Java lab programs

```
//Part A: 1. write a java program to find the factorial of given list of numbers
//reading input as command line argument
public class Factorial
{
	public static void main(String[] args) //"1","2","3"
	{
		int[] arr = new int[10];    //1 2 3 4 5
		int fact;
		if(args.length==0)
		{
			System.out.println(" no command line argument");
			return;
		}
		for(int i=0; i<args.length;i++)
		{
			arr[i]=Integer.parseInt(args[i]);
		}
		for(int i=0;i<args.length;i++)  //5
		{
			fact=1;
			while(arr[i]>0)  //3>0
			{
				fact=fact*arr[i]; //24
				arr[i]--;  //0
			}
			System.out.println("factorial of"+args[i]+"is:"+fact);
```

```
                }


        }
}
//part A 2
public class PrimeNumber
{
        public static void main(String[] args) {
                int i,j, count=0;
                if (args.length<2) {
                        System.out.println("no command line arument");
                        return;


                }
                int num1=Integer.parseInt(args[0]);
                int num2=Integer.parseInt(args[1]);
                System.out.println("PrimeNumbers 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);

                    }

                }


        }
}
```

3.Sorting

```
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("\n before Sorting \n");
                    for(int i=0;i<5;i++)
                        System.out.print(" "+a[i]);
```

```
                bubblesort(a,5);
                System.out.println("\n \n after Sorting \n");
                System.out.println("Ascending order \n");
                for(int i=0;i<5;i++)
                        System.out.print(" "+a[i]);
                        System.out.println("\nDescending\n");
                        for(int i=4;i>=0;i--)
                                System.out.print(" "+a[i]);
                }
                catch(NumberFormatException e)
                {
                        System.out.println("Enter only intergers");

                }
                catch(ArrayIndexOutOfBoundsException e)
                {
                        System.out.println("Enter only 5 intergers");
                }

}
private static void bubblesort(int[] arr, int len)
{
        int temp, i, j;
        for(i=0;i<len-1;i++)   //i=4    i<4
        {
                for (j=0;j<len-1-i ;j++ ) //j=0 j<1
                {
```

```
                                        if (arr[j]>arr[j+1])
                                        {
                                                temp=arr[j];
                                                arr[j]=arr[j+1];
                                                arr[j+1]=temp;  //

                                        }

                                }
                        }
                }
}
```

4. 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");

  }

}
```

5. Area of geometrical figures

```
import java.io.*;
import java.lang.Exception;
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 RreadLine()   //user defined method
{
  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(RreadLine());
   System.out.println("Area of circle = " + circleArea(radius));

   System.out.println("Enter the side");
   Double side=Double.parseDouble(RreadLine());
   System.out.println("Area of square = "+squareArea(side));

   System.out.println("Enter the Width");
   Double width=Double.parseDouble(RreadLine());
   System.out.println("Enter the height");
   Double height=Double.parseDouble(RreadLine());
   System.out.println("Area of Rectangle = " + rectArea(width,height));

   System.out.println("Enter the Base");
   Double base=Double.parseDouble(RreadLine());
   System.out.println("Enter the Height");
   Double height1=Double.parseDouble(RreadLine());
   System.out.println("Area of traingle ="+triArea(base,height1));
}
```

```
}
```

6. Constructor overloading

```
public class Box
{
  int length,breadth,height;  //global

  Box()
  {
     length=breadth=height=2;
     System.out.println("this is zero argument 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());

    }

}
```

```
//Part - A
//Program 7 : Write a java program to create student report using applet,
//            read the input using text boxes and display the output using buttons.

import java.awt.*;
import java.applet.*;
import java.awt.event.*;
/*<applet code=StudentReport height=400 width=400> </applet> */

public class StudentReport extends Applet implements ActionListener
{
 Label lblregno,lblname,lblcourse,lblsem,lblsub1,lblsub2,lblsub3;
 TextField txtregno,txtname,txtcourse,txtsem,txtsub1,txtsub2,txtsub3;
 Button cmdreport;
 String regno=" ",name=" ",course=" ",sem=" ",sub1=" ",sub2=" ",sub3=" ",total=" ",average=" ";
 int tot;
 float avg;
 public void init()
 {
  lblregno=new Label("Register Number : ");
  lblname=new Label("Student Name : ");
  lblcourse=new Label("Course : ");
  lblsem=new Label("Semester : ");
  lblsub1=new Label("Subject 1 : ");
  lblsub2=new Label("Subject 2 : ");
  lblsub3=new Label("Subject 3 : ");
```

```
  txtregno=new TextField(20);
  txtname=new TextField(20);
  txtcourse=new TextField(20);
  txtsem=new TextField(20);
  txtsub1=new TextField(3);
  txtsub2=new TextField(3);
  txtsub3=new TextField(3);

  cmdreport=new Button("View Report");

  add(lblregno);
  add(txtregno);
  add(lblname);
  add(txtname);
  add(lblcourse);
  add(txtcourse);
  add(lblsem);
  add(txtsem);
  add(lblsub1);
  add(txtsub1);
  add(lblsub2);
  add(txtsub2);
  add(lblsub3);
  add(txtsub3);
  add(cmdreport);
  cmdreport.addActionListener(this);
}
```

```
public void actionPerformed(ActionEvent e)
{
 if(e.getSource()==cmdreport)
  {
            regno=txtregno.getText();
            name=txtname.getText();
            course=txtcourse.getText();
            sem=txtsem.getText();
            sub1=txtsub1.getText();
            sub2=txtsub2.getText();
            sub3=txtsub3.getText();

            tot=Integer.parseInt(sub1)+Integer.parseInt(sub2)+Integer.parseInt(sub3);
            avg=(float)tot/3;

            regno="Register Number : "+regno;
       name="Name : "+name;
            course="Course : "+course;
            sem="Semester : "+sem;
            sub1="Subject1 : "+sub1;
            sub2="Subject2 : "+sub2;
            sub3="Subject3 : "+sub3;
            total="Total : "+String.valueOf(tot);
            average="Average : "+String.valueOf(avg);
            repaint();
 }
```

```
}
public void paint(Graphics g)
{
 g.drawString(regno,20,200);
 g.drawString(name,20,220);
 g.drawString(course,20,240);
 g.drawString(sem,20,260);
 g.drawString(sub1,20,280);
 g.drawString(sub2,20,300);
 g.drawString(sub3,20,320);
 g.drawString(total,20,340);
 g.drawString(average,20,360);
}
}
```

8. bonus of different department

```
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;

   }
}

 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("-----------------------------------------------------------");
    }
}
//PartA 9)
import java.awt.*;
import java.applet.*;

public class ballmoving 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=150;
                y=50;
                dx=20;
```

```
        dy=20;
}
public void start() //start method of applet
{
        flag=true;
        t=new Thread(this);
        t.start();    //thread
}
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() //applet
                {
                        t=null;
                        flag=false;
                }
        }
```

10. a. Keyboard event

```
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);
}
}
```

10. b. Mouse event

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

/*<applet code="mouse" width=400 height=300> </applet> */
public class mouse extends Applet implements MouseListener, MouseMotionListener
{
	String str="";
	public void init()
```

```
{
        addMouseListener(this);
        addMouseMotionListener(this);
}
public void paint(Graphics g)
{
        g.drawString(str,20,20);
}
public void mouseEntered(MouseEvent me)
{
        str="mouse button entered";
        repaint();
}
public void mousePressed(MouseEvent me)
{
        str="mouse pressed";
        repaint();
}
public void mouseClicked(MouseEvent me)
{
        str="mouse button clicked";
        repaint();
}

public void mouseReleased(MouseEvent me)
{
        str="mouse button Released";
```

```
            repaint();
        }

        public void mouseExited(MouseEvent me)
        {
            str="mouse button exited";
            repaint();
        }
        public void mouseMoved(MouseEvent me)
        {
            str="mouse button moved";
            repaint();
        }
        public void mouseDropped(MouseEvent me)
        {
            str="mouse button Dropped";
            repaint();
        }
        public void mouseDragged(MouseEvent me)
        {
            str="mouse button Dragged";
            repaint();
        }

}
```

11. mathematical functions

```
import java.lang.Math;
```

```
class Abs
{
	public static void main(String[] args) {
		System.out.println(" Sin(90): \t \t \t"+Math.sin(90));
		System.out.println(" Cos(90): \t \t \t"+Math.cos(90));
		System.out.println(" Tan(90): \t \t \t"+Math.tan(90));
		System.out.println(" abs(-1234.59): \t \t \t"+Math.abs(-1234.59));
		System.out.println(" Ceil(9.01): \t \t \t"+Math.ceil(9.01));
		System.out.println(" Floor(9.01): \t \t \t"+Math.floor(9.01));
		System.out.println(" Round(9.0): \t \t \t"+Math.round(9.01));
		System.out.println(" pow(4,2): \t \t \t"+Math.pow(4,2));
	}
}
```

12. Date and time

```
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.time.format.FormatStyle;

class DateTime
{
  public static void main(String[] args)
  {
   LocalDateTime current = LocalDateTime.now();
   DateTimeFormatter formatter = DateTimeFormatter.ofLocalizedDateTime(FormatStyle.MEDIUM);
   String Formatted  = current.format(formatter);
   System.out.println("current date is : "+ Formatted);
  }
```

```
}
```

13. sum of the digits

```
// wap to find the sum of the digits.
import java.util.Scanner;
class Digitsum
{
   public static void main(String[] args)
   {
                int m, n, sum=0;
          Scanner s= new Scanner(System.in);
                System.out.println("Enter the number");
                m=s.nextInt();
                while(m>0)
                {
                        n=m%10;   //n=3
                        sum=sum+n;  //sum=12
                        m=m/10;    //m=0
                }
        System.out.println("the sum of the digit is: "+ sum);
        }
}
```

14. multiplication table

```
import java.util.Scanner;
class Table
{
        public static void main(String[] args)
        {
```

```
                int n,c;
                System.out.println("Enter a number");
                Scanner s = new Scanner(System.in);
                n=s.nextInt();
                System.out.println("Multiplication table of "+n+" is");
                for (c=1;c<=10;c++)
                {
                        System.out.println(n+"*"+c+"="+(c*n));

                }

        }
}
```

15. vector operations

```
import java.util.*;
 class fruits {
      public static void main(String args[]) {

        Vector vec = new Vector(5);

        vec.add("Apple");
        vec.add("Banana");
        vec.add("papaya");
        vec.add("Pomogranate");

        System.out.println("Size of the vector is: "+vec.size());
```

```
        System.out.println("Vector elements are: "+vec);
        vec.addElement("watermelon");
        vec.addElement("pineapple");
        vec.addElement("Butterfruit");

        System.out.println("Size after addition: "+vec.size());

        System.out.println("Elements are: "+vec);
        vec.removeElement("papaya"); //vec.removeElementAt(3);
        System.out.println("after removing the papaya the vector is"+ vec);


        System.out.println("Butter fruit is present at the index " +vec.indexOf("Butterfruit"));

        System.out.println("The first fruit of the vector is = "+vec.firstElement());
        System.out.println("The last fruit of the vector is = "+vec.lastElement());

        vec.set(4, "grapes");
        System.out.println("after replacing the pineapple with grapes the vector is"+ vec);
    }
}
```

16. Thread priority

```
class ThreadA extends Thread
{
        public void run() //overriding the run()
        {
```

```
		for(int i=1;i<=5;i++)
		{
			System.out.println(Thread.currentThread().getName()+"="+i);
		}
		System.out.println("end of Thread one");
	}
}
class ThreadB extends Thread
{
	public void run() //overriding
	{
		for(int i=1;i<=5;i++)
		{
			System.out.println(Thread.currentThread().getName()+"="+i);
		}
		System.out.println("end of Thread two");
	}
}
class ThreadPriority
{
	public static void main(String[] args)
	{
		ThreadA ta = new ThreadA();
		Thread t1 = new Thread(ta,"Thread one");
		t1.setPriority(2);
		ThreadB tb = new ThreadB();
		Thread t2 = new Thread(tb,"Thread two");
```

```
                t2.setPriority(10);
                t1.start();
                t2.start();
        }
}
```

17. palindrome

```
class pal
{
  public static void main(String[] args)
  {
    String s1 = "bca";  //"mom"
    String s2 = "";  //trying to store bca as acb
    int len = s1.length();  //len=3
    for(int i=len-1;i>=0;i--)  //i=0
    {
      s2=s2+s1.charAt(i);  //s2="acb"
    }
    //System.out.println(s2);
    if(s1.equals(s2))
    {
      System.out.println("palindrome");

    }
    else
    {
      System.out.println(" not a palindrome");
    }
```

```
 }
}
```

18. Barchart

```
import java.applet.*;
import java.awt.*;
/*<applet code="barchart" width=500 height=500>
</applet>*/
public class barchart extends Applet
{
public void paint(Graphics g)
{
String year[]={"2006","2007","2008","2009","2010"};
int amount[]={120,140,135,150,170};
for(int i=0;i<5;i++)
{
g.drawString(year[i],20,i*50+30);     //drawString(what to print, x,y);
g.fillRect(50,i*50+10,amount[i],40); //fillRect(x,y,width,height)
}
}
}
```

19. Shapes

```
import java.applet.Applet;
import java.awt.*;
 /*<applet code=GraphicsDemo width=400 height=400>
</applet> */
public class GraphicsDemo extends Applet{
```

```
public void paint(Graphics g){
g.setColor(Color.red);
g.drawString("Welcome",50, 50);
g.drawLine(20,30,70,300);
g.drawRect(70,100,30,50);
g.fillRect(170,100,30,30);
g.drawOval(70,200,30,50);

g.setColor(Color.pink);
g.fillOval(170,200,30,30);
g.drawArc(90,150,30,30,30,270);
g.fillArc(270,150,30,30,0,180);

}
}
```

20. file operations

```
import java.io.*;
class Fileprogram
{
        public static void main(String[] args) throws IOException
        {
                File primitive = new File("bca.txt"); //constructor
                FileOutputStream fos = new FileOutputStream(primitive);//to write data into a fille
                DataOutputStream dos = new DataOutputStream(fos);
                dos.writeInt(1999);
                dos.writeFloat(37.85f);
                dos.writeChar('x');
```

```
                dos.close();
                fos.close();
                FileInputStream fis = new FileInputStream(primitive);//to read data
                DataInputStream dis = new DataInputStream(fis);
                System.out.println(dis.readInt());
                System.out.println(dis.readFloat());
                System.out.println(dis.readChar());
                dis.close();
                fis.close();
        }
}
```