**PART A**

|  |  |
| --- | --- |
| **1** | **Write a Java program to find factorial of list of number reading input as command line argument.** |
| **2** | **Write a Java program to sort list of elements in ascending and descending order and show the exception handling.** |
| **3** | **Write a Java program to implement all string operations.** |
| **4** | **Write a Java program to find area of geometrical figures using method.** |
| **5** | **Write a Java program to implement constructor overloading by passing different**  n**umber of parameter of different types.** |
| **6** | **Write a Java Program to implement multiple inheritance using interface.** |
| **7** | **Write a Java program to implement keyboard events.** |
| **8** | **Write a Java program to implement thread priorities.** |

**PART B**

|  |  |
| --- | --- |
| 1 | **Write a Java program to display all prime numbers between two limits.** |
| 2 | **Write a Java program to print the elements of an array in reverse order** |
| 3 | **Write a Java program to implement Rhombus pattern reading the limit form user.** |
| 4 | **Write a Java program to illustrate the concept of inheritance** |
| 5 | **Write a Java Program to find the number of the words in the given text file** |
| 6 | **Write a Java program to calculate bonus for different departments using method overriding** |
| 7 | **Write a program in java to enter the number through command line argument if first and second number is not entered it will generate the exception. Also divide the first number with second number and generate the arithmetic exception**. |
| 8 | **Write a Java program to implement thread, applets and graphics by implementing animation of ball moving** |

**BCA504P – JAVA Programming Lab Manual**

**PART A**

**1.     Write a Java program to find factorial of list of number reading input as command line argument.**

**Source Code**

public class Factorial

{

public static void main(String args[])

{

int[] arr = new int[10];

int fact;

if(args.length==0)

{

System.out.println("No Command line arguments");

return;

}

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

{

arr[i]=Integer.parseInt(args[i]);

}

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

{

fact=1;

while(arr[i]>0)

{

fact=fact\*arr[i];

arr[i]--;

}

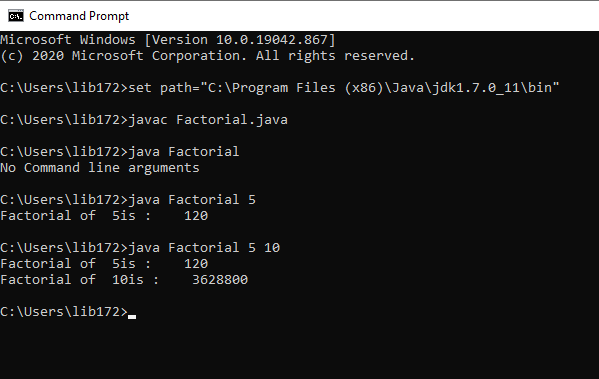
System.out.println("Factorial of "+ args[i]+"is : "+fact);

}

}

}

OUTPUT



**2.      Write a Java program to sort list of elements in ascending and descending order and show the exception handling.**

class Sorting

{

public static void main(String args[])

{

int a[] = new int[5];

try

{

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

a[i]=Integer.parseInt(args[i]);

System.out.println("Before Sorting\n");

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

System.out.println(" " + a[i]);

bubbleSort(a,5);

System.out.println("\n\n After Sorting\n");

System.out.println("\n\nAscending order \n");

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

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

System.out.println("\n\nDescending order \n");

for(int i=4;i>=0;i--)

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

}

catch(NumberFormatException e)

{

System.out.println("Enter only integers");

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("Enter only 5 integers");

}

}

private static void bubbleSort(int [] arr, int length)

{

int temp,i,j;

for(i=0;i<length-1;i++)

{

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

{

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

{

temp=arr[j];

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

arr[j+1]=temp;

}

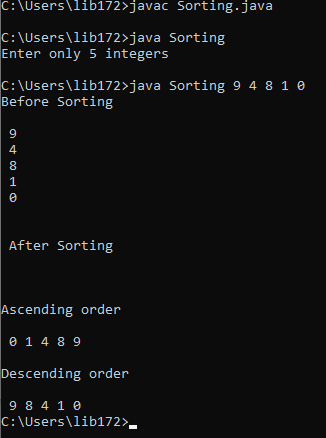
}

}

}

}

OUTPUT



**3     Write a Java program to implement all string operations.**

class StringOperation

{

public static void main(String args[])

{

String s1="Hello";

String s2="World";

System.out.println("The strings are "+s1+"and"+s2);

int len1=s1.length();

int len2=s2.length();

System.out.println("The length of "+s1+" is :"+len1);

System.out.println("The length of "+s2+" is :"+len2);

System.out.println("The concatenation of two strings = "+s1.concat(s2));

System.out.println("First character of "+s1+"is="+s1.charAt(0));

System.out.println("The uppercase of "+s1+"is="+s1.toUpperCase());

System.out.println("The lower case of "+s2+"is="+s2.toLowerCase());

System.out.println(" the letter e occurs at position"+s1.indexOf("e")+"in"+s1);

System.out.println("Substring of "+s1+"starting from index 2 and ending at 4 is =

"+s1.substring(2,4));

System.out.println("Replacing 'e' with 'o' in "+s1+"is ="+s1.replace('e','o'));

boolean check = s1.equals(s2);

if(check==false)

System.out.println(""+s1+" and "+s2+" are not same");

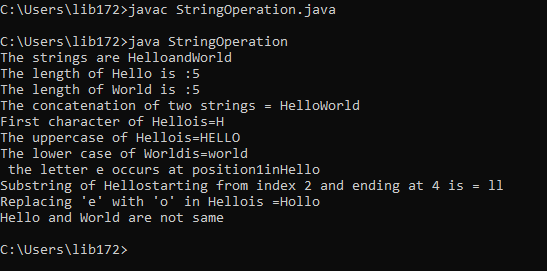
else

System.out.println("" + s1+" and " + s2+"are same");

}

}

OUTPUT



**4     Write a Java program to find area of geometrical figures using method.**

import java.io.\*;

class Area

{

public static double circleArea(double r)

{

return Math.PI\*r\*r;

}

public static double squareArea(double side)

{

return side\*side;

}

public static double rectArea(double width, double height)

{

return width\*height;

}

public static double triArea(double base, double height1)

{

return 0.5\*base\*height1;

}

public static String readLine()

{

String input=" ";

BufferedReader in=new BufferedReader(new InputStreamReader(System.in));

try

{

input = in.readLine();

}

catch(Exception e)

{

System.out.println("Error" + e);

}

return input;

}

public static void main(String args[])

{

System.out.println("Enter the radius");

Double radius=Double.parseDouble(readLine());

System.out.println("Area of circle = " + circleArea(radius));

System.out.println("Enter the side");

Double side=Double.parseDouble(readLine());

System.out.println("Area of square = "+squareArea(side));

System.out.println("Enter the Width");

Double width=Double.parseDouble(readLine());

System.out.println("Enter the height");

Double height=Double.parseDouble(readLine());

System.out.println("Area of Rectangle = " + rectArea(width,height));

System.out.println("Enter the Base");

Double base=Double.parseDouble(readLine());

System.out.println("Enter the Height");

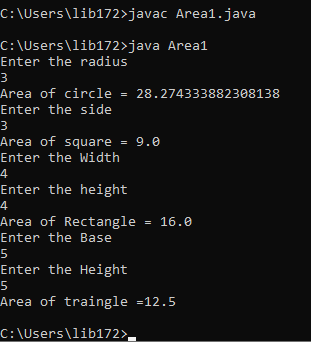
Double height1=Double.parseDouble(readLine());

System.out.println("Area of traingle ="+triArea(base,height1));

}

}

OUTPUT



**5.    Write a Java program to implement constructor overloading by passing different number of parameter of different types.**

public class Box

{

int length,breadth,height;

Box()

{

length=breadth=height=2;

System.out.println("Intialized with default constructor");

}

Box(int l, int b)

{

length=l;

breadth=b;

height=2;

System.out.println("Initialized with parameterized constructor having 2 params");

}

Box(int l, int b, int h)

{

length=l;

breadth=b;

height=h;

System.out.println("Initialized with parameterized constructor having 3 params");

}

public int getVolume()

{

return length\*breadth\*height;

}

public static void main(String args[])

{

Box box1 = new Box();

System.out.println("The volume of Box 1 is :"+ box1.getVolume());

Box box2 = new Box(10,20);

System.out.println("Volume of Box 2 is :" + box2.getVolume());

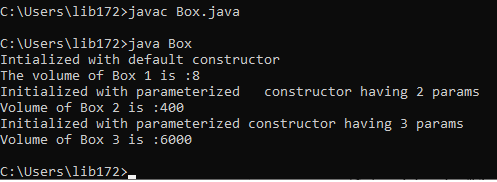
Box box3 = new Box(10,20,30);

System.out.println("Volume of Box 3 is :" + box3.getVolume());

}

}

OUTPUT



**6. Write a JAVA program to implement Multiple inheritance using interface**.

interface vehicleone{

int speed=90;

public void distance();

}

interface vehicletwo{

int distance=100;

public void speed();

}

class Vehicle implements vehicleone,vehicletwo{

public void distance()

{

int distance=speed\*100;

System.out.println("distance travelled is "+distance);

}

public void speed()

{

int speed=distance/100;

}

}

class MultipleInheritanceUsingInterface

{

public static void main(String args[])

{

Vehicle obj=new Vehicle();

System.out.println("Vehicle");

obj.distance();

obj.speed();

}

}

**Output:**

**Vehicle**

**distance travelled is 9000**

**7.  Write a Java program to implement Keyboard events.**

**Source code**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*<applet code="KeyBoardEvents" width=400 height=400></applet>\*/

public class KeyBoardEvents extends Applet implements KeyListener

{

String str="";

public void init()

{

addKeyListener(this);

requestFocus();

}

public void keyTyped(KeyEvent e)

{

str+=e.getKeyChar();

repaint(0);

}

public void keyPressed(KeyEvent e)

{

showStatus("Key Pressed");

}

public void keyReleased(KeyEvent e)

{

showStatus("Key Released");

}

public void paint(Graphics g)

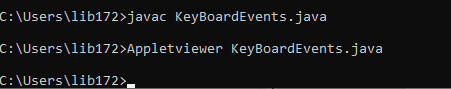
{

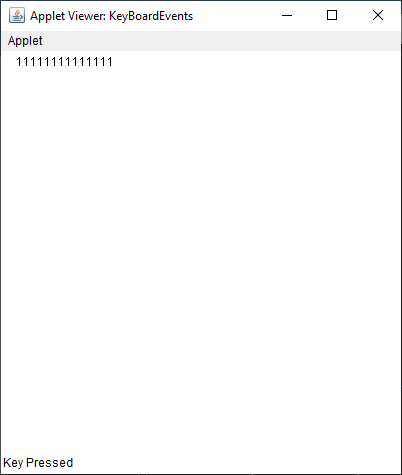
g.drawString(str,15,15);

}

}

OUTPUT





**8. Write a Java program to implement thread priorities.**

**Source code**

class A extends Thread

{

public void run()

{

System.out.println(" Thread A started");

for(int i=1;i<5;i++)

System.out.println(" Thread A : i = "+i);

System.out.println("Exit from Thread A");

}

}

class B extends Thread

{

public void run()

{

System.out.println(" Thread B started");

for(int i=1;i<5;i++)

System.out.println(" Thread B : i = "+i);

System.out.println("Exit from Thread B");

}

}

class C extends Thread

{

public void run()

{

System.out.println(" Thread C started");

for(int i=1;i<5;i++)

System.out.println(" Thread C : i = "+i);

System.out.println("Exit from Thread C");

}

}

class ThreadPriority

{

public static void main(String args[])

{

A threadA = new A();

B threadB = new B();

C threadC = new C();

threadA.setPriority(Thread.NORM\_PRIORITY);

threadB.setPriority(Thread.MAX\_PRIORITY);

threadC.setPriority(Thread.MIN\_PRIORITY);

System.out.println("Start Thread A");

threadA.start();

System.out.println("Start Thread B");

threadB.start();

System.out.println("Start Thread C");

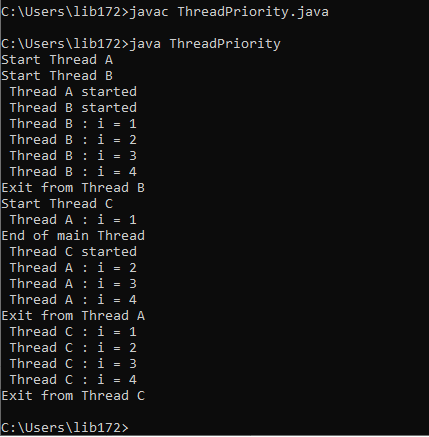
threadC.start();

System.out.println("End of main Thread");

}

}

OUTPUT:



**PART B**

**1. Write a Java program to display all prime numbers between two limits.**

class Prime

{

public static void main(String args[])

{

int i,j;

if(args.length<2)

{

System.out.println("No command line Argruments ");

return;

}

int num1=Integer.parseInt(args[0]);

int num2=Integer.parseInt(args[1]);

System.out.println("Prime number between"+num1+"and" +num2+" are:");

for(i=num1;i<=num2;i++)

{

for(j=2;j<i;j++)

{

int n=i%j;

if(n==0)

{

break;

}

}

if(i==j)

{

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

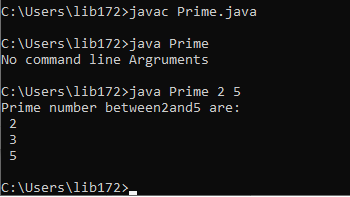
}

}

}

}

OUTPUT

****

## 2. Write Java Program to print the elements of an array in reverse order

**public** **class** ReverseArray {

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

        //Initialize array

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

        System.out.println("Original array: ");

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

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

        System.out.println();

        System.out.println("Array in reverse order: ");

        //Loop through the array in reverse order

**for** (**int** i = arr.length-1; i >= 0; i--) {

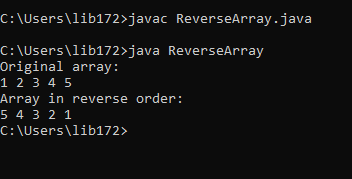
            System.out.print(arr[i] + " ");

        }

    }

}

OUTPUT

****

**3.   Write a Java program to implement Rhombus pattern reading the limit form user.**

import java.io.\*;

public class RhombusPattern

{

public static void main(String args[]) throws IOException

{

int i,j,limit;

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the limit");

limit=Integer.parseInt(br.readLine());

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

{

for(j=limit-i;j>0;j--)

System.out.print(" ");

for(j=1;j<=2\*i-1;j++)

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

System.out.println();

}

for(i=limit-1;i>=1;i--)

{

for(j=1;j<=limit-i;j++)

System.out.print(" ");

for(j=1;j<=2\*i-1;j++)

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

System.out.println();

}

}

}

OUTPUT

****

**4 Write a Java program to illustrate the concept of inheritance**

// base class

class Bicycle {

    // the Bicycle class has two fields

    public int gear;

    public int speed;

    // the Bicycle class has one constructor

    public Bicycle(int gear, int speed)

    {

        this.gear = gear;

        this.speed = speed;

    }

     // the Bicycle class has three methods

    public void applyBrake(int decrement)

    {

        speed -= decrement;

    }

     public void speedUp(int increment)

    {

        speed += increment;

    }

    public String toString()

    {

        return ("No of gears are " + gear + "\n"

                + "speed of bicycle is " + speed);

    }

}

 class MountainBike extends Bicycle {

    public int seatHeight;

    public MountainBike(int gear, int speed,  int startHeight)

    {

        super(gear, speed);

        seatHeight = startHeight;

    }

    public void setHeight(int newValue)

    {

        seatHeight = newValue;

    }

   public String toString()

    {

        return (super.toString() + "\nseat height is "

                + seatHeight);

    }

}

// driver class

public class Test {

    public static void main(String args[])

    {

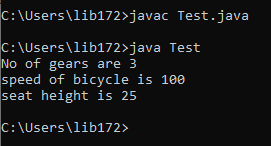
        MountainBike mb = new MountainBike(3, 100, 25);

        System.out.println(mb.toString());

    }

}

OUTPUT

****

5. **Java Program to find the number of the words in the given text file**

**import** java.io.BufferedReader;

**import** java.io.FileReader;

**public** **class** CountWordFile

{

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

        String line;

**int** count = 0;

        //Opens a file in read mode

        FileReader file = **new** FileReader("data.txt ");

        BufferedReader br = **new** BufferedReader(file);

        //Gets each line till end of file is reached

**while**((line = br.readLine()) != **null**) {

            //Splits each line into words

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

            //Counts each word

            count = count + words.length;

        }

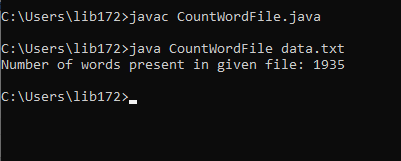
        System.out.println("Number of words present in given file: " + count);

        br.close();

    }

}

OUTPUT

****

**6.    Write a Java program to calculate bonus for different departments using method overriding.**

**Source code**

abstract class Department

{

double salary,bonus,totalsalary;

public abstract void calBonus(double salary);

public void displayTotalSalary(String dept)

{

System.out.println(dept+"\t"+salary+"\t\t"+bonus+"\t"+totalsalary);

}

}

class Accounts extends Department

{

public void calBonus(double sal)

{

salary = sal;

bonus = sal \* 0.2;

totalsalary=salary+bonus;

}

}

class Sales extends Department

{

public void calBonus(double sal)

{

salary = sal;

bonus = sal \* 0.3;

totalsalary=salary+bonus;

}

}

public class BonusCalculate

{

public static void main(String args[])

{

Department acc = new Accounts();

Department sales = new Sales();

acc.calBonus(10000);

sales.calBonus(20000);

System.out.println("Department \t Basic Salary \t Bonus \t Total Salary");

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

acc.displayTotalSalary("Accounts Dept");

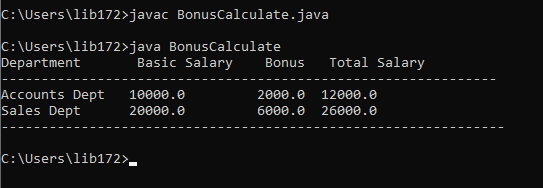
sales.displayTotalSalary("Sales Dept");

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

}

}

OUTPUT



7. **Write a program in java to enter the number through command line argument if first and second number is not entered it will generate the exception. Also divide the first number with second number and generate the arithmetic exception**.

class EXCEPTION

{

public static void main(String arr[])

{

try

{

if(arr.length<2)

throw(new Exception("two argument must be provided"));

int a= Integer.parseInt(arr[0]);

int b=Integer.parseInt(arr[1]);

if(b==0)

throw(new Exception("second argument should be non zero"));

int c=a/b;

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

}

catch(Exception e)

{

System.out.println(e);

}

}}

**Output:**

**Javac EXCEPTOIN.java**

**Java Exceprtion**

**java.lang.Exception: two argument must be provided**

**Java EXCEPTION 10 0**

**java.lang.Exception: second argument should be non zero**

**8.   Write a Java program to implement thread, applets and graphics by implementing animation of ball moving.**

**Source code :**

import java.awt.\*;

import java.applet.\*;

/\* <applet code="MovingBall.class" height=300 width=300></applet> \*/

public class MovingBall extends Applet implements Runnable

{

int x,y,dx,dy,w,h;

Thread t;

boolean flag;

public void init()

{

w=getWidth();

h=getHeight();

setBackground(Color.yellow);

x=100;

y=10;

dx=10;

dy=10;

}

public void start()

{

flag=true;

t=new Thread(this);

t.start();

}

public void paint(Graphics g)

{

g.setColor(Color.blue);

g.fillOval(x,y,50,50);

}

public void run()

{

while(flag)

{

if((x+dx<=0)||(x+dx>=w))

dx=-dx;

if((y+dy<=0)||(y+dy>=h))

dy=-dy;

x+=dx;

y+=dy;

repaint();

try

{

Thread.sleep(300);

}

catch(InterruptedException e)

{}

}

}

public void stop()

{

t=null;

flag=false;

}

}

OUTPUT

