|  |
| --- |
| **Program List** |
| 1. Wap a java program to print "Hello World." WAP in java to print the values of various primitive data types. 2. WAP in java to demonstrate operator precedence. 3. Write a program to read an integer value through Scanner class and find addition of   two numbers using function.   1. Write a program that uses length property for displaying any number of command line arguments 2. Write a program to create constructor of a class and initialize values in it and later print them. 3. Wap in java to sort n numbers using bubble sort. 4. WAP I java to find addition of two Matrices. 5. Write a java code to implement the concept of method overloading. 6. Create a class Shape and override area () method to calculate area of rectangle, square and circle. 7. Write a program to show how method overriding is implemented in java. Create a child class of Animal named ‘Bird’ and override the parent class methods. Add a new method named ﬂy (). 8. Write a program to implement the concept of abstract classes. 9. Write programs for Exception handling using try, catch, throw and finally. 10. Write a java program to implement the usage of customized exceptions 11. Implement concept of multithreading in Java by 12. Extending Thread class 13. Implementing Runnable interface 14. Write a java code to implement the concept of simple inheritance, multilevel inheritance, and hierarchical inheritance. 15. Implementation of Linear data structure using Programming language. 16. Implementation of Non- Linear data structure using Programming language. 17. Implementation of searching and sorting techniques using Programming language. 18. Implementation of Graph traverses using Programming language. 19. Implementation of Trees Concepts using Programming language. 20. Implementation of Dynamic Programming Concepts. 21. Implementation of Shortest-path algorithms Concepts. 22. Implementation Advanced project Concepts using programming language. |
|  |

**Continuous Assessment Pattern**

|  |  |  |
| --- | --- | --- |
| **Internal Assessment (IA)** | **End Term Test (ETE)** | **Total Marks** |
| 100 |  | 100 |

**Coordinator:**

Dr. J.N SINGH



EXP.01

Java program to print "Hello World." WAP in java to

print the values of various primitive data types

# CODE(DATA TYPES):

public class Test {

int k; double d; ﬂoat f;

boolean istrue; String p;

public void printValue() {

System.out.println("int default value = "+ k); System.out.println("double default value = "+ d); System.out.println("ﬂoat default value = "+ f); System.out.println("boolean default value = "+ istrue); System.out.println("String default value = "+ p);

}

}

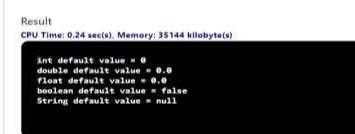
public class HelloWorld {

public static void main(String argv[]) { Test = new Test(); test.printValue();

}

}

# OUTPUT:





EXP.02

WAP in java to demonstrate operator

precedence.

## CODE:

public class Precedence {

public static void main(String[] args) {

int a = 10, b = 5, c = 1, result;

result = a-++c-++b;

System.out.println(result);

}

}

## Exp3: Write program find addition of two number using function.

import java.util.Scanner; public class AddTwoNumbers

{

public static void main(String[] args)

{

int num1, num2, sum;

Scanner sc = new Scanner(System.*in*); System.*out*.println("Enter first number : "); num1 = sc.nextInt(); System.*out*.println("Enter second number : "); num2 = sc.nextInt();

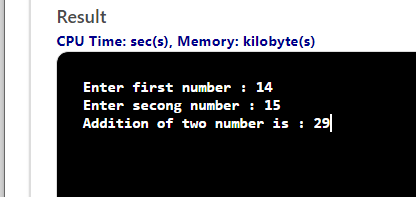
sum = *addTwo*(num1, num2); System.*out*.println("Addition of two numbers : " + sum); sc.close();

}

public static int addTwo(int a, int b)

{

int sum = a + b;



return sum;

}

}



EXP.04

Write a program read data from keyboard and find sum and their average.

Import java.util.Scanner;

public class Average {

public static void main(String[] args)

{

int n, count = 1;

ﬂoat xF, averageF, sumF = 0;

Scanner sc = new Scanner(System.in); System.out.println("Enter the value of n");

n= sc.nextInt();

while (count <= n)

{

System.out.println("Enter the "+count+" number?");

xF= sc.nextInt();

sumF += xF;

++count;

}

averageF = sumF/n;

System.out.println("The Average is"+averageF);

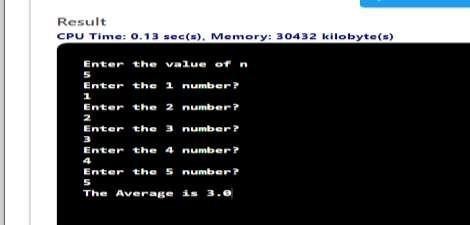
}

}

# INPUT:

Number of all elements = 5

# OUTPUT:



Program 6: Wap in java to sort n numbers using bubble sort.

## CODE:

public class SortAsc {

public static void main(String[] args) {

//Initialize array

int [] arr = new int [] {5, 2, 8, 7, 1}; int temp

= 0;

//Displaying elements of original array System.out.println("Elements of original array: "); for (int i = 0; i < arr.length; i++) {

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

}

//Sort the array in ascending order for (int i = 0; i < arr.length; i++) {

for (int j = i+1; j < arr.length; j++) { if(arr[i] > arr[j]) {

temp = arr[i]; arr[i] = arr[j]; arr[j] = temp;

}

}

}

System.out.println();

//Displaying elements of array after sorting System.out.println("Elements of array sorted in ascending order: "); for (int i = 0; i < arr.length; i++) {

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

}

}

}

# OUTPUT:

Elements of original array:

5 2 8 7 1

Elements of array sorted in ascending order: 1 2 5 7 8



EXP.7

WAP in java to perform matrix addition using two 2D

arrays.

class MatixAdditionExample{ public static void main(String args[]){ int a[][]={{1,3,4},

{2,4,3},{3,4,5}}; intb[][]={{1,3,4},{2,4,3},{1,2,4}};

int c[][]=new int[3][3]; for(int i=0;i<3;i++)

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

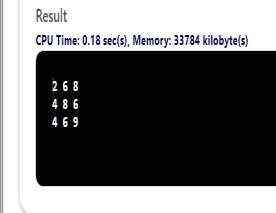
c[i][j]=a[i][j]+b[i][j];

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

}

}







EXP.8

WAP in java that implements method

overloading.

# CODE:

class MethodOverloading {

private static void display(int a)

{ System.out.println("Got Integerdata.");

}

private static void display(String a)

{ System.out.println("Got Stringobject.");

}

public static void main(String[] args)

{ display(1); display("Hello");

}

}

EXP.9

public class OverloadDemo

{

Create a class Shape and override area() method to calculate area of rectangle, square and circle

void area(ﬂoat x)

{

System.out.println("the area of the square is "+Math.pow(x, 2)+" sq units");

}

void area(ﬂoat x, ﬂoat y)

{

System.out.println("the area of the rectangle is "+x\*y+" sq units");

}

void area(double x)

{

double z = 3.14 \* x \* x;

System.out.println("the area of the circle is "+z+" sq units");

}

}

class Overload

{

public static void main(String args[])

{

OverloadDemo ob = new OverloadDemo(); ob.area(5);

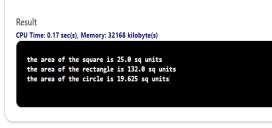
ob.area(11,12);

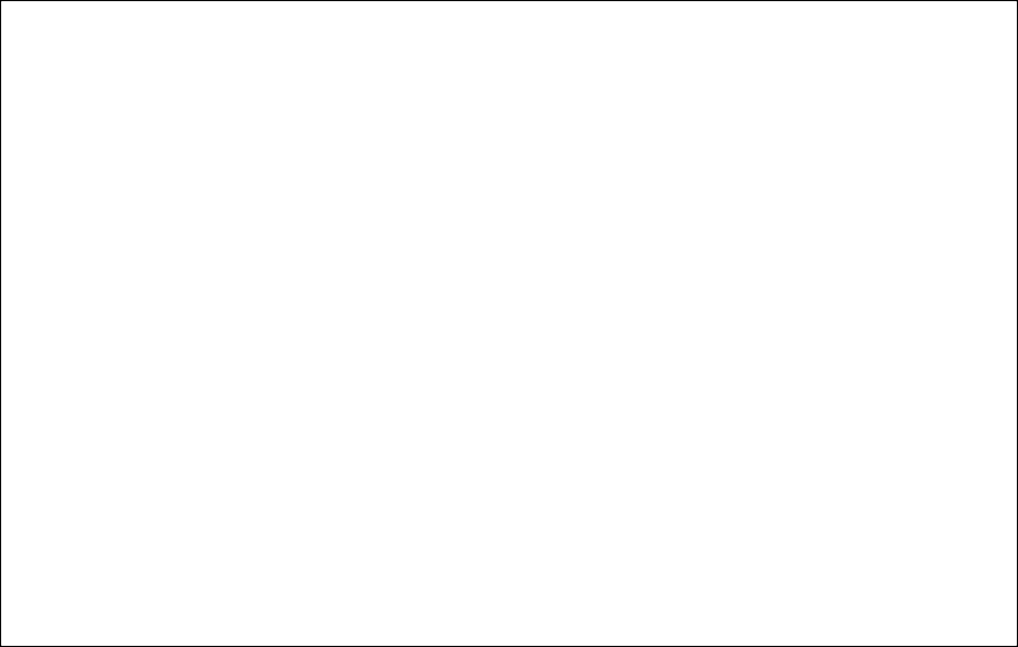
ob.area(2.5);

}

}

# OUTPUT:





EXP.10

Create a child class of Animal named ‘Bird’ and

override the parent class methods. Add a new method

named ﬂy().

Create an instance of Animal class and invoke the eat

and sleep methods using this object.

## CODE:

class Animal {

protected String type = "animal";

public void eat ()

{

System.out.println("I can eat");

}

public void sleep ()

{

System.out.println("I can sleep");

}

}

class Bird extends Animal {

public void eat()

{

System.out.println("I eat Bird food"

)

}

public void ﬂy()

{

System.out.println("I can ﬂy");

}

}

class Main {

public static void main(String[] args) {

Bird bird1 = new Bird(); bird1.eat();

bird1.sleep(); bird1.bark();

}

}



Program:11

Wap in java for abstract class to find area of different shapes

import java.util.\*;

abstract class Diagram

{

private int d1,d2;

abstract public void areaCalculation(); public void readData()

{

Scanner sin=new Scanner(System.in); System.out.println("Enter two dimensions:"); d1=sin.nextInt();

d2=sin.nextInt();

}

public void displayData()

{

System.out.println("D1:"+d1+"\nD2:"+d2);

}

public int getD1()

{

return d1;

}

public int getD2()

{

return d2;

}

}

class Rectangle extends Diagram

{

private int area;

public void areaCalculation()

{

int x=super.getD1();

int y=super.getD2(); area=x\*y;

}

public void displayData()

{

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

}

}

class Triangle extends Diagram

{

private int area;

public void areaCalculation()

{

int x=super.getD1();

int y=super.getD2(); area=(1/2)\*x\*y;

}

public void displayData()

{

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

}

}

class Ellipse extends Diagram

{

private int area;

public void areaCalculation()

{

int x=super.getD1(); int y=super.getD2(); area=(1/2)\*x\*y;

}

public void displayData()

{

super.displayData();

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

}

}

public class Draw

{

public static void main(String args[])

{

Scanner sin=new Scanner(System.in); Diagram d1;

int ch; do

{

System.out.println("1.RECTANGLE(default)");

System.out.println("2.TRIANGLE"); System.out.println("3.ELLIPSE"); System.out.print("Enter ur choice:"); ch=sin.nextInt();

switch(ch)

{

case 1:

d1=new Rectangle();

break;

case 2:

d1=new Triangle();

break;

; case 3:

d1=new Ellipse();

break;

default:

d1=new Rectangle ();

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

break;

}

d1.readData(); d1.areaCalcultion(); d2.displayData();

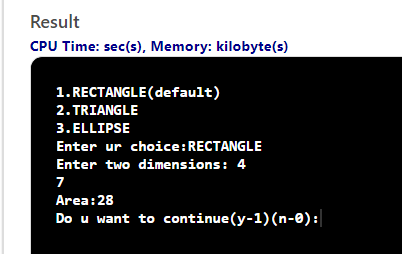
System.out.print("Do u want to continue(y-1)(n-0):"); ch=sin.nextInt();

}while(ch==1);

}

}

# OUTPUT:



## Program: 12

1. WAP in java demonstrating arithmetic exception and ArrayOutOf BoundsException using try catch block in java to demonstrate the use of nested try block.
2. WAP in java to demonstrate the use of throw and throws.

class Example1 {

public static void main(String args[]) { int num1, num2;

try {

num1 = 0;

num2 = 62 / num1; System.out.println(num2);

System.out.println("Hey I'm at the end of try block");

}

catch (ArithmeticException e) {

System.out.println("You should not divide a number by zero");

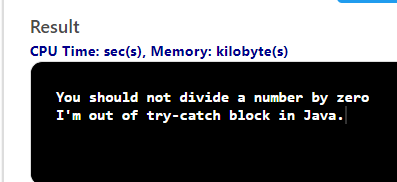
}

catch (Exception e)

{System.out.println("Exception occurred");

}

System.out.println("I'm out of try-catch block in Java.");



}

}

# OUTPUT-

## CODE: (b)

class ThrowExcep

{

static void fun()

{

try

{

throw new NullPointerException("demo");

}

catch(NullPointerException e)

{

System.out.println("Caught inside fun()."); throw e; // rethrowing the exception

}

}

public static void main(String args[])

{

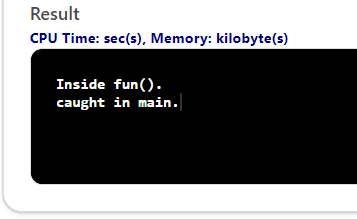
try

{

fun();

}catch(NullPointerException e)

{

System.out.println("Caught in main.");

}

}

}

**Program13:**

Write a java program to implement the usage of customized exceptions.

package Exception;

@SuppressWarnings("serial")

class InvalidException extends Exception

{

InvalidException(String str)

{

super(str);

}}

public class UserDefinedException

{

public static void main(String args[])

{

int age=17;

try

{

chaeckage( age);

}catch(Exception e)

{

System.out.println(e);

}}

public static void chaeckage(int age)throws InvalidException

{

if(age<18)

{

throw new InvalidException("age is less than 18");

}

else{

throw new InvalidException(" he is eligible for voting");

}

}

}

**Output:**

age is less than 18

Program 14:

Wap a program in java to implement concept of multithreading in Java by

(a) Extending Thread class

(b) Implementing Runnable interface

CODE:

class Multi extends Thread

{

public void run()

{

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

{

System.out.println(i);

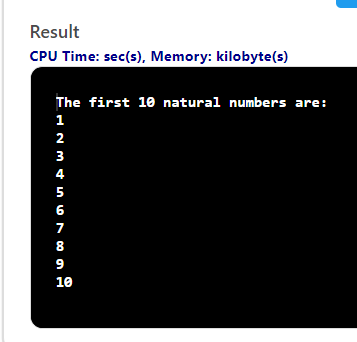
}

public static void main(String args[]){ Multi t1=new Multi();

t1.start();

}

}

**Output :**

# CODE:

# (b)

**import** java. Lang.\*;

**class** thread1 **implements** Runnable

{

**public** **void** run()

{

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

{

System.***out***.println("thread1 " + i);

}}}

**class** thread2 **implements** Runnable

{

**public** **void** run()

{

**for**(**int** j=2;j<=10;j=j+2)

{

**try** {

Thread.*sleep*(2000);

System.***out***.println("thread2 " + j);

} **catch** (Interrupted Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

}

**public** **class** First{

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

{

thread1 th=**new** thread1();

Thread t1=**new** Thread(th);

thread2 th2=**new** thread2();

Thread t2=**new** Thread(th2);

t2.setPriority(10);

t1.setPriority(4);

t1.start();

t2.start();

}

}

**Output:**

thread1 0

thread1 1

thread1 2

thread1 3

thread1 4

thread1 5

Thread[main,5,main]

Thread-0

Thread-1

thread2 2

thread2 4

thread2 6

thread2 8

thread2 10

Program15:

**package** class1;

**class** pqr

{

**int** a,b;

pqr(**int** a,**int** b)

{

**this**.a= a;

**this**.b=b;

}

**void** display()

{

System.***out***.println(" sum is" + (a+b));

}

}

**class** Xyz **extends** pqr

{

**int** c;

Xyz(**int** a,**int** b,**int** c)

{

**super**(a,b);

**this**.c=c;

}

**void** display ()

{

**super**.display();

System.***out***.println(" sum is" + (a+b+c));

}

}

**public** **class** Inheritance5{

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

{

// **TODO** Auto-generated method stub

Xyz ob=**new** Xyz(20,30,40);

ob.display();

}

}

## Output:

sum is50

## sum is90

## Program16:

**package** class1;

**interface** area

{

**final** **static** **double** ***pi*** = 3.14;

**public** **double** calc(**double** x,**double** y);

**static** **void** display()

{

System.***out***.println(" interface1");

}

}

**interface** one **extends** area

{

**public** **void** display();

}

**class** rect **implements** area

{

**public** **double** calc(**double** x,**double** y)

{

**return**(x\*y);

}

}

**class** cir **implements** area,one

{

**public** **double** calc(**double** x,**double** y)

{

**return**(***pi***\*x\*x);

}

**public** **void** display()

{

System.***out***.println(" derived classs");

}

}

**class** VolumeSphere

{

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

{

rect r = **new** rect();

cir c = **new** cir();

area a;

//rect.display();

a = r;

System.***out***.println("\nArea of Rectangle is " +a.calc(10,20));

a = c;

System.***out***.println("\nArea of Circle is : " +a.calc(15,0));

}

## }