage());

} catch (InvalidNameException e) {

System.out.println(e.getMessage());

}

}

}

Slip 20

Q.1 Create a Hash table containing Employee name and Salary. Display the details of the hash table.

Solution:

import java.util.\*;

import java.io.\*;

public class EmployeeHashTable

{

public static void main(String[] args)

{

String name;

float sal;

Hashtable ht=new Hashtable();

Scanner sc = new Scanner(System.in);

System.out.println("Enter no.of Employees: ");

int n=sc.nextInt();

for(int i=0;i<n;i++)

{

System.out.println("Enter name of Employee: ");

name=sc.next();

System.out.println("Enter salary of Employee: ");

sal=sc.nextFloat();

ht.put(name,sal);

}

System.out.println("Hash table = "+ht);

}

}

Q.2 Write a program to design the following GUI using Swing components. On click of submit button check whether user has entered all the details or not. If any details are missing then display appropriate message on screen using label.

OR

Q.2 Define an interface “Operation” which has methods area(),volume(). Define a constant PI having value 3.142. Create a class circle (member – radius) which implements this interface. Calculate and display the area and volume.

Solution:

import java.util.Scanner;

public class InterfaceOperation {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the radius of the circle: ");

double radius = scanner.nextDouble();

Circle circle = new Circle(radius);

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

System.out.println("Volume of the circle is: " + circle.volume());

}

}

interface Operation {

double area();

double volume();

}

class Circle implements Operation {

final double PI = 3.142;

private double radius;

public Circle(double radius) {

this.radius = radius;

}

public double area() {

return PI \* radius \* radius;

}

public double volume() {

return (4 / 3) \* PI \* Math.pow(radius, 3);

}

}