**Q.1 Write a Program that displays Welcome to AMTICS.**

**CODE:**

public class practical1

{

public static void main(String[] args)

{

System.out.println("Welcome To AMTICS!");

}

}

**Q.2 Write a program for calculator.**

**CODE:**

import java.util.Scanner;

public class practical2

{

public static void main(String[] args)

{

double a,b,result;

char op;

Scanner input = new Scanner(System.in);

System.out.print("-->> Enter First number: ");

a = input.nextInt();

System.out.print("-->> Enter a operator(+,-,\*,/,%): ");

op = input.next().charAt(0);

System.out.print("-->> Enter Second number: ");

b = input.nextInt();

switch(op)

{

case '+':

{

result = a + b;

System.out.println(a + " + " + b + " = " + result);

break;

}

case '-':

{

result = a - b;

System.out.println(a + " - " + b + " = " + result);

break;

}

case '\*':

{

result = a \* b;

System.out.println(a + " \* " + b + " = " + result);

break;

}

case '/':

{

result = a / b;

System.out.println(a + " / " + b + " = " + result);

break;

}

case '%':

{

result = a % b;

System.out.println(a + " % " + b + " = " + result);

break;

}

default :

{

System.out.println("INVALID OPERATOR!");

break;

}

}

input.close();

}

}

**Q.3 Write a program that reads a number in meters, converts it to feet, and displays the result.**

**CODE:**

import java.util.Scanner;

public class practical3

{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

System.out.print("-->> Enter the distance in meters: ");

int m = input.nextInt();

double feet = m \* 3.28084;

System.out.println(m + " meters --> " + feet + " feet");

input.close();

}

}

**Q.4 Write a program to print inputs given from command line arguments on the console.**

**CODE:**

public class practical4

{

public static void main(String[] args)

{

for(String str:args)

{

System.out.print(str);

}

*// for(int i=0; i<args.length; i++)*

*// System.out.println(args[i]);*

}

}

**Q.5 Write a program that prompts the user to enter three integers and display the integers in decreasing order.**

**CODE:**

import java.util.Scanner;

public class practical5

{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

int arr[]= new int[3];

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

{

System.out.print("-->> Enter " +(i+1) + " Number: ");

arr[i] = input.nextInt();

}

for(int i=0; i<3-1; i++)

{

for(int j=0; j<3-i-1; j++)

{

if(arr[j] < arr[j+1])

{

int temp = arr[j+1];

arr[j+1] = arr[j];

arr[j] = temp;

}

}

}

System.out.print("Decreasing Order of the entered elements --> ");

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

{

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

}

System.out.println();

input.close();

}

}

**Q.6 Write a program that prompts the user to enter a letter and check whether a letter is a vowel or constant.**

**CODE:**

import java.util.Scanner;

public class practical6

{

public static void main(String[] args)

{

char a;

Scanner input = new Scanner(System.in);

System.out.print("-->> Enter a Letter: ");

a = input.next().charAt(0);

if(a == 'A' || a == 'a' || a == 'E' || a == 'e' || a == 'I' || a == 'i' || a == 'O' || a == 'o' || a == 'U' || a == 'u')

{

System.out.println("The Entered Character is a 'VOWEL'.");

}

else

{

System.out.println("The Entered Character is a 'CONSONANT'.");

}

input.close();

}

}

**Q.7 Assume a vehicle plate number consists of three uppercase letters followed by four digits. Write a program to generate a plate number.**

**CODE:**

public class practical7

{

public static void main(String[] args)

{

int l1 = 'A' + (int)(Math.random() \* ('Z' - 'A'));

int l2 = 'A' + (int)(Math.random() \* ('Z' - 'A'));

int l3 = 'A' + (int)(Math.random() \* ('Z' - 'A'));

int d1 = (int)(Math.random() \* 10);

int d2 = (int)(Math.random() \* 10);

int d3 = (int)(Math.random() \* 10);

int d4 = (int)(Math.random() \* 10);

System.out.println("" + ((char)(l1)) + ((char)(l2)) +

((char)(l3)) + d1 + d2 + d3 + d4);

}

}

**Q.8 Write a test program that prompts the user to enter ten numbers, invoke a method to reverse the numbers, display the numbers.**

**CODE:**

import java.util.Scanner;

public class practical8

{

public static void reverse(int arr[])

{

int left = 0;

int right = arr.length - 1;

while (left < right)

{

int temp = arr[left];

arr[left] = arr[right];

arr[right] = temp;

left++;

right--;

}

}

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

int number[] = new int[10];

for(int i=0; i<10; i++)

{

System.out.print("-->> Enter "+ (i+1) +" Number: ");

number[i] = input.nextInt();

}

reverse(number);

System.out.print("--> Reversed Numbers: ");

for(int num: number)

{

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

}

System.out.println();

input.close();

}

}

**Q.9 Write a program that generate 6\*6 two-dimensional matrices, filled with 0’s and 1’s , display the matrix, check every row and column have an odd number of 1’s.**

**CODE:**

public class practical9

{

public static int[][] generateMatrix()

{

int[][] matrix = new int[6][6];

for (int i = 0; i < 6; i++)

{

for (int j = 0; j < 6; j++)

{

matrix[i][j] = (int)((Math.random() \* 5) % 2);

}

}

return matrix;

}

public static void displayMatrix(int[][] matrix)

{

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

for (int i = 0; i < 6; i++)

{

for (int j = 0; j < 6; j++)

{

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

}

System.out.println();

}

}

public static void main(String[] args)

{

int matrix[][];

int i, j, count;

matrix = generateMatrix();

displayMatrix(matrix);

System.out.println("\n--> Rows Having ODD number of 1s: ");

for (i = 0; i < 6; i++)

{

count = 0;

for (j = 0; j < 6; j++)

{

if (matrix[i][j] == 1)

{

count++;

}

}

if (count % 2 != 0)

{

System.out.println("Row - " + (i + 1) + " have ODD no of 1s");

}

}

System.out.println("\n--> Columns Having ODD no of 1s: ");

for (i = 0; i < 6; i++)

{

count = 0;

for (j = 0; j < 6; j++)

{

if (matrix[j][i] == 1)

{

count++;

}

}

if (count % 2 != 0)

{

System.out.println("Column - " + (i + 1) + " have ODD number of 1s");

}

}

}

}

**Q.10 Write an application that illustrates method overriding in the same package and different packages.**

**CODE:**

*// Beverage.java (Same Package)*

package Cafe;

public class Beverage

{

public void prepare()

{

System.out.println("Preparing a basic beverage.");

}

}

*// Coffee.java (Same Package)*

package Cafe;

public class Coffee extends Beverage

{

public void prepare()

{

System.out.println("Brewing a cup of coffee.");

}

public void addCream()

{

System.out.println("Adding cream to the coffee.");

}

}

*// Tea.java (Same Package)*

package Cafe;

public class Tea extends Beverage

{

public void prepare()

{

System.out.println("Steeping a cup of tea.");

}

public void addLemon()

{

System.out.println("Adding lemon to the tea.");

}

}

*// ChaiTea.java (Different Package)*

package special;

import Cafe.Tea;

public class ChaiTea extends Tea

{

public void prepare()

{

System.out.println("Brewing a cup of chai tea.");

}

}

*// practical10.java*

package Cafe;

import special.ChaiTea;

public class practical10

{

public static void main(String[] args)

{

Beverage basicBeverage = new Beverage();

Coffee coffee = new Coffee();

Tea tea = new Tea();

ChaiTea chaiTea = new ChaiTea();

System.out.println("Basic Beverage:");

basicBeverage.prepare();

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

coffee.prepare();

coffee.addCream();

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

tea.prepare();

tea.addLemon();

System.out.println("\nChai Tea:");

chaiTea.prepare();

}

}

**Q.11 Describe abstract class called Shape which has three subclasses say Triangle, Rectangle, Circle. Define one method area() in the abstract class and override this area() in these three subclasses to calculate for specific object i.e. area() of Triangle subclass should calculate area of triangle etc. Same for Rectangle and Circle.**

**CODE:**

import java.util.Scanner;

abstract class Shape

{

public abstract double area();

}

class Triangle extends Shape

{

private double base;

private double height;

public Triangle(double base, double height)

{

*this*.base = base;

*this*.height = height;

}

public double area()

{

return 0.5 \* base \* height;

}

}

class Rectangle extends Shape

{

private double length;

private double width;

public Rectangle(double length, double width)

{

*this*.length = length;

*this*.width = width;

}

public double area()

{

return length \* width;

}

}

class Circle extends Shape

{

private double radius;

public Circle(double radius)

{

*this*.radius = radius;

}

public double area()

{

return Math.PI \* radius \* radius;

}

}

public class practical11

{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

System.out.print("-->> Enter the Base and Height for the Triangle: ");

double b = input.nextDouble();

double h = input.nextDouble();

System.out.print("-->> Enter the Length and Width for the Rectangle: ");

double l = input.nextDouble();

double w = input.nextDouble();

System.out.print("-->> Enter the Radius for the Circle: ");

double r = input.nextDouble();

Shape triangle = new Triangle(b, h);

Shape rectangle = new Rectangle(l, w);

Shape circle = new Circle(r);

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

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

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

}

}

**Q.12 Write a program in Java to demonstrate implementation of multiple inheritance using interfaces.**

**CODE:**

import java.util.Scanner;

interface Shape

{

double getArea();

}

interface Color

{

String getColor();

}

class Rectangle implements Shape, Color

{

private double length;

private double width;

private String color;

public Rectangle(double length, double width, String color)

{

*this*.length = length;

*this*.width = width;

*this*.color = color;

}

public double getArea()

{

return length \* width;

}

public String getColor()

{

return color;

}

}

public class practical12

{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

System.out.print("-->> Enter the Length of the Rectangle: ");

double l = input.nextDouble();

System.out.print("-->> Enter the Width of the Rectangle: ");

double w = input.nextDouble();

System.out.print("-->> Enter the Color of the Rectangle: ");

String c = input.next();

Rectangle redRectangle = new Rectangle(l, w, c);

double area = redRectangle.getArea();

String color = redRectangle.getColor();

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

System.out.println("Rectangle Color: " + color);

input.close();

}

}

**Q.13 Write a program in Java to develop user defined exception for ‘Divide by Zero’ error.**

**CODE:**

class DivideByZeroException extends Exception

{

public DivideByZeroException(String message)

{

*super*(message);

}

}

public class practical13

{

public static void main(String[] args)

{

try

{

int result = divide(10, 0);

System.out.println("Result: " + result);

}

catch (DivideByZeroException e)

{

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

}

}

public static int divide(int numerator, int denominator) throws DivideByZeroException

{

if (denominator == 0)

{

throw new DivideByZeroException("Division by zero is not allowed.");

}

return numerator / denominator;

}

**}**

**Q.14 Write a program in Java to demonstrate multiple try block and multiple catch exception.**

**CODE:**

public class practical14

{

public static void main(String[] args)

{

try

{

int[] num = {1, 2, 3};

System.out.println("--> Element at index 5: " + num[5]);

}

catch (ArrayIndexOutOfBoundsException e)

{

System.err.println("---> ArrayIndexOutOfBoundsException: " + e.getMessage());

}

try

{

int result = divide(10, 0);

System.out.println("--> Result: " + result);

}

catch (ArithmeticException e)

{

System.err.println("---> ArithmeticException: " + e.getMessage());

}

try

{

String str = null;

System.out.println("--> Length of the string: " + str.length());

}

catch (NullPointerException e)

{

System.err.println("---> NullPointerException: " + e.getMessage());

}

}

public static int divide(int numerator, int denominator)

{

return numerator / denominator;

}

}

**Q.15 Write a program that executes two threads. One thread displays “Thread1” every 2,000 milliseconds, and the other displays “Thread2” every 4,000 milliseconds. Create the threads by extending the Thread class.**

**CODE:**

class Thread1 extends Thread

{

public void run()

{

while (true)

{

System.out.println("Thread1");

try

{

Thread.sleep(2000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

}

}

}

class Thread2 extends Thread

{

public void run()

{

while (true)

{

System.out.println("Thread2");

try

{

Thread.sleep(4000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

}

}

}

public class practical15

{

public static void main(String[] args)

{

Thread thread1 = new Thread1();

Thread thread2 = new Thread2();

thread1.start();

thread2.start();

}

}