Java Programming Lab

II-B.Tech, CSE I - Semester

2022-23

Prepared by

Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DNR COLLEGE OF ENGINEERING & TECHNOLOGY

BHIMAVARAM-534202

(Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada)

II Year – II Semester		0	0	3	1.5
JAVA PROGRAMMING LAB					

Course Objectives:
The aim of this lab is to

- Practice programming in the Java
- Gain knowledge of object-oriented paradigm in the Java programming language
- Learn use of Java in a variety of technologies and on different platforms

Course Outcomes:
By the end of the course student will be able to write java program for

- Evaluate default value of all primitive data type, Operations, Expressions, Control-flow, Strings
- Determine Class, Objects, Methods, Inheritance, Exception, Runtime Polymorphism, User defined Exception handling mechanism
- Illustrating simple inheritance, multi-level inheritance, Exception handling mechanism
- Construct Threads, Event Handling, implement packages, developing applets

Exercise - 1 (Basics)

a) Write a JAVA program to display default value of all primitive data type of JAVA

b) Write a java program that display the roots of a quadratic equation ax^2+bx=0. Calculate the discriminate D and basing on value of D, describe the nature of root.

c) Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers.

Exercise - 2 (Operations, Expressions, Control-flow, Strings)

a) Write a JAVA program to search for an element in a given list of elements using binary search mechanism.

b) Write a JAVA program to sort for an element in a given list of elements using bubble sort

c) Write a JAVA program to sort for an element in a given list of elements using merge sort.

d) Write a JAVA program using StringBuffer to delete, remove character.

Exercise - 3 (Class, Objects)

a) Write a JAVA program to implement class mechanism. Create a class, methods and invoke them inside main method.

b) Write a JAVA program to implement constructor.

Exercise - 4 (Methods)

a) Write a JAVA program to implement constructor overloading.

b) Write a JAVA program implement method overloading.

Exercise - 5 (Inheritance)

a) Write a JAVA program to implement Single Inheritance

b) Write a JAVA program to implement multi level Inheritance

c) Write a java program for abstract class to find areas of different shapes

Exercise - 6 (Inheritance - Continued)

a) Write a JAVA program give example for “super” keyword.

b) Write a JAVA program to implement Interface. What kind of Inheritance can be achieved?

Exercise - 7 (Exception)

a) Write a JAVA program that describes exception handling mechanism

b) Write a JAVA program Illustrating Multiple catch clauses.

Exercise – 8 (Runtime Polymorphism)

a) Write a JAVA program that implements Runtime polymorphism

b) Write a Case study on run time polymorphism, inheritance that implements in above problem

Exercise – 9 (User defined Exception)

a) Write a JAVA program for creation of Illustrating throw

b) Write a JAVA program for creation of Illustrating finally

c) Write a JAVA program for creation of Java Built-in Exceptions

d) d)Write a JAVA program for creation of User Defined Exception

Exercise – 10 (Threads)

a) Write a JAVA program that creates threads by extending Thread class .First thread display "Good Morning "every 1 sec, the second thread displays "Hello "every 2 seconds and the third display "Welcome" every 3 seconds ,(Repeat the same by implementing Runnable)

b) Write a program illustrating **isAlive** and **join ()**

c) Write a Program illustrating Daemon Threads.

Exercise - 11 (Threads continuity)

a) Write a JAVA program Producer Consumer Problem

b) Write a case study on thread Synchronization after solving the above producer consumer problem

Exercise – 12 (Packages)

a) Write a JAVA program illustrate class path

b) Write a case study on including in class path in your os environment of your package.

c) Write a JAVA program that import and use the defined your package in the previous Problem

Exercise - 13 (Applet)

a) Write a JAVA program to paint like paint brush in applet.

b) Write a JAVA program to display analog clock using Applet.

c) Write a JAVA program to create different shapes and fill colors using Applet.

Exercise - 14 (Event Handling)

a) Write a JAVA program that display the x and y position of the cursor movement using Mouse.

b) Write a JAVA program that identifies key-up key-down event user entering text in a Applet.

LAB INTRODUCTION

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
KAKINADA - 533 003, Andhra Pradesh, India
COMPUTER SCIENCE & ENGINEERING
(Applicable for batches admitted from 2021-2022)

II B. TECH II SEMESTER	L	T	P	C
	0	0	3	1.5
Java Programming Lab				

Exercise - 1 (Basics)

1 a) Write a JAVA program to display default value of all primitive data type of JAVA

Aim: A JAVA program to display default values of all primitive data types in JAVA.

Program:

```
class Exe1a
{
        static boolean b;
        static byte by;
        static short s;
        static int i;
        static long li;
        static char c;
        static float f;
        static double d;
        static String str;
        public static void main (String args[])
        {
                System.out.println("Boolean="+b);
                System.out.println("Byte="+by);
                System.out.println("Short="+s);
                System.out.println("Int="+i);
                System.out.println("Long="+li);
                System.out.println("Char="+c);
                System.out.println("Float="+f);
                System.out.println("Double="+d);
                System.out.println("String="+str);
        }
}
```

Output:

```
D:\DNRCET>javac Exe1a.java
D:\DNRCET>java Exe1a
Boolean=false
Byte=0
Short=0
Int=0
Long=0
```

```
Char=
Float=0.0
Double=0.0
String=null

D:\DNRCET>
```

1 b) Write a java program that display the roots of a quadratic equation ax2+bx=0. Calculate the discriminate D and basing on value of D, describe the nature of root.

Aim: A java program that display the roots of a quadratic equation ax2+bx=0. Calculate the discriminate D and basing on value of D, describe the nature of root.

Program:-

```
import java.io.*;
class Exe1b
{
        public static void main(String args[]) throws IOException
        {
                int a,b,c,d;
                double r1,r2;
                DataInputStream dis=new DataInputStream(System.in);
                System.out.println("Enter a,b and c values");
                a=Integer.parseInt(dis.readLine());
                b=Integer.parseInt(dis.readLine());
                c=Integer.parseInt(dis.readLine());
                d=b*b-4*a*c;
                if(d==0)
                {
                        r1=r2=(double)-b/(2*a);
                        System.out.println("Root1 is "+r1+" Root2 is "+r2);
                }
                else if(d>0)
                {
                        r1=(-b+Math.sqrt(d))/(2*a);
                        r2=(-b-Math.sqrt(d))/(2*a);
                        System.out.println("Root1 is "+r1+" Root2 is "+r2);
                }
                else
                {
                        System.out.println("Roots are imaginary");
                }
        }
}
```

Output:

```
D:\DNRCET>javac Exe1b.java
D:\DNRCET>java Exe1b
Enter a,b and c values
9
```

```
1
5
Roots are imaginary

D:\DNRCET>
```

1c) Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers.

Aim: Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers.

Program:-

```
import java.util.*;
class racedemo
{
public static void main(String[] args)
{
float s1,s2,s3,s4,s5,average;
Scanner s = new Scanner(System.in);
System.out.println("Enter speed of first racer:");
s1 = s.nextFloat();
System.out.println("Enter speed of second racer:");
s2 = s.nextFloat();
System.out.println("Enter speed of third racer:");
s3 = s.nextFloat();
System.out.println("Enter speed of fourth racer:");
s4 = s.nextFloat();
System.out.println("Enter speed of fifth racer:");
s5 = s.nextFloat(); average=(s1+s2+s3+s4+s5)/
5;
if(s1>average)
System.out.println("First racer is qualify racer:");
else if(s2>average)
System.out.println("Second racer is qualify racer:");
else if(s3>average)
System.out.println("Third racer is qualify
racer:"); else if(s4>average)
System.out.println("Fourth racer is qualify racer:");
else if(s5>average)
System.out.println("Fifth racer is qualify racer:");
}
}
```

Output:

```
Enter speed of first racer:
4.5
```

```
Enter speed of second racer:
6.7
Enter speed of third racer:
3.8
Enter speed of fourth racer:
5.3
Enter speed of fifth racer:
4.9
Second racer is qualify racer:
```

Exercise - 2 (Operations, Expressions, Control-flow, Strings)

2. a). Write a JAVA program to search for an element in a given list of elements using binary Search mechanism.

Aim: a).A JAVA program to search for an element in a given list of elements using binary Search mechanism.

Program:

```
import java.util.Scanner;
class A
{
        int binarysearch(int L[],int n,int k)
        {
                int low=0,high=n-1,mid;
                while(low<=high)
                {
                        mid=(low+high)/2;
                        if(L[mid]==k)
                        {
                                return mid;
                        }
                        else if(L[mid]>k)
                        {
                                high=mid-1;
                        }
                        else
                        {
                                low=mid+1;
                        }
                }
                return -1;
        }
}
class Exe2a
{
        public static void main(String args[])
        {
                int L[]=new int[20];
                int n,k,pos,i;
                A obj=new A();
                Scanner s=new Scanner(System.in);
                System.out.println("Enter n value");
                n=s.nextInt();
                System.out.println("Enter "+n+" elements");
                for(i=0;i<n;i++)
                {
                        L[i]=s.nextInt();
                }
                System.out.println("Enter searching element");
                k=s.nextInt();
                pos=obj.binarysearch(L,n,k);
                if(pos==-1)
                {
                        System.out.println("Searching element not found in the list");
                }
                else
                {
                        System.out.println("Element found at "+(pos+1)+" location");
                }
        }
}
```

Output:

```
D:\DNRCET>javac Exe2a.java

D:\DNRCET>java Exe2a
Enter n value
5
Enter 5 elements
1  2  3  4  5
Enter searching element
3
Element found at 3 location

D:\DNRCET>java Exe2a
Enter n value
4
Enter 4 elements
10  9  8  7
Enter searching element
5
Searching element not found in the list

D:\DNRCET>
```

2. b). Write a JAVA program to sort for an element in a given list of elements using bubble sort

Aim: b). A JAVA program to sort for an element in a given list of elements using bubble sort
Program:

```
import java.util.Scanner;

class Exe2b

{

        public static void main(String []args)

        {

                int n,t,i,j;

                Scanner s = new Scanner(System.in);

                System.out.println("How many numbers are to be sorted:");

                n = s.nextInt();

                int a[] = new int[n];

                System.out.println("Enter " + n + " integers");

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

                a[i] = s.nextInt();

                for (i = 0; i < ( n - 1 ); i++) {

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

                                if (a[j] > a[j+1]){

                                        t      = a[j];

                                        a[j]   = a[j+1];

                                        a[j+1] = t;

                                }

                        }

                }


        System.out.println("After after sorting: ");

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

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

```
                    }

          }

}
```

Output:-

```
D:\DNRCET>javac Exe2b.java

D:\DNRCET>java Exe2b

How many numbers are to be sorted:

6

Enter 6 integers

1         9         2         8         5         3

After after sorting:

      1      2      3      5      8      9

D:\DNRCET>
```

2.c) Write a JAVA program to sort for an element in a given list of elements using merge sort.

Aim:- A JAVA program to sort for an element in a given list of elements using merge sort.

Program:-

```
import java.util.*;
class Exe2c
{
          private int[] a;
          private int[] tempar;
          private int length;
          public static void main(String args[])
          {
                    int[] a = {6,1,3,4,2,5};
                    System.out.println("array before sorting is ");
                    for(int i=0;i < a.length;i++)
                    System.out.print("\t"+a[i]);
                    Exe2c obj = new Exe2c();
                    obj.sort(a);
                    System.out.println("\n array after sorting is ");
                    for(int i=0;i < a.length;i++)
                    {
                              System.out.print("\t"+a[i]);
                    }
          }
          void sort(int a[])
          {
                    this.a = a;
                    this.length = a.length;
```

```
                this.tempar = new int[length];
                mergesort(0, length - 1);
        }
        void mergesort(int low, int high)
        {
                if (low < high)
                {
                                int mid = (low + high) / 2;
                                mergesort(low, mid);
                                mergesort(mid + 1, high);
                                merge(low, mid, high);
                }
        }
        void merge(int low, int mid, int high)
        {
                int i;
                for (i = low; i <= high; i++)
                {
                        tempar[i] = a[i];
                }
                i = low;
                int j = mid + 1;
                int k = low;
                while (i <= mid && j <= high)
                {
                        if (tempar[i] <= tempar[j])
                        {
                                        a[k] = tempar[i];
                                        i++;
                          }
                        else
                        {
                                        a[k] = tempar[j];
                                        j++;
                        }
                        k++;
                }
                while (i <= mid)
                {
                                a[k] = tempar[i];
                                k++;
                                i++;
                }

        }
}
```

Output:-

```
D:\DNRCET>javac Exe2c.java

D:\DNRCET>java Exe2c
```

```
array before sorting is
      6      1      3      4      2      5
 array after sorting is
      1      2      3      4      5      6
D:\DNRCET>
```

2.d) Write a JAVA program using StringBuffer to delete, remove character.

Aim:- A JAVA program using StringBuffer to delete, remove character.

Program:-

```
class Exe2d
{
   public static void main(String arg[])
   {
      StringBuffer s = new StringBuffer("Kumar");
      System.out.println(s);
      s.delete(1, 3);
      System.out.println(s);
      s.deleteCharAt(0);
      System.out.println(s);
   }
}
```

Output:-

```
D:\DNRCET>javac Exe2d.java

D:\DNRCET>java Exe2d
Kumar
Kar
ar

D:\DNRCET>
```

Exercise - 3 (Class, Objects)

3.a). Write a JAVA program to implement class mechanism. – Create a class, methods and invoke them inside main method.

Aim:-A JAVA program to implement class mechanism. – Create a class, methods and invoke them inside main method.

Program:

```
public class Exe3a

   {

         private static void call()

         {
```

```
                        System.out.println("World");
            }
            public static void main(String[] args)
        {
                        System.out.print("Hello");
                call();
            }


    }
```

Output:

```
D:\DNRCET>javac Exe3a.java

D:\DNRCET>java Exe3a

HelloWorld

D:\DNRCET>
```

3.b). Write a JAVA program to implement constructor.

Aim:-A JAVA program to implement constructor.

Program:-

```
class Exe3b
    {
            Exe3b()
            {
                        System.out.println("Constructor called");
            }
            public static void main(String[] args)
        {
                        Exe3b obj= new Exe3b();
            }

    }
```

Output:-

```
D:\DNRCET>javac Exe3b.java

D:\DNRCET>java Exe3b
Constructor called

D:\DNRCET>
```

Exercise - 4 (Methods)

4. a). Write a JAVA program to implement constructor overloading.

Aim: A JAVA program to implement constructor overloading.

Program:

```
class Employee
{
        int age;
        String name;
        Employee()
        {
                age =20;
                name="Harish";
        }
        Employee(int age,String name)
        {
                this.age =age;
                this.name=name;
        }
        public void disp()
        {
                System.out.println("Name:"+name+"\t"+ "Age:"+age);
        }

}
class Exe4a
{
        public static void main(String args[])
        {
                Employee e1 = new Employee();
                Employee e2 = new Employee(15,"Sirish");
                e1.disp();
                e2.disp();
        }
}
```

Output:-

```
D:\DNRCET>javac Exe4a.java

D:\DNRCET>java Exe4a
Name:Harish     Age:20
Name:Sirish     Age:15
```

D:\DNRCET>

4.b). Write a JAVA program implement method overloading.

Aim:-A JAVA program implement method overloading.

Program:

```
class Exe4b
{
    int n1;
    int n2;
    Exe4b()
    {
        n1 = 10;
        n2 = 20;
    }
    void square()
   {
      System.out.println("The Square is " + n1 * n2);
    }
    void square(int p1)
    {
        n1 = p1;
        System.out.println("The Square is " + n1 * n2);
    }
    void square(int p1, int p2)
    {
        n1 = p1;
        n2 = p2;
        System.out.println("The Square is " + n1 * n2);
    }
    public static void main(String args[])
    {
        Exe4b obj = new Exe4b();
        obj.square(); //call non parameterise method
        obj.square(4);   //call method which has 1 argument
        obj.square(7,8);  //call method which has 2 argument
    }
}
```

Output:

```
D:\DNRCET>java Exe4b
The Square is 200
The Square is 80
The Square is 56

D:\DNRCET>
```

Exercise - 5 (Inheritance)

5. a). Write a JAVA program to implement Single Inheritance

Aim: A JAVA program to implement Single Inheritance

Program:

```
class A
{
        int a=10,b=5;
        void add()
        {
                System.out.println("Addition is= "+(a+b));
        }
}
class B extends A
{
        void sub()
        {
                System.out.println("Subtraction is= " +(a-b));
        }
}
class Exe5a
{
	public static void main(String args[])
	{
		B obj=new B();
		obj.add();
		obj.sub();
	}
}
```

Output:

```
D:\DNRCET>javac Exe5a.java

D:\DNRCET>java Exe5a
Addition is= 15
Subtraction is= 5

D:\DNRCET>
```

5 b) Write a JAVA program to implement multi level Inheritance

Aim:-b). a JAVA program to implement multi level Inheritance

Program:

```
class A
{
	int a=10,b=5;
	void add()
	{
		System.out.println("Addition is= "+(a+b));
	}
}
class B extends A
{
	void sub()
	{
		System.out.println("Subtraction is= " +(a-b));
	}
}
class C extends B
{
	void mul()
	{
		System.out.println("Multiplication is= "+(a*b));
	}
}
class Exe5b
{
	public static void main(String args[])
	{
		C obj = new C();
		obj.add();
		obj.sub();
		obj.mul();
	}
}
```

Output:

```
D:\DNRCET>java Exe5b

Addition is= 15

Subtraction is= 5

Multiplication is= 50

D:\DNRCET>
```

5. c). Write a java program for abstract class to find areas of different shapes

Aim:- a java program for abstract class to find areas of different shapes

Program:-

```
//Program for abstract class to find areas of different shapes
import java.util.Scanner;
abstract class Calcarea
{
	abstract void findTriangle(double b, double h);
	abstract void findRectangle(double l, double b);
	abstract void findSquare(double s);
	abstract void findCircle(double r);
}

class findArea extends Calcarea
{
	void findTriangle(double b, double h){
		double area = (b*h)/2;
		System.out.println("Area of Triangle: "+area);
	}

	void findRectangle(double l, double b){
		double area = l*b;
		System.out.println("Area of Rectangle: "+area);
	}
```

```
        void findSquare(double s){
                double area = s*s;
                System.out.println("Area of Square: "+area);
        }

        void findCircle(double r){
                double area = 3.14*r*r;
                System.out.println("Area of Circle: "+area);
        }
}

class Exe5c
{
        public static void main(String args[])
      {
                double l, b, h, r, s;
                findArea area = new findArea();
                Scanner obj = new Scanner(System.in);

                System.out.print("\nEnter Base & Vertical Height of Triangle: ");
                b = obj.nextDouble();
                h = obj.nextDouble();
                area.findTriangle(b, h);

                System.out.print("\nEnter Length & Breadth of Rectangle: ");
                l = obj.nextDouble();
                b = obj.nextDouble();
                area.findRectangle(l, b);

                System.out.print("\nEnter Side of a Square: ");
                s = obj.nextDouble();
                area.findSquare(s);

                System.out.print("\nEnter Radius of Circle: ");
                r = obj.nextDouble();
                area.findCircle(r);
        }
}
```

Output:-

```
D:\DNRCET>javac Exe5c.java

D:\DNRCET>java Exe5c

Enter Base & Vertical Height of Triangle: 5
3
Area of Triangle: 7.5

Enter Length & Breadth of Rectangle: 4
6
```

```
Area of Rectangle: 24.0

Enter Side of a Square: 1
Area of Square: 1.0

Enter Radius of Circle: 8
Area of Circle: 200.96

D:\DNRCET>
```

Exercise - 6 (Inheritance - Continued)
6 a). Write a JAVA program give example for “super” keyword.

Aim:- a). A JAVA program gives example for “super” keyword.

Program:

```
class A
{
	int a=10;
	A()
	{
		System.out.println("This is default constructor of A");
	}
	A(int a)
	{
		System.out.println("This is parameterised constructor of A "+a);
	}
	void display()
	{
		System.out.println("This is display method in A");
	}
}

class B extends A
{
	int a=20;
	B()
	{
	System.out.println("This is default constructor of B");
	}
	B(int a)
	{

		super(6);
		//this();
		System.out.println("This is parameterised constructor of B i.e addition
"+(a+this.a+super.a));
```

```
                super.display();
                this.display();
        }
        void display()
        {
                System.out.println("This is display method in B");
        }
}
class Exe6a
{
        public static void main(String args[])
        {
                B obj=new B();
                B obj2=new B(9);
        }
}
```

Output:

```
D:\DNRCET>javac Exe6a.java

D:\DNRCET>java Exe6a

This is default constructor of A

This is default constructor of B

This is parameterised constructor of A 6

This is parameterised constructor of B i.e addition 39

This is display method in A

This is display method in B


D:\DNRCET>
```

6 b). Write a JAVA program to implement Interface. What kind of Inheritance can be achieved?

Aim:- A JAVA program to implement Interface. What kind of Inheritance can be achieved?
Program:

```
interface add{

    int addition(int i,int j);

}

interface sub{

    int substraction(int i,int j);

}
```

```
class Exe6b implements add,sub
{
        public int addition(int i,int j){
                return (i+j);
        }
        public int substraction(int i,int j){
                return (i-j);
        }
        public static void main(String args[]){
                Exe6b obj=new Exe6b();
                int a=10,b=5;
                System.out.println("Addition is="+obj.addition(a,b));
                System.out.println("Substraction is="+obj.substraction(a,b));
        }
}
```

Output:

```
D:\DNRCET>javac Exe6b.java
D:\DNRCET>java Exe6b
Addition is=15
Substraction is=5
D:\DNRCET>
```

Note:- Using interface mechanism we achieved multiple inheritance.

Exercise - 7 (Exception)

7 a).Write a JAVA program that describes exception handling mechanism

Aim:-A JAVA program that describes exception handling mechanism

Program:

```
class Exe7a
{
        public static void main (String args[])throws Exception
        {
                int n=20,result;
                try
                {
                result=n/0;
                System.out.println("The result is"+result);
                }
                catch(ArithmeticException ex)
                {
                        System.out.println("Arithmetic exception occoured: ");
                }
                finally
                {
                     System.out.println("This is finally block");
                }
                result=n*n;
                System.out.println("Square of given number is "+result);
        }
}
```

Output:

```
D:\DNRCET>javac Exe7a.java

D:\DNRCET>java Exe7a

Arithmetic exception occurred:

This is finally block

Square of given number is 400

D:\DNRCET>
```

7 b).Write a JAVA program Illustrating Multiple catch clauses

Aim- b).A JAVA program Illustrating Multiple catch clauses

Program:

```
class Exe7b
{
   public static void main(String args[])
   {
       try
      {
       int a[]=new int[5];
       a[5]=1/0;
       //a[2]=a[20];
      }
      catch(ArithmeticException e)
      {
        System.out.println("Arithmetic Exception found");
      }
      catch(ArrayIndexOutOfBoundsException e)
      {
        System.out.println("Array Index Out of Bounds");
      }
      catch(Exception e)
      {
        System.out.println("Exception found");
      }
        System.out.println("End of the Program");
```

```
    }
}
```

Output:-

```
D:\DNRCET>javac Exe7b.java
D:\DNRCET>java Exe7b
Arithmetic Exception found
End of the Program

D:\DNRCET>
```

Exercise – 8 (Runtime Polymorphism)

8. a). Write a JAVA program that implements Runtime polymorphism

Aim:- A JAVA program that implements Runtime polymorphism

Program:

```
class Human{
  //Overridden method
  public void eat()
  {
    System.out.println("Human is eating");
  }
}
class Exe8a extends Human
{
  //Overriding method
  public void eat()
  {
    System.out.println("Boy is eating");
  }
  public static void main( String args[])
  {
    Exe8a obj = new Exe8a();
    //This will call the child class version of eat()
    obj.eat();
  }
}
```

Output:-

```
D:\DNRCET>javac Exe8a.java

D:\DNRCET>java Exe8a
Boy is eating

D:\DNRCET>
```

8. b). Write a Case study on run time polymorphism, inheritance that implements in above problem

Aim:- A Case study on run time polymorphism, inheritance that implements in above problem

Exercise – 9 (User defined Exception)
9 a). Write a JAVA program for creation of Illustrating throw

Aim:- a JAVA program for creation of Illustrating throw

Program:

```
class VoteException extends Exception
{
 VoteException(String s)
 {
  super(s);
 }

}

class Exe9a
{
   static void validate(int age) throws VoteException
   {
     if(age < 18)
         throw new VoteException("Candidate is not eligible to vote");
     else
         System.out.println("Welcome to Vote");
   }
   public static void main(String args[])
   {
      try
      {
        validate(13);
      }
      catch(Exception e)
     {
         System.out.println("Exception occured: "+e);
      }
      System.out.println("End of the program");
  }
}
```

Output:-

```
D:\DNRCET>javac Exe9a.java

D:\DNRCET>java Exe9a
Exception occured: VoteException: Candidate is not eligible to vote
End of the program
```

```
D:\DNRCET>javac Exe9a.java  //if age=19 then output

D:\DNRCET>java Exe9a
Welcome to Vote
End of the program

D:\DNRCET>
```

9 b). Write a JAVA program for creation of Illustrating finally

Aim:- A JAVA program for creation of Illustrating finally

Program:-

```
class Exe9b
{
   public static void main(String args[]){
     try{
        System.out.println("First statement of try block");
        int num=45/0;
        System.out.println(num);
     }
     catch(ArrayIndexOutOfBoundsException e)
     {
        System.out.println("ArrayIndexOutOfBoundsException");
     }
     catch(ArithmeticException e)
     {
        System.out.println("Arithmetic Exception");
     }
     finally
    {
        System.out.println("finally block");
     }
     System.out.println("Out of try-catch-finally block");
   }
}
```

Output:-

```
D:\DNRCET>javac Exe9b.java

D:\DNRCET>java Exe9b
First statement of try block
Arithmetic Exception
finally block
Out of try-catch-finally block

D:\DNRCET>
```

9 c). Write a JAVA program for creation of Java Built-in Exceptions

Aim:- A JAVA program for creation of Java Built-in Exceptions

Program:-

```
// Java program to demonstrate String Index Out Of Bounds Exception

class Exe9c
{
   public static void main(String args[])
    {
       try
      {
            String a = "This is built-in exception"; // length is 26
            char c = a.charAt(26); // accessing 27th element
            System.out.println(c);
      }
        catch (StringIndexOutOfBoundsException e)
        {
            System.out.println("String Index Out Of Bounds Exception");
        }
    }
}
```

Output:-

```
D:\DNRCET>javac Exe9c.java

D:\DNRCET>java Exe9c
String Index Out Of Bounds Exception

D:\DNRCET>
```

9 d). Write a JAVA program for creation of User Defined Exception

Aim:- A JAVA program for creation of User Defined Exception

Program:

```
//Program for creation of user defined exception

import java.util.Scanner;
 class OddNumberException extends Exception
    {
        OddNumberException(String s)
        {
            super(s);
        }
    }
    class Exe9d
    {
        public static void main(String[] args)
        {
            int num;
            Scanner S = new Scanner(System.in);
            System.out.print("Enter any number : ");
num = S.nextInt();
//num = Integer.parseInt(S.nextLine());
            try
            {
                if(num%2 != 0)
                    throw new OddNumberException("Odd number exception");
                else
                    System.out.println(" " + num + " is an even number");
            }
            catch(OddNumberException Ex)
            {
                System.out.println("Error : " + Ex.getMessage());
            }

            System.out.println("End of program");
        }
    }
```

Output:

```
D:\DNRCET>javac Exe9d.java

D:\DNRCET>java Exe9d
Enter any number : 456
 456 is an even number
End of program

D:\DNRCET>java Exe9d
Enter any number : 237
Error : Odd number exception
End of program

D:\DNRCET>
```

Exercise – 10 (Threads)

10 a.i). Write a JAVA program that creates threads by extending Thread class .First thread display “Good Morning “every 1 sec, the second thread displays “Hello “every 2 seconds and the third display “Welcome” every 3 seconds

Aim:- Write a JAVA program that creates threads by extending Thread class .First thread display “Good Morning “every 1 sec, the second thread displays “Hello “every 2 seconds and the third display “Welcome” every 3 seconds

Program:-

```
//Program for Multithreading by extending Thread Class

class ChildThread extends Thread
{
        Thread t;
        ChildThread(String name)
        {
                t = new Thread(this, name);
                t.start();
        }
        public void run()
        {
                for(int i=1;i<=5;i++)
                {
                try
                {
                        if(t.getName().equals("First Thread"))
                        {
                                Thread.sleep(1000);
                                System.out.println(t.getName()+": Good Morning");
                        }
                        else if(t.getName().equals("Second Thread"))
                        {
                                Thread.sleep(2000);
                                System.out.println(t.getName()+": Hello");
                        }
                        else
                        {
                                Thread.sleep(3000);
                                System.out.println(t.getName()+": Welcome");
                        }
                }
                catch(InterruptedException e)
                {
                        System.out.println(t.getName()+" is interrupted");
                }
                }
        }
}
class Exe10ai
{
        public static void main(String args[])

        {

                ChildThread one = new ChildThread("First Thread");

                ChildThread two = new ChildThread("Second Thread");

                ChildThread three = new ChildThread("Third Thread");

        }

}
```

Output:-

```
D:\DNRCET>javac Exe10ai.java

D:\DNRCET>java Exe10ai
First Thread: Good Morning
Second Thread: Hello
First Thread: Good Morning
Third Thread: Welcome
First Thread: Good Morning
Second Thread: Hello
First Thread: Good Morning
First Thread: Good Morning
Third Thread: Welcome
Second Thread: Hello
Second Thread: Hello
Third Thread: Welcome
Second Thread: Hello
Third Thread: Welcome
```

10 a.ii). Write a JAVA program that creates threads by implementing Runnable interface.First thread display “Good Morning “every 1 sec, the second thread displays “Hello “every 2 seconds and the third display “Welcome” every 3 seconds

Aim:-A JAVA program that creates threads by implementing Runnable interface.First thread display “Good Morning “every 1 sec, the second thread displays “Hello “every 2 seconds and the third display “Welcome” every 3 seconds

Program:-

//Program for Multithreading by implementing Runnable interface

```
class ChildThread implements Runnable
{
        Thread t;
        ChildThread(String name)
        {
                t = new Thread(this, name);
                t.start();
        }
        public void run()
        {
                for(int i=1;i<=5;i++)
                {
                try
                {
                        if(t.getName().equals("First Thread"))
                        {
                                Thread.sleep(1000);
                                System.out.println(t.getName()+": Good Morning");
                        }
                        else if(t.getName().equals("Second Thread"))
                        {
                                Thread.sleep(2000);
                                System.out.println(t.getName()+": Hello");
                        }
                        else
                        {
                                Thread.sleep(3000);
                                System.out.println(t.getName()+": Welcome");
                        }
                }
                catch(InterruptedException e)
                {
                        System.out.println(t.getName()+" is interrupted");
                }
                }
        }
}
class Exe10aii
{
        public static void main(String args[])
        {
                ChildThread one = new ChildThread("First Thread");
                ChildThread two = new ChildThread("Second Thread");
                ChildThread three = new ChildThread("Third Thread");
        }
}
```

Output:-

```
D:\DNRCET>javac Exe10aii.java

D:\DNRCET>java Exe10aii
First Thread: Good Morning
Second Thread: Hello
First Thread: Good Morning
Third Thread: Welcome
First Thread: Good Morning
Second Thread: Hello
First Thread: Good Morning
First Thread: Good Morning
Third Thread: Welcome
Second Thread: Hello
Second Thread: Hello
Third Thread: Welcome
Second Thread: Hello
Third Thread: Welcome
Third Thread: Welcome

D:\DNRCET>
```

10 b). Write a java program illustrating isAlive and join ()

Aim:-A JAVA program illustrating isAlive and join ().

Program:-

```
//import java.lang.*;
public class Exe10b implements Runnable {
   public void run() {
 Thread t = Thread.currentThread();
   // tests if this thread is alive
   System.out.println("status = " + t.isAlive());
   }
   public static void main(String args[]) throws Exception {
   Thread t = new Thread(new Exe10b());
   // this will call run() function
   t.start();
   // waits for this thread to die
   t.join();
   // tests if this thread is alive
   System.out.println("status = " + t.isAlive());
   }
}
```

Output:-

```
D:\DNRCET>javac Exe10b.java

D:\DNRCET>java Exe10b
status = true
status = false

D:\DNRCET>
```

10 c). Write a Program illustrating Daemon Threads.

Aim:-A java Program illustrating Daemon Threads

Program:-

```
// Program for illustrating Daemon Threads
class Daemonthread extends Thread
{
 public void run()
 {
  if(Thread.currentThread().isDaemon())
    System.out.println("Daemon Thread");
  else
     System.out.println("User Thread");
}
public static void main(String args[])
{
  Daemonthread t1= new Daemonthread();
  Daemonthread t2= new Daemonthread();
  Daemonthread t3= new Daemonthread();
  t1.setDaemon(true);
  t1.start();
  t2.start();
  t3.start();
}
}
```

Output:-

```
D:\DNRCET>javac Daemonthread.java

D:\DNRCET>java Daemonthread
Daemon Thread
User Thread
User Thread

D:\DNRCET>
```

Exercise - 11 (Threads continuity)

11 a).Write a JAVA program Producer Consumer Problem

Aim:-A JAVA program Producer Consumer Problem

Program:-

```
// Java program to implement solution of producer consumer problem

import java.util.LinkedList;
public class Exe11a
{
   public static void main(String[] args) throws InterruptedException
   {
      // Object of a class that has both produce()
      // and consume() methods
      final PC pc = new PC();
       // Create producer thread
      Thread t1 = new Thread(new Runnable()
      {
         public void run()
         {
            try
            {
               pc.produce();
            }
            catch(InterruptedException e)
            {
               e.printStackTrace();
            }
         }
      });
       // Create consumer thread
      Thread t2 = new Thread(new Runnable()
      {
         public void run()
         {
            try
            {
               pc.consume();
            }
            catch(InterruptedException e)
            {
                e.printStackTrace();
            }
        }
    });
    // Start both threads
    t1.start();
    t2.start();
    // t1 finishes before t2
    t1.join();
    t2.join();
}
 // This class has a list, producer (adds items to list and consumber (removes items).
public static class PC
{
    // Create a list shared by producer and consumer
    // Size of list is 2.
    LinkedList<Integer> list = new LinkedList<>();
    int capacity = 1;
    // Function called by producer thread
    public void produce() throws InterruptedException
    {
        int value = 0;
        while (true)
        {
            synchronized (this)
            {
                // producer thread waits while list is full
                while (list.size()==capacity)
                    wait();
                 System.out.println("Producer produced-" + value);
                 // to insert the jobs in the list
                list.add(value++);
                 // notifies the consumer thread that now it can start consuming
                notify();
                 // makes the working of program easier to  understand
                Thread.sleep(1000);
            }
        }
    }
    // Function called by consumer thread
    public void consume() throws InterruptedException
    {
        while (true)
        {
            synchronized (this)
            {
                // consumer thread waits while list is empty
```

```
                    while (list.size()==0)
                        wait();
                     //to retrive the first job in the list
                    int val = list.removeFirst();
                     System.out.println("Consumer consumed-" + val);
                     // Wake up producer thread
                    notify();
                   Thread.sleep(1000);
                }
            }
        }
    }
}
```

Output:-

```
D:\DNRCET>javac Exe11a.ja

D:\DNRCET>java Exe11a
Producer produced-0
Consumer consumed-0
Producer produced-1
Consumer consumed-1
Producer produced-2
Consumer consumed-2
Producer produced-3
Consumer consumed-3
Producer produced-4
Consumer consumed-4
Producer produced-5
Consumer consumed-5
Producer produced-6
Consumer consumed-6
Producer produced-7
Consumer consumed-7
Producer produced-8
Consumer consumed-8
Producer produced-9
Consumer consumed-9
Producer produced-10
Consumer consumed-10


D:\DNRCET>
```

11 b).Write a case study on thread Synchronization after solving the above producer consumer problem

Aim:- A case study on thread Synchronization after solving the above producer consumer problem

In computing, the producer–consumer problem (also known as the bounded-buffer problem) is a classic example of a multi-process synchronization problem. The problem describes two processes, the producer and the consumer, which share a common, fixed-size buffer used as a queue.

- The producer's job is to generate data, put it into the buffer, and start again.
- At the same time, the consumer is consuming the data (i.e. removing it from the buffer), one piece at a time.

Problem

To make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer.

Solution

The producer is to either go to sleep or discard data if the buffer is full. The next time the consumer removes an item from the buffer, it notifies the producer, who starts to fill the buffer again. In the same way, the consumer can go to sleep if it finds the buffer to be empty. The next time the producer puts data into the buffer, it wakes up the sleeping consumer. An inadequate solution could result in a deadlock where both processes are waiting to be awakened.

Implementation of Producer Consumer Class

- A **LinkedList list** – to store list of jobs in queue.
- **A Variable Capacity** – to check for if the list is full or not
- A mechanism to control the insertion and extraction from this list so that we do not insert into list if it is full or remove from it if it is empty.

Important Points

- In **PC class** (A class that has both produce and consume methods), a linked list of jobs and a capacity of the list is added to check that producer does not produce if the list is full.
- In **Producer class**, the value is initialized as 0.
  - Also, we have an infinite outer loop to insert values in the list. Inside this loop, we have a synchronized block so that only a producer or a consumer thread runs at a time.
  - An inner loop is there before adding the jobs to list that checks if the job list is full, the producer thread gives up the intrinsic lock on PC and goes on the waiting state.
  - If the list is empty, the control passes to below the loop and it adds a value in the list.
- In the **Consumer class**, we again have an infinite loop to extract a value from the list.
  - Inside, we also have an inner loop which checks if the list is empty.
  - If it is empty then we make the consumer thread give up the lock on PC and passes the control to producer thread for producing more jobs.
  - If the list is not empty, we go round the loop and removes an item from the list.
- In both the methods, we use notify at the end of all statements. The reason is simple, once you have something in list, you can have the consumer thread consume it, or if you have consumed something, you can have the producer produce something.

- sleep() at the end of both methods just make the output of program run in step wise manner and not display everything all at once so that you can see what actually is happening in the program.

Exercise – 12 (Packages)
12 a). Write a JAVA program illustrate class path

Aim:- A JAVA program illustrate class path

Program:-

```
package mypackage;
class Shape
{
 int d1,d2;
 Shape(int x,int y)
 {
   d1=x;
   d2=y;
 }
}
public class Rectangle extends Shape
{
 public Rectangle(int x,int y)
 {
   super(x,y);
 }
 public void area()
{
  System.out.println("Area:"+ (d1*d2));
}
}
import mypackage.*;
class ShapeDemo1
{
public static void main(String args[])
{
 Rectangle r = new Rectangle(20,30);
 r.area();
}
}
```

Output:-

```
C:\Users\admin>d:

D:\>cd DNRCET

D:\DNRCET>cd mypackage

D:\DNRCET\mypackage>javac Rectangle.java
```

```
D:\DNRCET\mypackage>cd..

D:\DNRCET>javac ShapeDemo1.java

D:\DNRCET>java ShapeDemo1
Area:600

D:\DNRCET>
```

b). Write a case study on including in class path in your os environment of your package.

Aim:-

Program:-

COPEING THE PATH OF JAVA:-
First of all go to MYCOMPUTER and go to the drive where the java is installed. In that go to

PROGRAM FILES and then double click on java folder.
In that we have observed there is a folder with name java jdk, double click on the java jdk folder and then go into the bin folder.
At this time we have to copy the path of the bin folder.

SETTING THE JAVA PATH:-
In order to set the path of the java in our system, first of all we need to open CONTROL PANEL in our system and go to SYSTEM SETTINGS .

In system settings we need to go into the ADVANCED SYSTEM SETTINGS settings.
In advanced system settings we just click on ENVIRONMENT VARIABLES option.
In USER VARIABLES click on NEW button and type the " path" at VARIABLE NAME.
We need to paste the previously copied path of the bin folder at the place of VARIABLE VALUE. Finally click on OK and then OK, then close the MYCOMPUTR window. Now the java path is set. We are ready to use java facilities in our computer.

c). Write a JAVA program that import and use the defined your package in the previous
Problem

Aim:- a JAVA program that import and use the defined your package in the previous
Problem

Program:-

```
import mypackage.Rectangle;
class ShapeDemo2
{
public static void main(String args[])
{
 Rectangle r = new Rectangle(20,30);
 r.area();
}
}
```

Output:-

```
D:\DNRCET>javac ShapeDemo2.java

D:\DNRCET>java ShapeDemo2
Area:600

D:\DNRCET>
```

```
class ShapeDemo3
{
public static void main(String args[])
{
mypackage.Rectangle r = new mypackage.Rectangle(20,30);
 r.area();
}
}
```

Output:-

```
D:\DNRCET>javac ShapeDemo3.java

D:\DNRCET>java ShapeDemo3
Area:600
```

Exercise - 13 (Applet)

13 a).Write a JAVA program to paint like paint brush in applet.

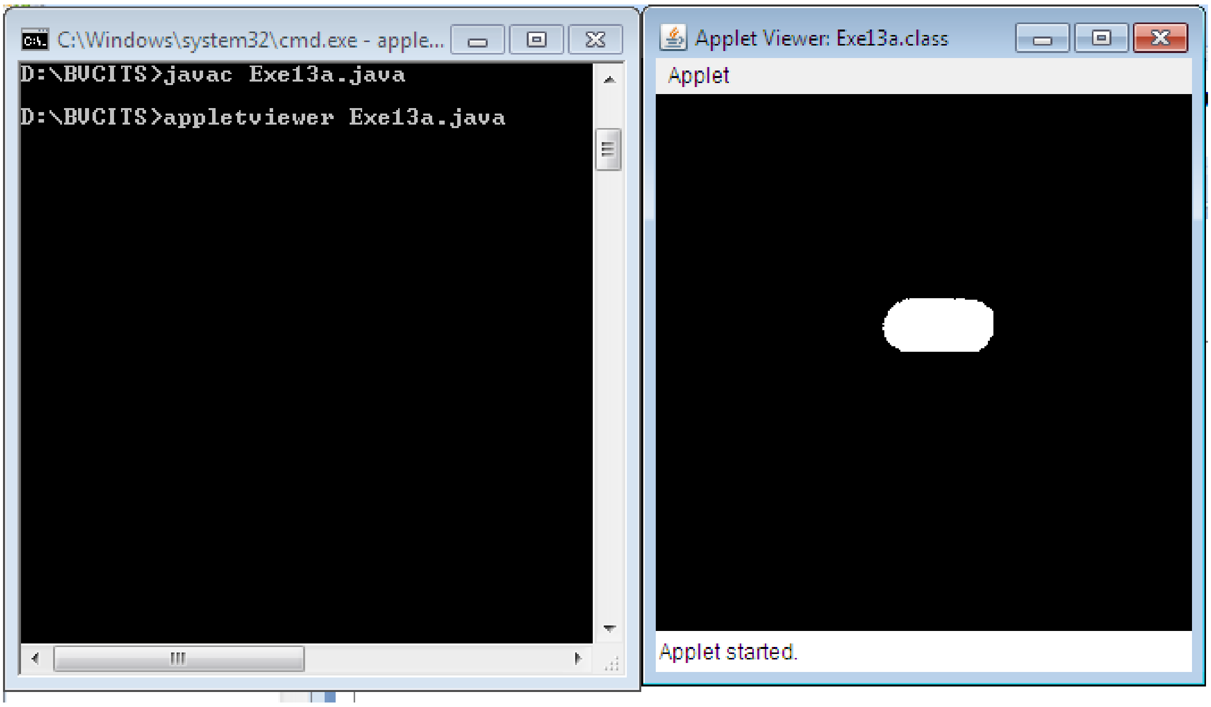
Aim:- A JAVA program to paint like paint brush in applet.

Program:-

/* JAVA program to paint like paint brush in applet*/

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class Exe13a extends Applet implements MouseMotionListener
{
public void init()
{
addMouseMotionListener(this);
setBackground(Color.black);
}
public void mouseDragged(MouseEvent me){
   Graphics g=getGraphics();
   g.setColor(Color.white);
   g.fillOval(me.getX(),me.getY(),5,5);
   }
   public void mouseMoved(MouseEvent me){}

   }
/*
<applet code="Exe13a.class" width="300" height="300">
</applet>
*/
```

Output:-



C:\Windows\system32\cmd.exe - apple...
Applet Viewer: Exe13a.class
Applet

```
D:\BVCITS>javac Exe13a.java

D:\BVCITS>appletviewer Exe13a.java
```

Applet started.

13 b) Write a JAVA program to display analog clock using Applet.

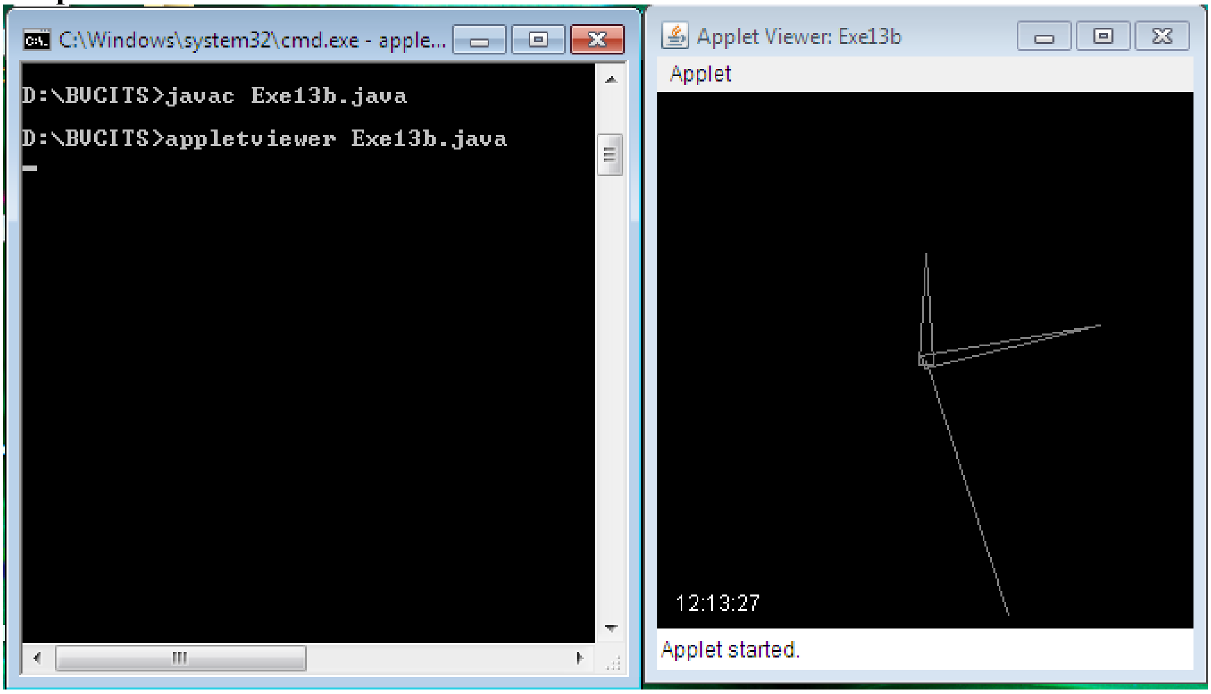
Aim:-A JAVA program to display analog clock using Applet.

Program:-

```
//JAVA program to display analog clock using Applet
import java.applet.*;
import java.awt.*;
import java.util.*;
import java.text.*;

public class Exe13b extends Applet implements Runnable {

  int width, height;
```

```
Thread t = null;
boolean threadSuspended;
int hours=0, minutes=0, seconds=0;
String timeString = "";

public void init() {
   width = getSize().width;
   height = getSize().height;
   setBackground( Color.black );
}

public void start() {
   if ( t == null ) {
      t = new Thread( this );
      t.setPriority( Thread.MIN_PRIORITY );
      threadSuspended = false;
      t.start();
   }
   else {
      if ( threadSuspended ) {
         threadSuspended = false;
         synchronized( this ) {
            notify();
         }
      }
   }
}

public void stop() {
   threadSuspended = true;
}

public void run() {
   try {
      while (true) {

         Calendar cal = Calendar.getInstance();
         hours = cal.get( Calendar.HOUR_OF_DAY );
         if ( hours > 12 ) hours -= 12;
         minutes = cal.get( Calendar.MINUTE );
         seconds = cal.get( Calendar.SECOND );

         SimpleDateFormat formatter
            = new SimpleDateFormat( "hh:mm:ss", Locale.getDefault() );
         Date date = cal.getTime();
         timeString = formatter.format( date );

         // Now the thread checks to see if it should suspend itself
         if ( threadSuspended ) {
            synchronized( this ) {
                while ( threadSuspended ) {
                   wait();
                }
             }
          }
          repaint();
          t.sleep( 1000 );  // interval specified in milliseconds
       }
    }
    catch (Exception e) { }
 }

 void drawHand( double angle, int radius, Graphics g ) {
    angle -= 0.5 * Math.PI;
    int x = (int)( radius*Math.cos(angle) );
    int y = (int)( radius*Math.sin(angle) );
    g.drawLine( width/2, height/2, width/2 + x, height/2 + y );
 }

 void drawWedge( double angle, int radius, Graphics g ) {
    angle -= 0.5 * Math.PI;
    int x = (int)( radius*Math.cos(angle) );
    int y = (int)( radius*Math.sin(angle) );
    angle += 2*Math.PI/3;
    int x2 = (int)( 5*Math.cos(angle) );
    int y2 = (int)( 5*Math.sin(angle) );
    angle += 2*Math.PI/3;
    int x3 = (int)( 5*Math.cos(angle) );
    int y3 = (int)( 5*Math.sin(angle) );
    g.drawLine( width/2+x2, height/2+y2, width/2 + x, height/2 + y );
    g.drawLine( width/2+x3, height/2+y3, width/2 + x, height/2 + y );
    g.drawLine( width/2+x2, height/2+y2, width/2 + x3, height/2 + y3 );
 }

 public void paint( Graphics g ) {
    g.setColor( Color.gray );
    drawWedge( 2*Math.PI * hours / 12, width/5, g );
    drawWedge( 2*Math.PI * minutes / 60, width/3, g );
    drawHand( 2*Math.PI * seconds / 60, width/2, g );
    g.setColor( Color.white );
    g.drawString( timeString, 10, height-10 );
 }
}
/*
<applet code="Exe13b" width="300" height="300">
</applet>
*/
```

Output:-



```
D:\BVCITS>javac Exe13b.java

D:\BVCITS>appletviewer Exe13b.java
```

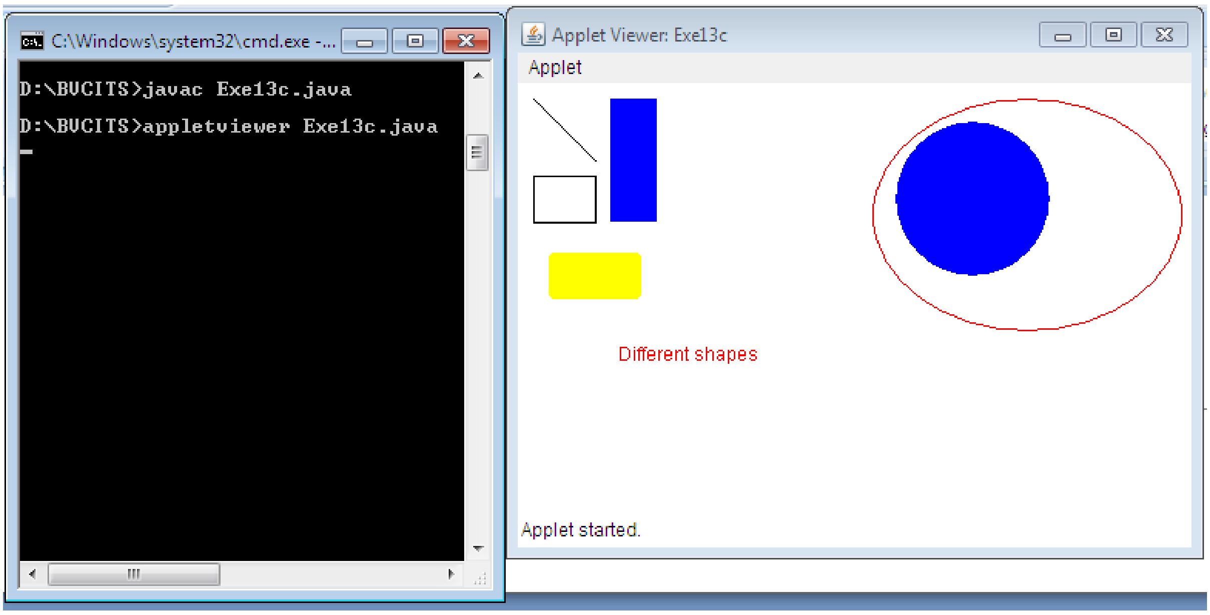
c). Write a JAVA program to create different shapes and fill colors using Applet.

Aim:- A JAVA program to create different shapes and fill colors using Applet

Program:-

```
import java.applet.*;
import java.awt.*;
public class Exe13c extends Applet
{
public void paint(Graphics g)
{
g.drawLine(10,10,50,50);
g.drawRect(10,60,40,30);
g.setColor(Color.blue);
g.fillRect(60,10,30,80);
g.setColor(Color.yellow);
g.fillRoundRect(20,110,60,30,5,5);
g.setColor(Color.red);
g.drawString("Different shapes ",65,180);
g.drawOval(230,10,200,150);
g.setColor(Color.blue);
g.fillOval(245,25,100,100);
}
}
/*<applet code="Exe13c"width=500 height=400></applet>*/
```

Output:-



```
D:\BVCITS>javac Exe13c.java

D:\BVCITS>appletviewer Exe13c.java
_
```

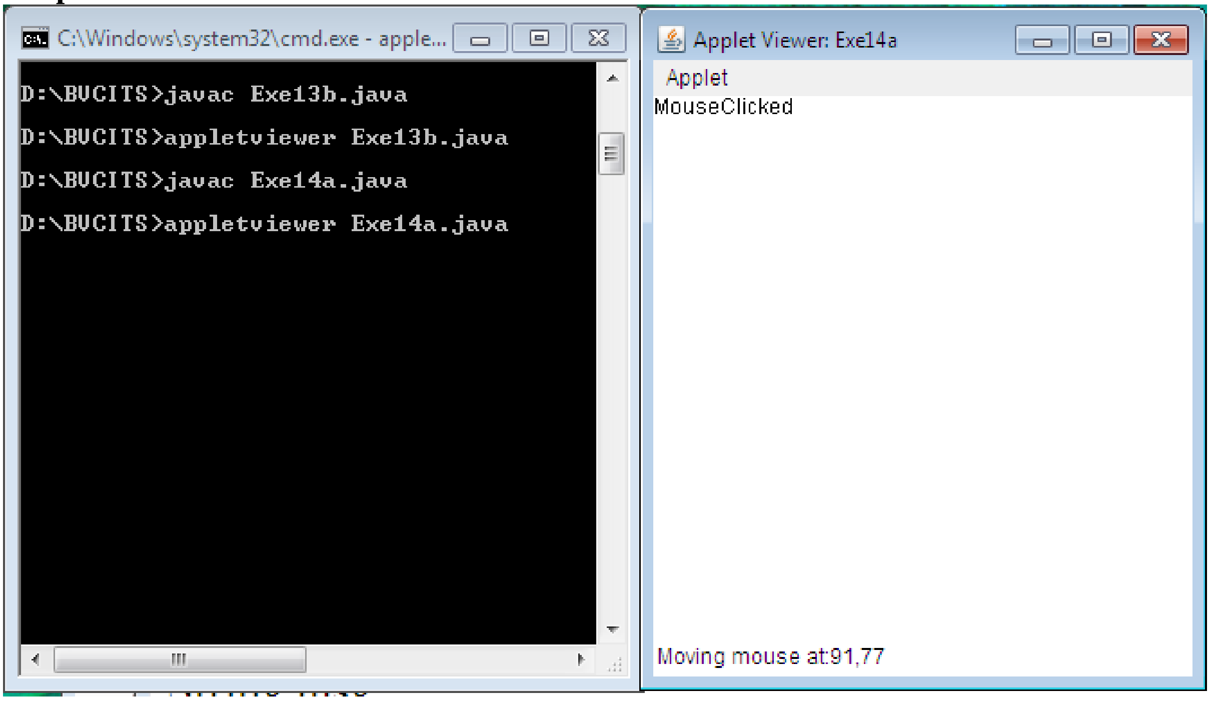
Exercise - 14 (Event Handling)
a).Write a JAVA program that display the x and y position of the cursor movement using Mouse.

Aim:- A JAVA program that display the x and y position of the cursor movement using Mouse.

Program:-

```
//java program that display the x and y position of the cursor movement using mouse
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="Exe14a" width=300 height=300>
</applet>*/
public class Exe14a extends Applet implements MouseListener,MouseMotionListener
{
String msg=" ";
int mousex=0,mousey=0;
public void init()
{
addMouseListener(this);
addMouseMotionListener(this);
}
public void mouseClicked(MouseEvent me)
{
mousex=0;
mousey=10;
msg="MouseClicked";
repaint();
}
public void mouseEntered(MouseEvent me)
{
mousex=0;
mousey=10;
msg="MouseEntered";
repaint();
}
public void mouseExited(MouseEvent me)
{
mousex=0;
mousey=10;
msg="MouseExited";
repaint();
}
public void mousePressed(MouseEvent me)
{
mousex=0;
mousey=10;
msg="down";
repaint();
}
public void mouseReleased(MouseEvent me)
{
mousex=me.getX();
mousey=me.getY();
msg="up";
repaint();
}
public void mouseDragged(MouseEvent me)
{
mousex=me.getX();
mousey=me.getY();
msg="*";
showStatus("Dragging mouse at:"+mousex+","+mousey);
repaint();
}
public void mouseMoved(MouseEvent me)
{
mousex=me.getX();
mousey=me.getY();
showStatus("Moving mouse at:"+mousex+","+mousey);
}
public void paint(Graphics g)
{
g.drawString(msg,mousex,mousey);
}
}
```

Output:-



```
D:\BVCITS>javac Exe13b.java

D:\BVCITS>appletviewer Exe13b.java

D:\BVCITS>javac Exe14a.java

D:\BVCITS>appletviewer Exe14a.java
```

```
MouseClicked

Moving mouse at:91,77
```

14 b).Write a JAVA program that identifies key-up key-down event user entering text in a Applet.

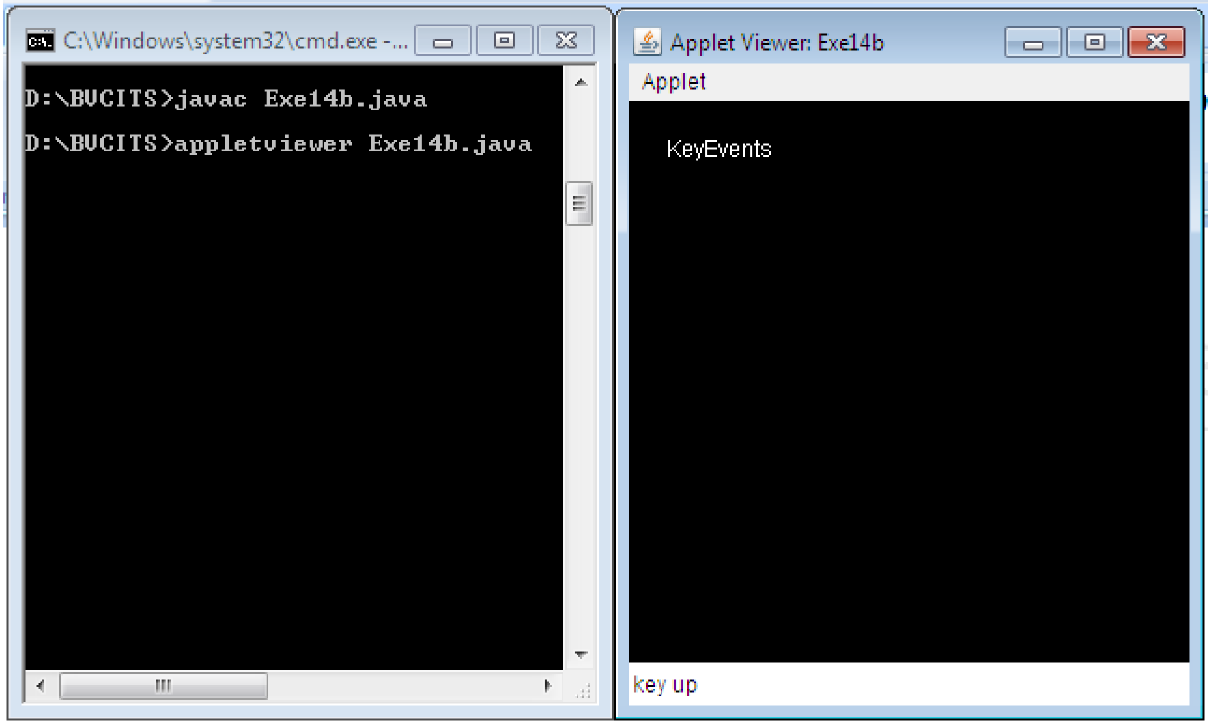
Aim:- A JAVA program that identifies key-up key-down event user entering text in a Applet.

Program:-

```
//java program that identifies key_up key_down event user entering text in a Applet
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="Exe14b"width=300 height=400>
</applet>
*/
public class Exe14b extends Applet implements KeyListener
{
int X=20,Y=30;
String msg="KeyEvents";
public void init()
{
addKeyListener(this);
requestFocus();
setBackground(Color.black);
setForeground(Color.white);
}
public void keyPressed(KeyEvent k)
{
showStatus("KeyDown");
int key=k.getKeyCode();
switch(key)
{
case KeyEvent.VK_UP:showStatus("Move to Up");
                break;
case KeyEvent.VK_DOWN:showStatus("Move to Down");
                 break;
case KeyEvent.VK_LEFT:showStatus("Move to Left");
                 break;
case KeyEvent.VK_RIGHT:showStatus("Move to right");
                 break;
}
repaint();
}
public void keyReleased(KeyEvent k)
{
showStatus("key up");
}
public void keyTyped(KeyEvent k)
{
```

```
msg+=k.getKeyChar();
repaint();
}
public void paint(Graphics g)
{
g.drawString(msg,X,Y);
}
}
```

Output:-



```
D:\BVCITS>javac Exe14b.java

D:\BVCITS>appletviewer Exe14b.java
```

```
KeyEvents
```

```
key up
```

Exercise - 15 (Swings)

a).Write a JAVA program to build a Calculator in Swings

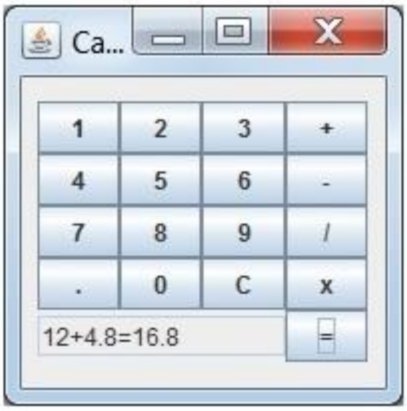
Aim:- A JAVA program to build a Calculator in Swings

Program:-

```
import java.awt.BorderLayout;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class Calculator implements ActionListener {
  private static JTextField inputBox;

  Calculator(){}
  public static void main(String[] args) {
    createWindow();
  }

  private static void createWindow() {
    JFrame frame = new JFrame("Calculator");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    createUI(frame);
    frame.setSize(200, 200);
    frame.setLocationRelativeTo(null);
    frame.setVisible(true);
  }

  private static void createUI(JFrame frame) {
    JPanel panel = new JPanel();
    Calculator calculator = new Calculator();
    GridBagLayout layout = new GridBagLayout();
    GridBagConstraints gbc = new GridBagConstraints();
    panel.setLayout(layout);

    inputBox = new JTextField(10);
    inputBox.setEditable(false);

    JButton button0 = new JButton("0");JButton button1 = new JButton("1");
    JButton button2 = new JButton("2");JButton button3 = new JButton("3");
    JButton button4 = new JButton("4");JButton button5 = new JButton("5");
    JButton button6 = new JButton("6");JButton button7 = new JButton("7");
    JButton button8 = new JButton("8");JButton button9 = new JButton("9");

    JButton buttonPlus = new JButton("+");JButton buttonMinus = new JButton("-");
    JButton buttonDivide = new JButton("/");JButton buttonMultiply = new JButton("x");
    JButton buttonClear = new JButton("C");JButton buttonDot = new JButton(".");
    JButton buttonEquals = new JButton("=");

    button1.addActionListener(calculator);button2.addActionListener(calculator);
    button3.addActionListener(calculator);button4.addActionListener(calculator);
    button5.addActionListener(calculator);button6.addActionListener(calculator);
    button7.addActionListener(calculator);button8.addActionListener(calculator);
    button9.addActionListener(calculator);button0.addActionListener(calculator);

    buttonPlus.addActionListener(calculator);buttonMinus.addActionListener(calculator);

buttonDivide.addActionListener(calculator);buttonMultiply.addActionListener(calculator);
    buttonClear.addActionListener(calculator);buttonDot.addActionListener(calculator);
    buttonEquals.addActionListener(calculator);

    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.gridx = 0; gbc.gridy = 0; panel.add(button1, gbc);
    gbc.gridx = 1; gbc.gridy = 0; panel.add(button2, gbc);
    gbc.gridx = 2; gbc.gridy = 0; panel.add(button3, gbc);
    gbc.gridx = 3; gbc.gridy = 0; panel.add(buttonPlus, gbc);
    gbc.gridx = 0; gbc.gridy = 1; panel.add(button4, gbc);
    gbc.gridx = 1; gbc.gridy = 1; panel.add(button5, gbc);
    gbc.gridx = 2; gbc.gridy = 1; panel.add(button6, gbc);
    gbc.gridx = 3; gbc.gridy = 1; panel.add(buttonMinus, gbc);
    gbc.gridx = 0; gbc.gridy = 2; panel.add(button7, gbc);
    gbc.gridx = 1; gbc.gridy = 2; panel.add(button8, gbc);
    gbc.gridx = 2; gbc.gridy = 2; panel.add(button9, gbc);
    gbc.gridx = 3; gbc.gridy = 2; panel.add(buttonDivide, gbc);
    gbc.gridx = 0; gbc.gridy = 3; panel.add(buttonDot, gbc);
    gbc.gridx = 1; gbc.gridy = 3; panel.add(button0, gbc);
    gbc.gridx = 2; gbc.gridy = 3; panel.add(buttonClear, gbc);
    gbc.gridx = 3; gbc.gridy = 3; panel.add(buttonMultiply, gbc);
    gbc.gridwidth = 3;

    gbc.gridx = 0; gbc.gridy = 4; panel.add(inputBox, gbc);
    gbc.gridx = 3; gbc.gridy = 4; panel.add(buttonEquals, gbc);
    frame.getContentPane().add(panel, BorderLayout.CENTER);
  }

  public void actionPerformed(ActionEvent e) {
    String command = e.getActionCommand();
    if (command.charAt(0) == 'C') {
      inputBox.setText("");
    }else if (command.charAt(0) == '=') {
      inputBox.setText(evaluate(inputBox.getText()));
```

```
        }else {
            inputBox.setText(inputBox.getText() + command);
        }
    }

    public static String evaluate(String expression) {
        char[] arr = expression.toCharArray();
        String operand1 = "";String operand2 = "";String operator = "";
        double result = 0;

        for (int i = 0; i < arr.length; i++) {
            if (arr[i] >= '0' && arr[i] <= '9' || arr[i] == '.') {
                if(operator.isEmpty()){
                    operand1 += arr[i];
                }else{
                    operand2 += arr[i];
                }
            }

            if(arr[i] == '+' || arr[i] == '-' || arr[i] == '/' || arr[i] == '*') {
                operator += arr[i];
            }
        }

        if (operator.equals("+"))
            result = (Double.parseDouble(operand1) + Double.parseDouble(operand2));
        else if (operator.equals("-"))
            result = (Double.parseDouble(operand1) - Double.parseDouble(operand2));
        else if (operator.equals("/"))
            result = (Double.parseDouble(operand1) / Double.parseDouble(operand2));
        else
            result = (Double.parseDouble(operand1) * Double.parseDouble(operand2));
        return operand1 + operator + operand2 + "=" +result;
    }
}
```

Output:-



12+4.8=16.8

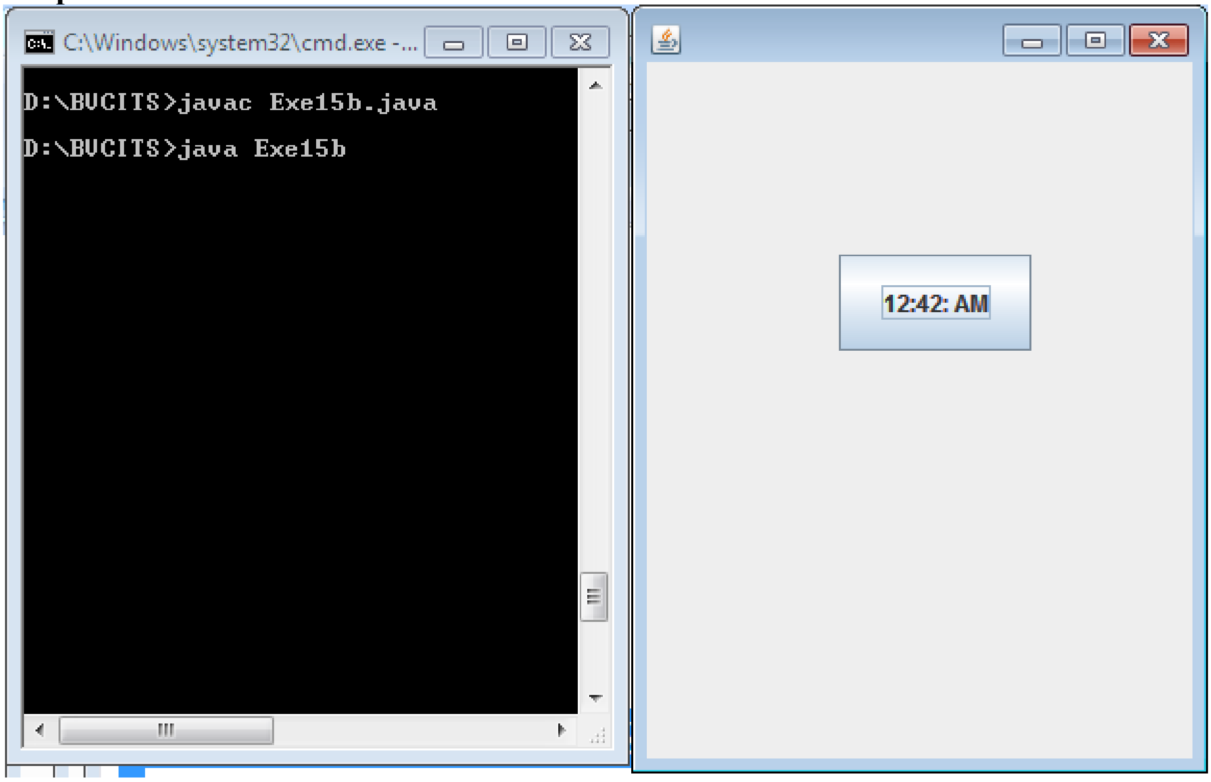
15 b). Write a JAVA program to display the digital watch in swing tutorial.

Aim:- A JAVA program to display the digital watch in swing tutorial.

Program:-

```
//java program to display the digital watch using swing
import javax.swing.*;
import java. awt.*;
import java.text.*;
import java.util.*;
public class Exe15b implements Runnable
{
JFrame f;
Thread t=null;
int hours=0,minutes=0,seconds=0;
String time="";
JButton b;
Exe15b()
{
f=new JFrame();
t=new Thread(this);
t.start();
b=new JButton();
b.setBounds(100,100,100,50);
f.add(b);
f.setSize(300,400);
f.setLayout(null);
f.setVisible(true);
f.setDefaultCloseOperation(f.EXIT_ON_CLOSE);
}
public void run()
{
try
{
while(true)
{
Calendar cal=Calendar.getInstance();
hours=cal.get(Calendar.HOUR_OF_DAY);
if(hours>12)hours-=12;
minutes=cal.get(Calendar.MINUTE);
seconds=cal.get(Calendar.SECOND);
SimpleDateFormat obj=new SimpleDateFormat("hh:mm: a");
Date date=cal.getTime();
time=obj.format(date);
printTime();
}
}
catch(Exception e)
{
}
}
public void printTime()
{
b.setText(time);
}
public static void main(String args[])
{
new Exe15b();
}
}
```

Output:-



```
C:\Windows\system32\cmd.exe -...

D:\BVCITS>javac Exe15b.java

D:\BVCITS>java Exe15b
```

Exercise – 16 (Swings - Continued)

a). Write a JAVA program that to create a single ball bouncing inside a JPanel.

Aim:- A JAVA program that to create a single ball bouncing inside a JPanel.

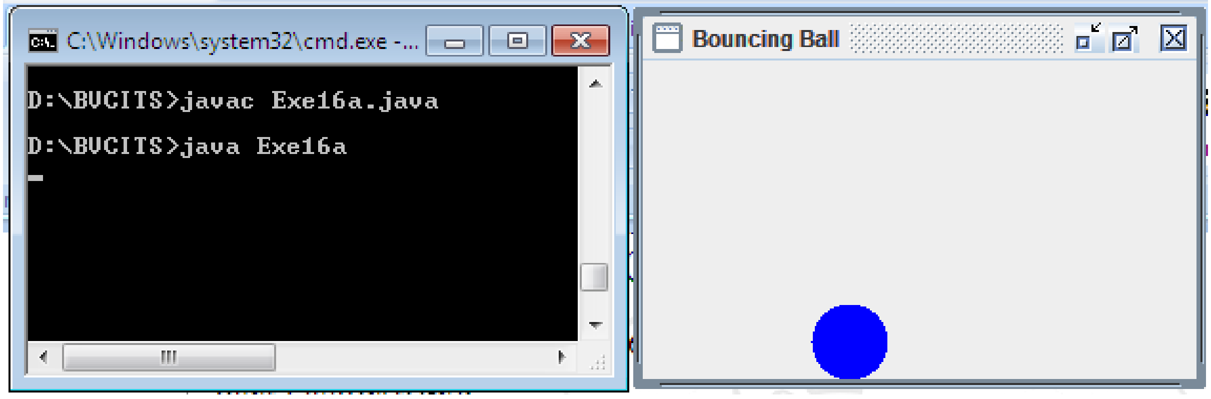
Program:-

```
//java program to create a single ball bouncing inside a JPanel
import java.awt.*;
import javax.swing.*;
public class Exe16a extends JPanel
{
int width;
int height;
float radius=20;
float diameter=radius*2;
float X=radius+50;
float Y=radius+20;
float dx=3;
float dy=3;
public Exe16a()
{
Thread thread=new Thread()
```

```
{
public void run()
{
while(true)
{
width=getWidth();
height=getHeight();
X=X+dx;
Y=Y+dy;
if(X-radius<0)
{
dx=-dx;
X=radius;
}
else if(X+radius>width)
{
dx=-dx;
X=width-radius;
}
if(Y-radius<0)
{
dy=-dy;
Y=radius;
}
else if(Y+radius>height)
{
dy=dy;
Y=height-radius;
}
repaint();
try
{
Thread.sleep(50);
}
catch(InterruptedException ex)
{
}
}
}
};
thread.start();
}
public void paintComponent(Graphics g)
{
super.paintComponent(g);
g.setColor(Color.BLUE);
g.fillOval((int)(X-radius),(int)(Y-radius),(int)diameter,(int)diameter);
}
public static void main(String args[])
{
```

```
JFrame.setDefaultLookAndFeelDecorated(true);
JFrame f=new JFrame("Bouncing Ball");
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setSize(300,200);
f.setContentPane(new Exe16a());
f.setVisible(true);
}
}
```

Output:-



```
D:\BUCITS>javac Exe16a.java

D:\BUCITS>java Exe16a
_
```

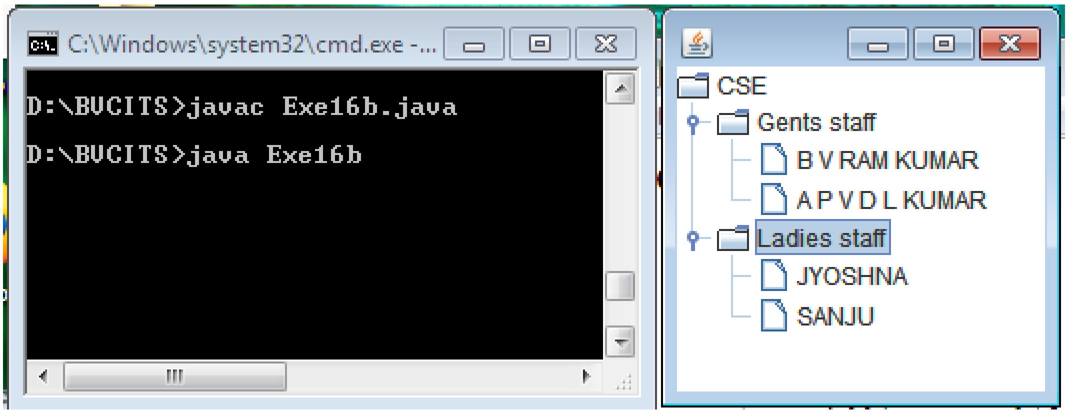
16 b). Write a JAVA program JTree as displaying a real tree upside down

Aim:-A JAVA program JTree as displaying a real tree upside down

Program:-

```
//java program for displaying a real tree using Jtree
import javax.swing.*;
import javax.swing.tree.DefaultMutableTreeNode;
public class Exe16b
{
JFrame f;
Exe16b()
{
f=new JFrame();
DefaultMutableTreeNode cse=new DefaultMutableTreeNode("CSE");
DefaultMutableTreeNode gents=new DefaultMutableTreeNode("Gents staff");
DefaultMutableTreeNode ladies=new DefaultMutableTreeNode("Ladies staff");
cse.add(gents);
cse.add(ladies);
DefaultMutableTreeNode s1=new DefaultMutableTreeNode("B V RAM KUMAR");
DefaultMutableTreeNode s2=new DefaultMutableTreeNode("A P V D L KUMAR");
gents.add(s1);
gents.add(s2);
DefaultMutableTreeNode l1=new DefaultMutableTreeNode("JYOSHNA");
DefaultMutableTreeNode l2=new DefaultMutableTreeNode("SANJU");
ladies.add(l1);
ladies.add(l2);
JTree jt=new JTree(cse);
f.add(jt);
f.setSize(200,200);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public static void main(String args[])
{
new Exe16b();
}
}
```

Output:-



```
D:\BUCITS>javac Exe16b.java

D:\BUCITS>java Exe16b
```

```
CSE
  Gents staff
    B V RAM KUMAR
    A P V D L KUMAR
  Ladies staff
    JYOSHNA
    SANJU
```

ADD ON EXPEREMENTS

1.
2.

MODEL PRACTICAL END EXAMINATION QUESTIONS

II-II CSE – JAVA PROGRAMMING LAB EXTERNAL QUESTION PAPER

INTERNAL HOD

NAME OF THE EXIMINERS:

i.	**Name of the Examiner**	
ii.	**Name of the Physical laboratory:**	JAVA LAB
iii.	Room No	LAB-1 & LAB-2
iv.	Name of the faculty in-charge (for curriculum lab- CL)	
iv.	Class with section	II B.Tech (CSE) – II Semester
v.	AY and Semester	2021-22