II B.Tech. I SEM Regulation: R23

Laboratory Manual

For the course of

Object Oriented Programming Through Java Lab

Branch: CSE /CSM/CAI/CSD/AIML

VIGNAN'S LARA
INSTITUTE OF TECHNOLOGY & SCIENCE
(AUTONOMOUS)
Approved by AICTE New Delhi & Affiliated to JNTUK Kakinada
Accredited by **NAAC 'A+'** and **NBA** | **ISO 9001 : 2015**
Vadlamudi - 522 213, Guntur District

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

II Year – I Semester	**L**	**T**	**P**	**C**
	0	**0**	**3**	**2**

OBJECT ORIENTED THROUGH JAVA LAB

Exercise - 1

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.

Exercise - 2

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 using String Buffer to delete, remove character.

Exercise - 3

a). Write a JAVA program to implement class mechanism. – Create a class, methods and invoke them inside main method.
b). Write a JAVA program implement method overloading.
c).Write a JAVA program to implement constructor.
d). Write a JAVA program to implement constructor overloading.

Exercise - 4

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 - 5

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?
c). Write a JAVA program that implements Runtime polymorphism

Exercise - 6

a).Write a JAVA program that describes exception handling mechanism
b).Write a JAVA program Illustrating Multiple catch clauses
c). Write a JAVA program for creation of Java Built-in Exceptions
d).Write a JAVA program for creation of User Defined Exception

Exercise – 7

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.
d).Write a JAVA program Producer Consumer Problem

Exercise – 8

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.

Exercise - 9 (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 - 10 (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.

Exercise - 11 (Swings)

a).Write a JAVA programto build a Calculator in Swings

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

Exercise – 12 (Swings - Continued)

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

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

Exercise - 1

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

Program:

```
class  DefaultDemo
{
        static byte b;
        static short s;
        static int i;
        static long l;
        static float f;
        static double d;
        static char c;
        static boolean bl;
        public static void main(String[] args)
        {
                System.out.println("The default values of primitive data types are:");
                System.out.println("Byte :"+b);
                System.out.println("Short :"+s);
                System.out.println("Int :"+i);
                System.out.println("Long :"+l);
                System.out.println("Float :"+f);
                System.out.println("Double :"+d);
                System.out.println("Char :"+c);
                System.out.println("Boolean :"+bl);
        }
}
```

Output:

```
The default values of primitive data types are:
Byte :0
Short :0
Int :0
Long :0
Float :0.0
Double :0.0
Char :
Boolean :false
```

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

Aim: To 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.

Program:

```
import java.util.*;
class QuadraticDemo
{
        public static void main(String[] args)
        {
                int a, b, c;
                double r1, r2, D;
                Scanner s = new Scanner(System.in);
                System.out.println("Given quadratic equation:ax^2 + bx + c");
                System.out.print("Enter a:");
                a = s.nextInt();
                System.out.print("Enter b:");
                b = s.nextInt();
                System.out.print("Enter c:");
                c = s.nextInt();
                D = b * b - 4 * a * c;
                if(D > 0)
                {
                           System.out.println("Roots are real and unequal");
                           r1 = ( - b + Math.sqrt(D))/(2*a);
                           r2 = (-b - Math.sqrt(D))/(2*a);
                           System.out.println("First root is:"+r1);
                           System.out.println("Second root is:"+r2);
               }
               else if(D == 0)
               {
                          System.out.println("Roots are real and equal");
                          r1 = (-b+Math.sqrt(D))/(2*a);
                          System.out.println("Root:"+r1);
               }
               else
               {
                         System.out.println("Roots are imaginary");
               }
        }
}
```

Output:

```
Given quadratic equation:ax^2 + bx + c
Enter a:2
Enter b:3
Enter c:1
Roots are real and unequal
First root is:-0.5
Second root is:-1.0
```

Exercise - 2

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

Aim: To write a JAVA program to search for an element in a given list of elements using binary search mechanism

Program:

```
import java.util.Scanner;
class BinarySearchDemo
{
        public static void main(String args[])
        {
                int n, i, num,first, last, middle;
                int a[ ]=new int[20];
                Scanner s = new Scanner(System.in);
                System.out.println("Enter total number of elements:");
                n = s.nextInt();
                System.out.println("Enter elements in sorted order:");
                for (i = 0; i < n; i++)
                a[i] = s.nextInt();
                System.out.println("Enter the search value:");
                num = s.nextInt();
                first = 0;
                last = n - 1;
                middle = (first + last)/2;
                while( first <= last )
                {
                        if ( a[middle] < num )
                        first = middle + 1;
                        else if ( a[middle] == num )
                        {
                                System.out.println("number found");
                                break;
                        }
                        else
                        {
                                last = middle - 1;
                        }
                        middle = (first + last)/2;
                 }
                 if ( first > last )
                 System.out.println( " Number is not found");
         }
}
```

Output:

```
Enter total number of elements:
5
Enter elements:
2 4 6 8 9
Enter the search value:
8
number found
```

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

Aim: To write a JAVA program to sort for an element in a given list of elements using bubble sort

Program:

```
import java.util.Scanner;
class BubbleDemo
{
        public static void main(String args[])
        {
                int n, i,j, temp;
                int a[ ]=new int[20];
                Scanner s = new Scanner(System.in);
                System.out.println("Enter total number of elements:");
                n = s.nextInt();
                System.out.println("Enter elements:");
                for (i = 0; i < n; i++)
                a[i] = s.nextInt();
                for(i=0;i<n;i++)
                {
                        for(j=0;j<n-1;j++)
                        {
                                if(a[j]>a[j+1])
                                {
                                        temp=a[j];
                                        a[j]=a[j+1];
                                        a[j+1]=temp;
                                }
                       }
               }
               System.out.println("The sorted elements are:");
               for(i=0;i<n;i++)
               System.out.print("\t"+a[i]);
         }
}
```

Output:

```
Enter total number of elements:
10
Enter elements:
3 2 5 7 6 8 9 1 4 0
The sorted elements are:
 0      1      2      3      4      5      6      7      8      9
```

2 c) Write a JAVA program using String Buffer to delete, remove character.

Aim: To write a JAVA program using StringBuffer to delete, remove character

Program:

```
class StringBufferDemo
{
	public static void main(String[] args)
	{
		StringBuffer sb1 = new StringBuffer("Hello World");
		sb1.delete(0,6);
		System.out.println(sb1);
		StringBuffer sb2 = new StringBuffer("Some Content");
		System.out.println(sb2);
		sb2.delete(0, sb2.length());
		System.out.println(sb2);
		StringBuffer sb3 = new StringBuffer("Hello World");
		sb3.deleteCharAt(0);
		System.out.println(sb3);
	}
}
```

Output:

```
World
Some Content

ello World
```

Exercise - 3

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

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

Programs:

1. **no return type and without parameter-list:**

```
class A
{
        int l=10,b=20;
        void display()
        {
                System.out.println(l);
                System.out.println(b);
        }
}
class methoddemo
{
        public static void main(String args[])
        {
                A a1=new A();
                a1.display();
        }
}
```

Output:

```
10
20
```

2. return type and without parameter-list

```
class A
{
        int l=10,b=20;
        int area()
        {
                return l*b;
        }
}
class methoddemo
{
         public static void main(String args[])
         {
                A a1=new A();
                int r=a1.area();
                System.out.println("The area is: "+r);
         }
}
```

Output:

```
The area is:200
```

3. return type and with parameter-list:

```
class A
{
        int area(int l,int b)
        {
              return l*b;
        }
}
class methoddemo
{
        public static void main(String args[])
        {
              A a1=new A();
              int r=a1.area(10,20);
              System.out.println(“The area is:”+r);
        }
}
```

Output:

```
The area is:200
```

3 b)Write a JAVA program implement method overloading.

Aim: To write a JAVA program implement method overloading

Program:

```
class A
{
         int l=10,b=20;
        int area()
        {
                 return l*b;
        }
        int area(int l,int b)
        {
                return l*b;
        }
}
class overmethoddemo
{
        public static void main(String args[])
        {
                A a1=new A();
                int r1=a1.area();
                System.out.println("The area is: "+r1);
                 int r2=a1.area(5,20);
                 System.out.println("The area is: "+r2);
        }
}
```

Output:

```
The area is: 200
The area is: 100
```

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

Aim: To write a JAVA program to implement constructor

Programs:

(i) A constructor with no parameters:

```
class A
{
        int l,b;
        A()
        {
                l=10;
                b=20;
        }
        int area()
        {
               return l*b;
        }
}
class constructordemo
{
        public static void main(String args[])
        {
               A a1=new A();
               int r=a1.area();
               System.out.println("The area is: "+r);

        }
}
```

Output:

```
The area is:200
```

(ii) A constructor with parameters

```
class A
{
        int l,b;
        A(int u,int v)
        {
              l=u;
              b=v;
        }
        int area()
        {
             return l*b;
        }
}
class constructordemo
{
        public static void main(String args[])
        {
              A a1=new A(10,20);
              int r=a1.area();
              System.out.println("The area is: "+r);
        }
}
```

Output:

```
The area is:200
```

3 d). Write a JAVA program to implement constructor overloading.

Aim: To write a JAVA program to implement constructor overloading

Program:

```
class A
{
	int l,b;
	A()
	{
		l=10;
		b=20;
		 }
	A(int u,int v)
	{
		l=u;
		b=v;
	}
	int area()
	{
		return l*b;
	}
}
class overconstructdemo
{
	 public static void main(String args[])
	 {
		 A a1=new A();
		int r1=a1.area();
		System.out.println("The area is: "+r1);
		A a2=new A(30,40);
		int r2=a2.area();
		System.out.println("The area is: "+r2);
	}
}
```

Output:

```
The area is: 200
The area is: 1200
```

Exercise - 4

a) Write a JAVA program to implement Single Inheritance

Aim: To write a JAVA program to implement Single Inheritance

Program:

```
class A
{
      A()
      {
              System.out.println("Inside A's Constructor");
      }
}
class B extends A
{
      B()
      {    Super();
              System.out.println("Inside B's Constructor");
      }
}
class Singledemo
{
       public static void main(String args[])
       {
              B b1=new B();//Constructor calling
        }
}
```

Output:

```
Inside A's Constructor
Inside B's Constructor
```

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

Aim: To write a JAVA program to implement multi level Inheritance

Program:

```
class A
{
      A()
      {
                System.out.println("Inside A's Constructor");
      }
}
class B extends A
{
      B()
      {
               System.out.println("Inside B's Constructor");
      }
}
class C extends B
{
      C()
      {
               System.out.println("Inside C's Constructor");
      }
}
class multidemo
{
       public static void main(String args[])
       {
               C c1=new C();

        }
}
```

Output:

```
Inside A's Constructor
Inside B's Constructor
Inside C's Constructor
```

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

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

Program:

```
abstract class shape//
{
          abstract double area();//
}
class rectangle extends shape
{
          double l=12.5,b=2.5;
          double area()
          {
                    return l*b;
          }
}
class triangle extends shape
{
         double b=4.2,h=6.5;
         double area()
         {
                   return 0.5*b*h;
         }
}
class square extends shape
{
         double s=6.5;
         double area()
         {
                  return 4*s;
         }
}
class Shapedemo
{
         public static void main(String[] args)
         {
                    rectangle r1=new rectangle();
                   triangle t1=new triangle();
                   square s1=new square();
                   System.out.println("The area of rectangle is: "+r1.area());
                   System.out.println("The area of triangle is: "+t1.area());
                   System.out.println("The area of square is: "+s1.area());
         }
}
```

Output:

```
The area of rectangle is: 31.25
The area of triangle is: 13.65
The area of square is: 26.0
```

Exercise - 5

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

Aim: Write a JAVA program give example for “super” keyword

Programs:

(i) Using super to call super class constructor (Without parameters)

```
class A
{
        int l,b;
        A()
        {
            l=10;
            b=20;
        }
}
class B extends A
{
        int h;
        B()
        {
             super();
             h=30;
        }
        int volume()
        {
             return l*b*h;
        }
}
class superdemo
{
          public static void main(String args[])
          {
                    B b1=new B();
                    int r=b1.volume();
                    System.out.println("The vol. is: "+r);
          }
}
```

Output:

```
The vol. is:6000
```

(ii) Using super to call super class constructor (With parameters)

```
class A
{
        int l,b;
        A(int u,int v)
        {
                l=u;
                b=v;
        }
}
class B extends A
{
         int h;
         B(int u,int v,int w)
         {
                  super(u,v);
                  h=w;
         }
         int volume()
         {
                  return l*b*h;
         }
}
class superdemo
{
         public static void main(String args[])
         {
                 B b1=new B(30,20,30);
                 int r=b1.volume();
                 System.out.println("The vol. is: "+r);
         }
}
```

Output:

```
The vol. is:18000
```

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

Aim: To write a JAVA program to implement Interface.

Programs:

(i) First form of interface implementation

```
interface A
{
        void display();
}
class B implements A
{
        public void display()
        {
              System.out.println("B's method");
        }
}
class C extends B
{
        public void callme()
        {
              System.out.println("C's method");
        }
}
class interfacedemo
{
        public static void main(String args[])
        {
             C c1=new C();
             c1.display();
             c1.callme();
        }
}
```

Output:

```
B's method
C's method
```

(ii) Second form of interface implementation

```
interface D
{
        void display();
}
interface E extends D
{
        void show();
}
class A
{
        void callme()
        {
                System.out.println("This is in callme method");
        }
}
```

```
class B extends A implements E
{
        public void display()
        {
                System.out.println("This is in display method");
        }
        public void show()
        {
               System.out.println("This is in show method");
        }
}
class C extends B
{
        void call()
        {
                  System.out.println("This is in call method");
        }
}
class interfacedemo
{
        public static void main(String args[])
        {
               C c1=new C();
               c1.display();
               c1.show();
               c1.callme();
               c1.call();
        }
}
```

Output:

```
This is in display method
This is in show method
This is in callme method
This is in call method
```

(iii) Third form of interface implementation

```
interface A
{
        void display();
}
class B implements A
{
        public void display()
        {
                  System.out.println("This is in B's method");
        }
}
class C implements A
{
        public void display()
        {
                  System.out.println("This is C's method");
        }
}
class interfacedemo
{
        public static void main(String args[])
        {
                B b1=new B();
                C c1=new C();
                b1.display();
                c1.display();
        }
}
```

Output:

```
This is in B's method
This is C's method
```

(iv) Fourth form of interface implementation

```
interface A
{
        void display();
}
interface B
{
        void callme();
}
interface C extends A,B
{
        void call();
}
class D implements C
{
        public void display()
        {
                System.out.println("interface A");
        }
        public void callme()
        {
                System.out.println("interface B");
        }
        public void call()
        {
                System.out.println("interface C");
         }
}
class interfacedemo
{
         public static void main(String args[])
         {
                D d1=new D();
                d1.display();
                d1.callme();
                d1.call();
         }
}
```

Output:

```
interface A
interface B
interface C
```

5 c) Write a JAVA program that implements Runtime polymorphism

Program:

Aim: To write a JAVA program that implements Runtime polymorphism

```
class A
{
          void display()
          {
                   System.out.println("Inside A class");
          }
}
class B extends A
{
         void display()
         {
                  System.out.println("Inside B class");
         }
}
class C extends A
{
         void display()
         {
                 System.out.println("Inside C class");
         }
}
class runtimedemo
{
        public static void main(String args[])
        {
                 A a1=new A();
                 B b1=new B();
                 C c1=new C();
                 A ref;
                 ref=c1;
                 ref.display();
                 ref=b1;
                 ref.display();
                 ref=a1;
                 ref.display();
       }
}
```

Output:

```
Inside C class
Inside B class
Inside A class
```

Exercise - 6

a) Write a JAVA program that describes exception handling mechanism

Aim: To write a JAVA program that describes exception handling mechanism

Program:

Usage of Exception Handling:

```
 class trydemo
 {
         public static void main(String args[])
         {
                 try
                 {
                         int a=10,b=0;
                         int c=a/b;
                         System.out.println(c);
                 }
                 catch(ArithmeticException e)
                 {
                         System.out.println(e);
                 }
                 System.out.println("After the catch statement");
         }
}
```

Output:

```
java.lang.ArithmeticException: / by zero
After the catch statement
```

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

Program:

Aim: To write a JAVA program Illustrating Multiple catch clauses

```
class multitrydemo
{
        public static void main(String args[])
        {
                try
                {
                        int a=10,b=5;
                        int c=a/b;
                        int d[]={0,1};
                        System.out.println(d[10]);
                        System.out.println(c);
                }
                catch(ArithmeticException e)
                {
                        System.out.println(e);
                }
                catch(ArrayIndexOutOfBoundsException e)
                {
                       System.out.println(e);
                }
                System.out.println("After the catch statement");
        }
}
```

Output:

```
java.lang.ArrayIndexOutOfBoundsException: 10
After the catch statement
```

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

Aim: To write a JAVA program for creation of Java Built-in Exceptions **Programs:**

(i) Arithmetic exception

```
class arithmeticdemo
{
	public static void main(String args[])
	{
		try
		{
			int a = 10, b = 0;
			int c = a/b;
			System.out.println (c);
		}
		catch(ArithmeticException e)
		{
			System.out.println (e);
		}
	}
}
```

Output:

```
java.lang.ArithmeticException: / by zero
```

(ii) NullPointer Exception

```
class  nullpointerdemo
{
	 public static void main(String args[])
	 {
		  try
		  {
			   String a = null;
			   System.out.println(a.charAt(0));
		  }
```

```
                catch(NullPointerException e)
                {
                        System.out.println(e);
                }
        }
}
```

Output:

```
java.lang.NullPointerException
```

(iii) StringIndexOutOfBound Exception

```
      public class StringIndexDemo
      {
          public static void main(String[] args)
         {
              String text = "Hello";

              try
              {
                  char ch = text.charAt(10);
                  System.out.println("Character at index 10: " + ch);
              }
             catch (StringIndexOutOfBoundsException e)
             {
                  System.out.println("Exception caught: " + e);
                  System.out.println("String length is: " + text.length());
              }

              System.out.println("Program continues...");
          }
      }
```

Output:

```
Exception caught: java.lang.StringIndexOutOfBoundsException: String index out of range: 10
String length is: 5
Program continues...
```

(iv) FileNotFound Exception

```
import java.io.*;
class filenotfounddemo
{
        public static void main(String args[])
        {
                try
                {
                        File file = new File("E://file.txt");
                        FileReader fr = new FileReader(file);
                }
                catch (FileNotFoundException e)
                {
                        System.out.println(e);
                }
        }
}
```

Output:

```
java.io.FileNotFoundException: E:\file.txt (The system cannot find the file specified)
```

(v) NumberFormat Exception

```
class  numberformatdemo
{
        public static void main(String args[])
        {
                try
                {
                        int num = Integer.parseInt ("akki") ;
                        System.out.println(num);
                }
                catch(NumberFormatException e)
                {
                        System.out.println(e);
                }
        }
}
```

Output:

```
java.lang.NumberFormatException: For input string: "akki"
```

(vi) ArrayIndexOutOfBounds Exception

```
class arraybounddemo
{
        public static void main(String args[])
        {
        try
                {
                                    int a[] = new int[5];
                                    a[6] = 9;

                }



 catch(ArrayIndexOutOfBoundsException e)
                {
                        System.out.println (e);
                }
        }
}
```

Output:

```
java.lang.ArrayIndexOutOfBoundsException: 6
```

a)creation of illustrating throw

Program:

Aim: To write a JAVA program for creation of Illustrating throw

```
class throwdemo
{
            public static void main(String args[])
            {
                        try
                        {
                                     throw new NullPointerException("demo");
                        }
                        catch(NullPointerException e)
                        {
                                    System.out.println(e);
                        }
           }
}
```

Output:

```
java.lang.NullPointerException: demo
```

b) creation of illustrating finally

Aim: To write a JAVA program for creation of Illustrating finally

Program(i):

```
class finallydemo
{
	public static void main(String args[])
	{
		try
		{
			int a=10,b=0;
			int c=a/b;
			System.out.println(c);
		}
		catch(ArithmeticException e)
		{
			System.out.println(e);
		}
		finally
		{
			System.out.println("This is inside finally block");
		}
	}
}
```

Output:

```
java.lang.ArithmeticException: / by zero
This is inside finally block
```

Program(ii):

```
class finallydemo
{
	public static void main(String args[])
	{
		try
		{
			int a=10,b=5;
			int c=a/b;
			System.out.println(c);
		}
		catch(ArithmeticException e)
		{
			System.out.println(e);
		}
		finally
		{
			System.out.println("This is inside finally block");
		}
	}
}
```

Output:

```
2
This is inside finally block
```

6 D) Write a JAVA program for creation of User Defined Exception

Aim: To write a JAVA program for creation of User Defined Exception **Program:**

```
class A extends Exception
{
	A(String s1)
	{
		super(s1);
	}
}
class owndemo
{
	public static void main(String args[])
	{
		try
		{
			throw new A("demo ");
		}
		catch(Exception e)
		{
			System.out.println(e);
		}
	}
}
```

Output:

```
A: demo
```

Exercise – 7 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)

Aim: To 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)

Programs:

(i) Creating multiple threads using Thread class

```
class A extends Thread
{
          public void run()
          {
                    try
                    {


     for(int i=1;i<=10;i++)
     {
     sleep(1000);
      System.out.println("good morning");
                                 }
                    }
                    catch(Exception e)
                    {
                                 System.out.println(e);
                    }
          }
}

class B extends Thread
{
          public void run()
          { try
                    {

    for(int j=1;j<=10;j++)
    {
    sleep(2000); System.out.println("hello");
                               }
                    }
                    catch(Exception e)
                    {
                              System.out.println(e);
                    }
         }
}
class C extends Thread
{
         public void run()
    {
        try
  {
            for(int k=1;k<=10;k++)
            {
                sleep(3000);
                System.out.println("welcome");
            }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
class threaddemo
{
    public static void main(String args[])
    {
        A a1=new A();
        B b1=new B();
        C c1=new C();
        a1.start();
        b1.start();
        c1.start();
    }
}
```

Output:

```
good morning
hello
good morning
good morning
welcome
hello
good morning
good morning
hello
good morning
welcome
good morning
hello
good morning
good morning
welcome
hello
good morning
hello
welcome
hello
welcome
hello
hello
welcome
hello
welcome
```

(ii) Creating multiple threads using Runnable interface

```
class A implements Runnable
{
        public void run()
        {
                try
                {
                        for(int i=1;i<=10;i++)
                        {
                                Thread.sleep(1000);
                                System.out.println("good morning");
                        }
                }
                catch(Exception e)
                {
                       System.out.println(e);
                 }
        }
}
class B implements Runnable
{
        public void run()
        {
               try
               {
                       for(int j=1;j<=10;j++)
                       {
                               Thread.sleep(2000);
                               System.out.println("hello");
                       }
              }
              catch(Exception e)
              {
                      System.out.println(e);
              }
      }
}
class C implements Runnable
{
      public void run()
      {
             try
             {
                    for(int k=1;k<=10;k++)
                    {
                           Thread.sleep(3000);
                           System.out.println("welcome");
                    }
             }
             catch(Exception e)
             {
                    System.out.println(e);
             }
        }
}
class runnabledemo
{
        public static void main(String args[])
        {
                A a1=new A();
                B b1=new B();
                C c1=new C();
                Thread t1=new Thread(a1);
                Thread t2=new Thread(b1);
                Thread t3=new Thread(c1);
                t1.start();
                 t2.start();
                 t3.start();
       }
}
```

Output:

```
good morning
good morning
hello
good morning
welcome
good morning
hello
good morning
good morning
welcome
hello
good morning
good morning
hello
good morning
welcome
good morning
hello
welcome
hello
hello
welcome
hello
welcome
hello
hello
welcome
welcome
welcome
welcome
```

7 B) Write a program illustrating **isAlive** and **join ()**

Aim: To write a program illustrating isAlive and join ()

Program:

```
class A extends Thread
{
        public void run()
        {
                try
                {
                        for(int i=1;i<=10;i++)
                        {
                                sleep(1000);
                                System.out.println("good morning");
                        }
                }
                catch(Exception e)
                {
                         System.out.println(e);
                }
        }
}
class B extends Thread
{
        public void run()
        {
                try
                {
                        for(int j=1;j<=10;j++)
                        {
                                sleep(2000);
                                System.out.println("hello");
                        }
                }
                catch(Exception e)
                {
                        System.out.println(e);
                }
        }
}
class C extends Thread
{
        public void run()
        {
                try
                {
                        for(int k=1;k<=10;k++)
                        {
                                sleep(3000);
                                System.out.println("welcome");
                        }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
class isalivedemo
{
    public static void main(String args[])
    {
        A a1=new A();
        B b1=new B();
        C c1=new C();
        a1.start();
        b1.start();
        c1.start();
        System.out.println(a1.isAlive());
        System.out.println(b1.isAlive());
        System.out.println(c1.isAlive());
        try
        {
            a1.join();
            b1.join();
            c1.join();
        }
        catch(InterruptedException e)
        {
            System.out.println(e);
        }
        System.out.println(a1.isAlive());
        System.out.println(b1.isAlive());
        System.out.println(c1.isAlive());
    }
}
```

Output:

```
true
true
true
good morning
good morning
hello
good morning
welcome
good morning
hello
good morning
good morning
welcome
hello
good morning
good morning
hello
good morning
welcome
good morning
hello
welcome
hello
hello
welcome
hello
welcome
hello
hello
welcome
welcome
welcome
welcome
false
false
false
```

c)Implementation of Daemon Threads

7 C) Write a Program illustrating Daemon Threads.

Aim: To write a Program illustrating Daemon Threads

Program:

```
class A extends Thread
{
        public void run()
        {
                if(Thread.currentThread().isDaemon())
                System.out.println("daemon thread work");
                else
                System.out.println("user thread work");
        }
}
class daemondemo
{
        public static void main(String[] args)
        {
                A a1=new A();
                A a2=new A();
                A a3=new A();
                a1.setDaemon(true);
                a1.start();
                a2.start();
                a3.start();
         }
}
```

Output:

```
daemon thread work
user thread work
user thread work
```

7 d).Write a JAVA program Producer Consumer Problem

Aim: Write a JAVA program Producer Consumer Problem

Program:

```
class A
{
        int n;
        boolean b=false;
        synchronized int get()
        {
                if(!b)
                try
                {
                         wait();
                }
                catch(Exception e)
                {
                        System.out.println(e);
                }
                System.out.println("Got:"+n);
                b=false;
                notify();
                return n;
       }
       synchronized void put(int n)
       {
                if(b)try
                {

                wait();

                }
                      catch(Exception e)
                {
                       System.out.println(e);
                }
                this.n=n;
                b=true;
                System.out.println("Put:"+n);
                notify();
        }
}
class producer implements Runnable
{
        A a1;
        Thread t1;
        producer(A a1)
        {
                this.a1=a1;
                t1=new Thread(this);
                t1.start();
        }
        public void run()
        {
                for(int i=1;i<=10;i++)
                {
                        a1.put(i);
                }
        }
}

class consumer implements Runnable
{
        A a1;
        Thread t1;
        consumer(A a1)
        {
                this.a1=a1;
                t1=new Thread(this);
                t1.start();
        }
        public void run()
        {
                for(int j=1;j<=10;j++)
                {
                        a1.get();
                }
        }
}
class interdemo
{
        public static void main(String args[])
        {
                A a1=new A();
                producer p1=new producer(a1);
                consumer c1=new consumer(a1);
        }
}
```

Output:

```
Put:1
Got:1
Put:2
Got:2
Put:3
Got:3
Put:4
Got:4
Put:5
Got:5
Put:6
Got:6
Put:7
Got:7
Put:8
Got:8
Put:9
Got:9
Put:10
Got:10
```

Exercise – 8

a). Write a java program to illustrate the classpath.

Package in Java is a mechanism to encapsulate a group of classes, sub-packages, and interfaces. Packages are used for:

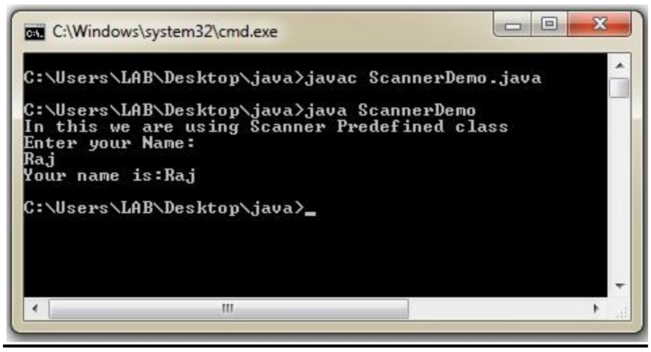
- Preventing naming conflicts. For example, there can be two classes with the name Employee in two packages, college.staff.cse.Employee and college.staff.ee.Employee
- Making searching/locating and usage of classes, interfaces, enumerations, and annotations easier Packages can be considered as data encapsulation (or data-hiding). Here we will be discussing the responsibility of the CLASSPATH environment variable while programming in Java as we move forward we for sure short need usage of importing statements.

Explantion: importvlits.cse.Student;

What does this import mean? It makes the Studentclass available in the package vlits.cseto our current class. Such that when we call the below command as follows: Student s = new Student();

```
importjava.util.*;
class  ScannerDemo
{
        public static void main(String[] args)
        {
                System.out.println("In this we are using Scanner Predefined class");
                Scanner s = new Scanner();
                System.out.println("Enter your Name:");
                String name=s.next();
                System.out.println("Your name is:"+name);
        }
}
```

Output:



```
C:\Users\LAB\Desktop\java>javac ScannerDemo.java

C:\Users\LAB\Desktop\java>java ScannerDemo
In this we are using Scanner Predefined class
Enter your Name:
Raj
Your name is:Raj

C:\Users\LAB\Desktop\java>_
```

In the above program we used Scanner(predefined) class. When we compile and execute the above program the system will look about the directories which belongs to java in the directory "C:\Program Files\Java\jdk1.8.0_202\bin" and this will be set in the class path.

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

Case study on including class path:

There are different ways to set a class path. **Java Virtual Machine(JVM)** uses classpath to locate the class files to execute the code.If your classpath is not set and the class file is not present in the same directory as your java file, then JVM will be unable to find the required class file, and it will throw an error (java.lang.ClassNotFoundException).

Different ways to set the classpath:

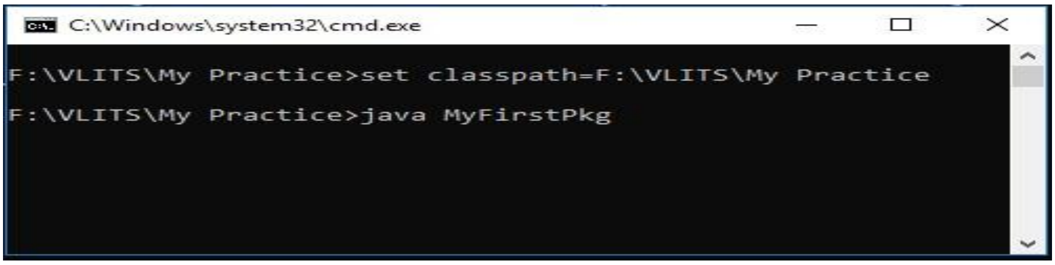
1. Temporary Setting

2. Permanent setting

1. Temporarily setting the class path:

>set classpath=<directory_location>

- The limitation of the **"set classpath"** command option is the classpath settings are available only for the current command prompt.

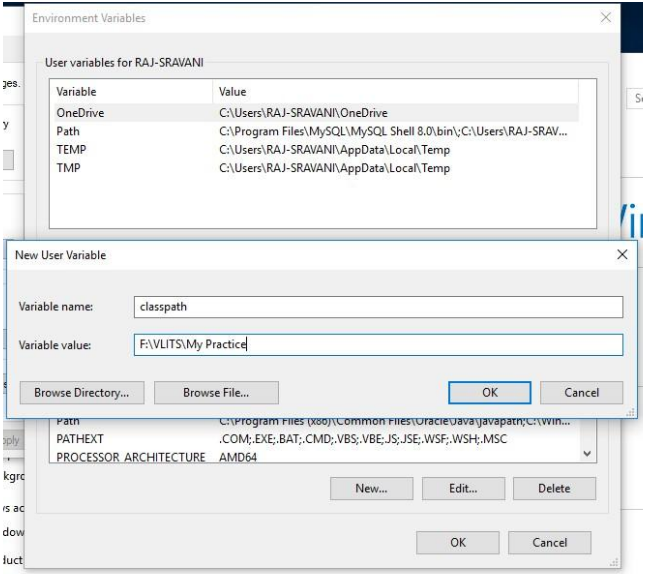


```
F:\VLITS\My Practice>set classpath=F:\VLITS\My Practice

F:\VLITS\My Practice>java MyFirstPkg
```

2. Permanent setting the class path:

If we set class path in environment variables it will available to all the command prompts.

1. Firstly, Right Click on "This PC".
2. Click Properties.
3. Click "Advanced System Settings".
4. Click "Environment Variables".
5. In the "User Variable Section", click the "New" button.
6. Enter Variable name :classpath [Don't give space between class path] Variable value:<directory_location>(for example in my F:\workspace\bin)
7. Click OK->OK->OK.
8. Close all windows, open a new command prompt, and run the java command



Variable	Value
OneDrive	C:\Users\RAJ-SRAVANI\OneDrive
Path	C:\Program Files\MySQL\MySQL Shell 8.0\bin\;C:\Users\RAJ-SRAV...
TEMP	C:\Users\RAJ-SRAVANI\AppData\Local\Temp
TMP	C:\Users\RAJ-SRAVANI\AppData\Local\Temp

Variable name: classpath

Variable value: F:\VLITS\My Practice

Path	C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Win...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

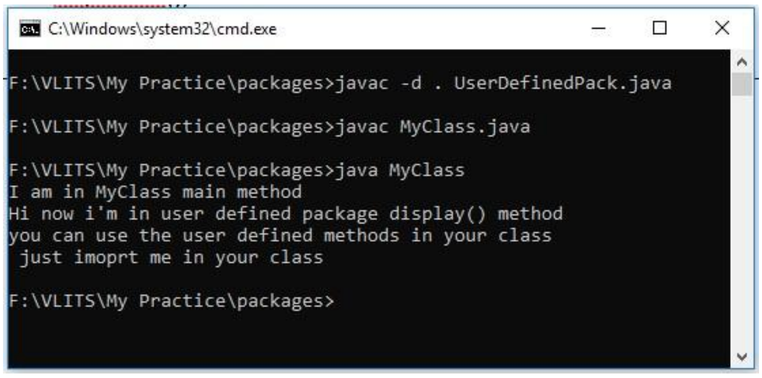
We must include **.;** in the classpath beginning so that the **JVM** can access both the current working directory anc the directory of the desired class file, respectively.

c). Write a java program that import and use the user defined packages.

```
package Mypkg;
public class  UserDefinedPack
{
        public void display()
         {
               System.out.println("Hi now i'm in user defined package display() method");
         }
        public void show()
         {
  System.out.println("you can use the user defined methods in your class\n just imoprt me in
your class");
         }
}
```

```
import Mypkg.UserDefinedPack;
public class  MyClass
{
        public static void main(String[] args)
         {
                System.out.println("I am in MyClass main method");
               UserDefinedPackudp = new UserDefinedPack();//creating object
              // calling the Mypkg package UserDefinedPack class methods in MyClass

                 udp.display();
                 udp.show();
         }
}
```

<u>Output:</u>



```
F:\VLITS\My Practice\packages>javac -d . UserDefinedPack.java

F:\VLITS\My Practice\packages>javac MyClass.java

F:\VLITS\My Practice\packages>java MyClass
I am in MyClass main method
Hi now i'm in user defined package display() method
you can use the user defined methods in your class
 just imoprt me in your class

F:\VLITS\My Practice\packages>
```

Exercise: 9(Applets)

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

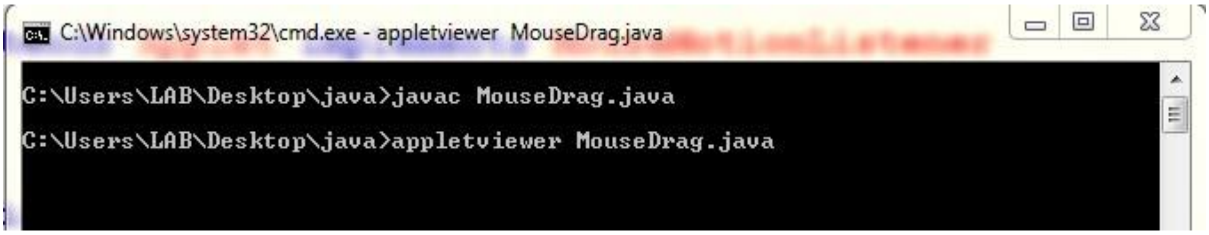
```
import java.awt.*;   import java.awt.event.*;   import java.applet.*;
public class MouseDrag extends Applet implements MouseMotionListener
{
 public void init()
 {
addMouseMotionListener(this);
setBackground(Color.red);
```

```
 }
 public void mouseDragged(MouseEvent me)
 {
 Graphics g=getGraphics();
g.setColor(Color.white);
g.fillOval(me.getX(),me.getY(),10,10);  // (x-position, y-postion, width, height)
 }
 public void mouseMoved(MouseEvent me)
 {
 }
}
```

Applet Code: Applet Code save as .html file.

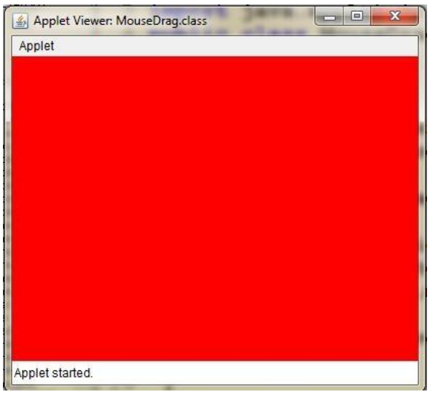
```
<html>
<applet code="MouseDrag.class" height=300 width=400>
</applet>
</html>
```

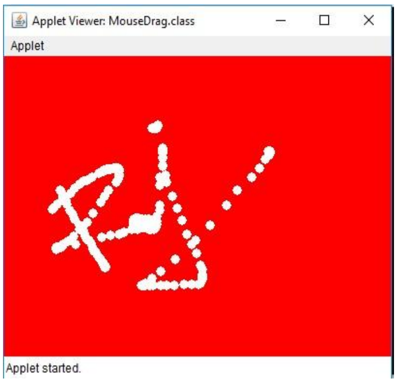
Output:

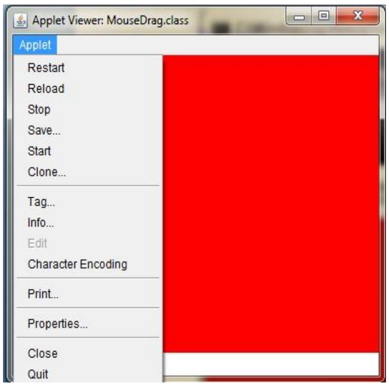


C:\Windows\system32\cmd.exe - appletviewer MouseDrag.java

```
C:\Users\LAB\Desktop\java>javac MouseDrag.java

C:\Users\LAB\Desktop\java>appletviewer MouseDrag.java
```







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

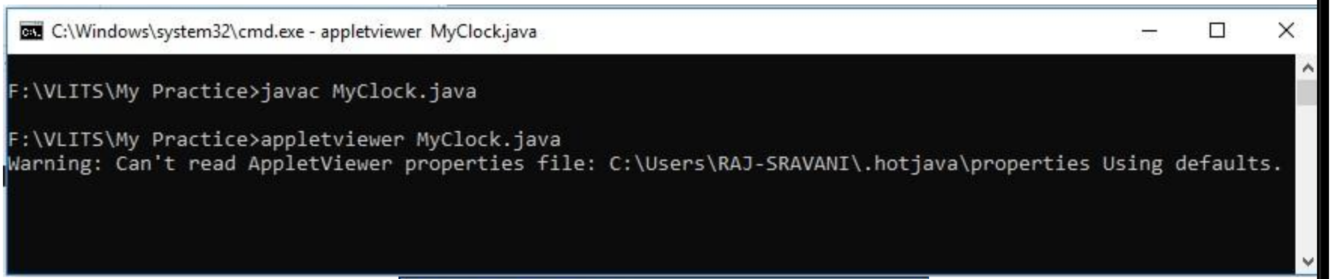
```
import java.applet.*;   import java.awt.*;   import java.util.*;   import java.text.*;
public class MyClock extends Applet implements Runnable
{
int width, height;    Thread t =
null;   boolean
threadSuspended;   int hours=0,
minutes=0, seconds=0;
 String timeString = "";
 public void init()
 {
       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.white );
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 );
  }
}
```

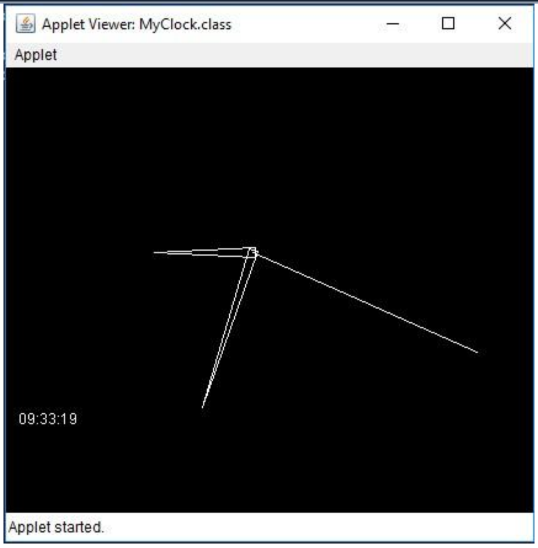
Applet Code: Applet Code save as .html file.

```
<html>
<applet code="MyClock.class" height=300 width=400>
</applet>
</html>
```

Output:



```
F:\VLITS\My Practice>javac MyClock.java

F:\VLITS\My Practice>appletviewer MyClock.java
Warning: Can't read AppletViewer properties file: C:\Users\RAJ-SRAVANI\.hotjava\properties Using defaults.
```



Applet started.

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

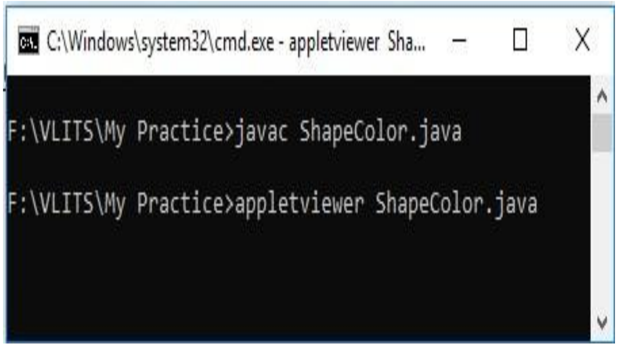
```
import java.applet.*;
import java.awt.*;
public class  ShapeColor extends Applet
{
int x=300,y=100,r=50;
 public void paint(Graphics g)
 {
g.setColor(Color.red);  //Drawing line color is red
g.drawLine(3,300,200,10);
g.setColor(Color.magenta);
g.drawString("Line",100,100);
g.drawOval(x-r,y-r,100,100);
g.setColor(Color.yellow);  //Fill the yellow color in circle
g.fillOval( x-r,y-r, 100, 100 );
g.setColor(Color.magenta);
g.drawString("Circle",275,100);
g.drawRect(400,50,200,100);
g.setColor(Color.yellow);  //Fill the yellow color in rectangel
g.fillRect( 400, 50, 200, 100 );
g.setColor(Color.magenta);
g.drawString("Rectangel",450,100);
   }
}
```

Applet Code: Applet Code save as .html file.

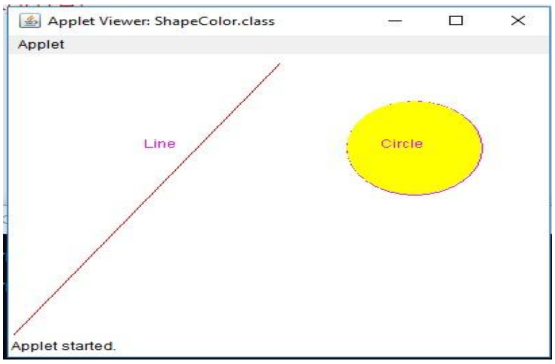
```
<html>
<applet code="ShapeColor.class" height=300 width=400>
```

```
</applet>
</html>
```

Output:



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

F:\VLITS\My Practice>javac ShapeColor.java

F:\VLITS\My Practice>appletviewer ShapeColor.java
```



Exercise: 10(Event Handling)

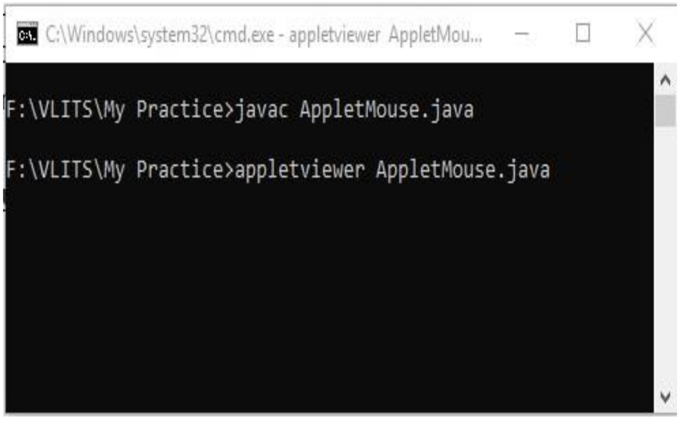
a).Write a JAVA program that display the x and y position of the cursor movement using Mouse.

```
import java.awt.*;
import java.awt.event.*;
import
java.applet.Applet;
public class AppletMouse extends Applet implements MouseListener, MouseMotionListener
{ int x, y;
String str="";
public void init()
  {
addMouseListener(this);
addMouseMotionListener(this);
  }
                                    // override ML 5 abstract
methods   public void mousePressed(MouseEvent e)
  {
    x =
e.getX();     y
= e.getY();
str = "Mouse Pressed";
repaint();
  }
  public void mouseReleased(MouseEvent e)
  {
    x = e.getX();     y =
e.getY(); str = "Mouse
Released";
    repaint();
   }
   public void mouseClicked(MouseEvent e)
   {
     x = e.getX();      y
= e.getY(); str =
"Mouse Clicked";
repaint();
   }
   public void mouseEntered(MouseEvent e)
  {
   x =
e.getX();      y
= e.getY();
str = "Mouse Entered";
   repaint();
  }
  public void mouseExited(MouseEvent e)
  {
   x = e.getX();
y = e.getY(); str =
"Mouse Exited";
repaint();
  }
                        // override two abstract methods of MouseMotionListener
  public void mouseMoved(MouseEvent e)
  {
   x =
e.getX();      y
= e.getY();
str = "Mouse Moved";
   repaint();
  }
  public void mouseDragged(MouseEvent e)
  {
   x =
e.getX();      y
= e.getY();
str = "Mouse dragged";
repaint();
  }
                        // called by repaint() method
  public void paint(Graphics g)
  {
g.setFont(new Font("Monospaced", Font.BOLD, 20));
g.fillOval(x, y, 10, 10);
g.drawString(x + "," + y,  x+10, y -10);
g.drawString(str, x+10, y+20);
showStatus(str + " at " + x + "," + y);
  }
}
```

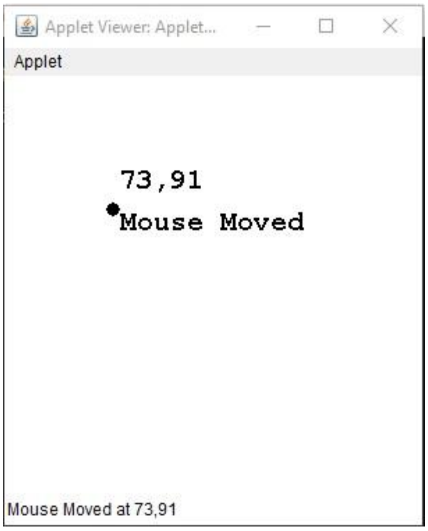
Applet code:Applet code save as .html file

```
<html>
<body>
<applet code="AppletMouse.class" width="300" height="300">
</applet>
</body>
</html>
```

Output:



```
F:\VLITS\My Practice>javac AppletMouse.java

F:\VLITS\My Practice>appletviewer AppletMouse.java
```



```
73,91
Mouse Moved

Mouse Moved at 73,91
```

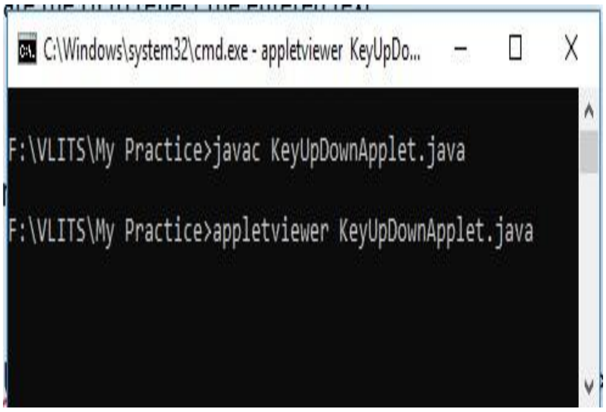
b).Write a JAVA program that identifies key-up key-down event user entering text in an Applet.

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

public class KeyUpDownApplet extends Applet implements KeyListener {

   String message = ""; // To display the key events
   String typedText = ""; // To display the entered text

   //Override method
public void init() {
       // Add KeyListener to the applet
addKeyListener(this);
       requestFocus(); // Request focus so the applet can capture key events
setBackground(Color.white);
   }

   //Override method
public void paint(Graphics g)
{
       g.drawString("Typed Text: " + typedText, 20, 100);
       g.drawString("Message: " + message, 20, 120);
   }

   // Handle key down event
   //Override method
   public void keyPressed(KeyEvent e) {        int key =
e.getKeyCode();        message = "Key Pressed: " +
KeyEvent.getKeyText(key);        repaint(); // Update
the UI to reflect key down event
   }

   // Handle key up event
//Override method
   public void keyReleased(KeyEvent e) {
       int key = e.getKeyCode();        message = "Key
Released: " + KeyEvent.getKeyText(key);
repaint(); // Update the UI to reflect key up event
    }

    // Handle key typed event (for entering text)
    //Override method
    public void keyTyped(KeyEvent e) {
char c = e.getKeyChar();
        typedText += c; // Append the typed character to the text string
repaint(); // Update the UI to reflect the entered text
    }

    // No need to override start, stop, or destroy methods for this task

}
/*<applet code="KeyUpDownApplet.class" width="400" height="200"></applet>*/
```

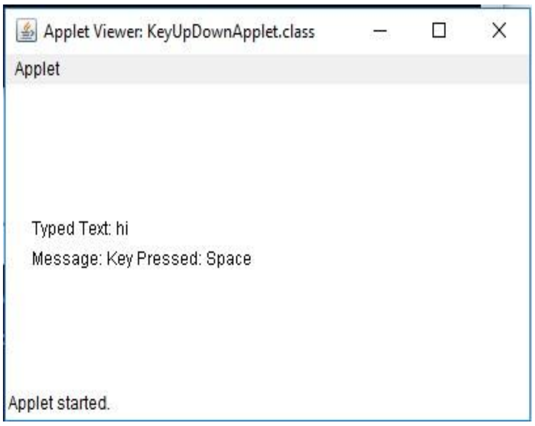
Applet code:Applet code save as .html

```
<html>
<body>
<applet code=" KeyUpDownApplet.class" width="300" height="300">
</applet>
</body>
</html>
```

Output:



```
F:\VLITS\My Practice>javac KeyUpDownApplet.java

F:\VLITS\My Practice>appletviewer KeyUpDownApplet.java
```



```
Typed Text: hi
Message: Key Pressed: Space

Applet started.
```

Exercise: 11(Swings)

a).Write a java program to built a Calculator in swings

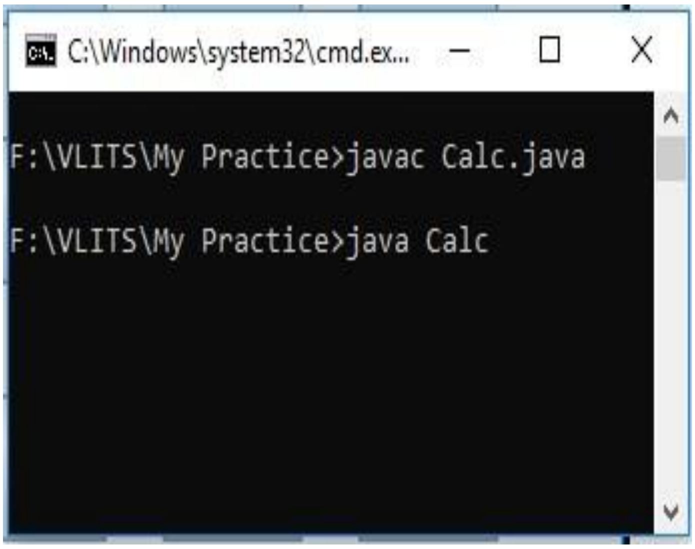
```
import javax.swing.*; import
java.awt.event.*; class Calc
implements ActionListener
{
JFrame f;
JTextField t;
JButton b1,b2,b3,b4,b5,b6,b7,b8,b9,b0,bdiv,bmul,bsub,badd,bdec,beq,bdel,bclr;
```

```
    static double a=0,b=0,result=0;
static int operator=0;

Calc()
    {
        f=new
JFrame("Calculator");
t=new JTextField();
b1=new JButton("1");
b2=new JButton("2");
b3=new JButton("3");
b4=new JButton("4");
b5=new JButton("5");
b6=new JButton("6");
b7=new JButton("7");
b8=new JButton("8");
b9=new JButton("9");
b0=new JButton("0");
bdiv=new JButton("/");
bmul=new JButton("*");
bsub=new JButton("-");
badd=new JButton("+");
bdec=new JButton(".");
beq=new JButton("=");
bdel=new JButton("Delete");
bclr=new JButton("Clear");

t.setBounds(30,40,280,30);
b7.setBounds(40,100,50,40);
b8.setBounds(110,100,50,40);
b9.setBounds(180,100,50,40);
bdiv.setBounds(250,100,50,40);

        b4.setBounds(40,170,50,40);
b5.setBounds(110,170,50,40);
b6.setBounds(180,170,50,40);
bmul.setBounds(250,170,50,40);

        b1.setBounds(40,240,50,40);
b2.setBounds(110,240,50,40);
b3.setBounds(180,240,50,40);
bsub.setBounds(250,240,50,40);

bdec.setBounds(40,310,50,40);
b0.setBounds(110,310,50,40);
beq.setBounds(180,310,50,40);
badd.setBounds(250,310,50,40);

bdel.setBounds(60,380,100,40);
bclr.setBounds(180,380,100,40);

f.add(t);
f.add(b7);
f.add(b8);
```

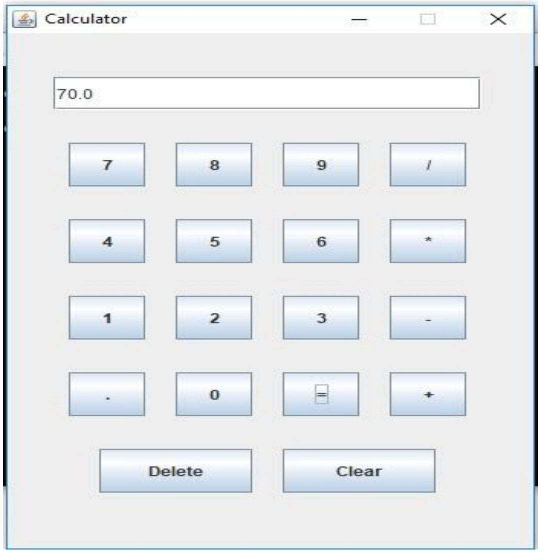
```
f.add(b9);
f.add(bdiv);
f.add(b4);
f.add(b5);
f.add(b6);
f.add(bmul);
f.add(b1);
f.add(b2);
f.add(b3);
f.add(bsub);
f.add(bdec);
f.add(b0);
f.add(beq);
f.add(badd);
f.add(bdel);
f.add(bclr);

f.setLayout(null);
f.setVisible(true);
f.setSize(350,500);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setResizable(false);

b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
b6.addActionListener(this);
b7.addActionListener(this);
b8.addActionListener(this);
b9.addActionListener(this);
b0.addActionListener(this);
badd.addActionListener(this);
bdiv.addActionListener(this);
bmul.addActionListener(this);
bsub.addActionListener(this);
bdec.addActionListener(this);
beq.addActionListener(this);
bdel.addActionListener(this);
bclr.addActionListener(this);
   }

   public void actionPerformed(ActionEvent e)
   {
      if(e.getSource()==b1)
t.setText(t.getText().concat("1"));

      if(e.getSource()==b2)
t.setText(t.getText().concat("2"));

      if(e.getSource()==b3)
t.setText(t.getText().concat("3"));

      if(e.getSource()==b4)
```

```
t.setText(t.getText().concat("4"));

        if(e.getSource()==b5)
t.setText(t.getText().concat("5"));

        if(e.getSource()==b6)
t.setText(t.getText().concat("6"));

        if(e.getSource()==b7)
t.setText(t.getText().concat("7"));

        if(e.getSource()==b8)
t.setText(t.getText().concat("8"));

        if(e.getSource()==b9)
t.setText(t.getText().concat("9"));

        if(e.getSource()==b0)
t.setText(t.getText().concat("0"));

        if(e.getSource()==bdec)
t.setText(t.getText().concat("."));

        if(e.getSource()==badd)
        {

a=Double.parseDouble(t.getText());
operator=1; t.setText("");
        }

        if(e.getSource()==bsub)
        {

a=Double.parseDouble(t.getText());
operator=2; t.setText("");
        }

        if(e.getSource()==bmul)
        {

a=Double.parseDouble(t.getText());
operator=3; t.setText("");
        }

        if(e.getSource()==bdiv)
        {

a=Double.parseDouble(t.getText());
operator=4; t.setText("");
        }

        if(e.getSource()==beq)
        {
            b=Double.parseDouble(t.getText());
            switch(operator)
            {
               case 1: result=a+b;
break;

               case 2: result=a-b;
break;

               case 3: result=a*b;
break;

               case 4: result=a/b;
break;

               default: result=0;
            }

t.setText(""+result);
        }

        if(e.getSource()==bclr)
t.setText("");

        if(e.getSource()==bdel)
        {
            String s=t.getText();
t.setText("");
for(inti=0;i<s.length()-1;i++)
t.setText(t.getText()+s.charAt(i));
        }
    }

    public static void main(String...s)
    {
        new Calc();
    }
}
```

Output:



```
F:\VLITS\My Practice>javac Calc.java

F:\VLITS\My Practice>java Calc
```

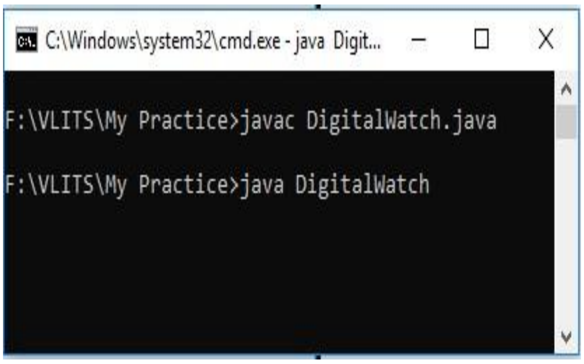


70.0

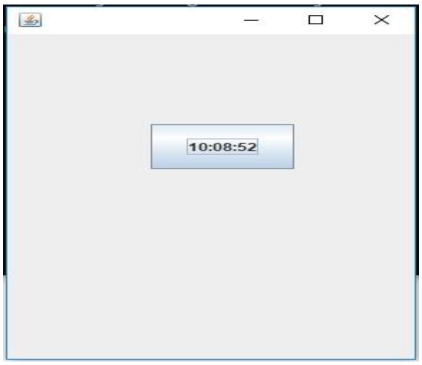
b). Write a JAVA program to display the digital watch in swing tutorial.

```
import javax.swing.*;
import java.awt.*;
import java.text.*;
import java.util.*;
public class DigitalWatch implements Runnable{
JFrame f;
Thread t=null;
int hours=0, minutes=0, seconds=0;
String timeString = "";
JButton b;

DigitalWatch(){
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);
}

 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");
            Date date = cal.getTime();
timeString = formatter.format( date );

printTime();

t.sleep( 1000 );  // interval given in milliseconds
         }
      }
      catch (Exception e) { }
 }

public void printTime(){
b.setText(timeString);
}

public static void main(String[] args) {
new DigitalWatch();


}
}
```

Output:



```
F:\VLITS\My Practice>javac DigitalWatch.java

F:\VLITS\My Practice>java DigitalWatch
```



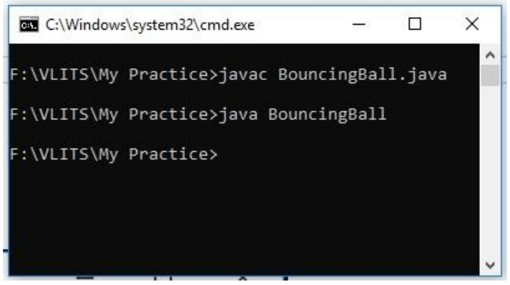
```
10:08:52
```

Exercise: 12(Swings Continued)

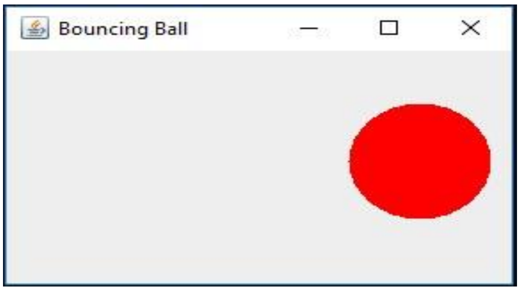
a). Write a JAVA program that to create a single ball bouncing inside a JPanel.

```
import java.awt.*;
import javax.swing.*;
public class BouncingBall extends JPanel {

  // Box height and width
int width;
int height;

  // Ball Size
float radius =
40;
  float diameter = radius * 2;
  // Center of Call
float X = radius +
50;   float Y = radius
+ 20;

  // Direction
float dx = 3;
float dy = 3;

  public BouncingBall() {

    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.RED);
g.fillOval((int)(X-radius), (int)(Y-radius), (int)diameter, (int)diameter);
  }
  public static void main(String[] args) {
JFrame frame = new JFrame("Bouncing Ball");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300, 200);
frame.setContentPane(new BouncingBall());
frame.setVisible(true);
  }
}
```

Output:



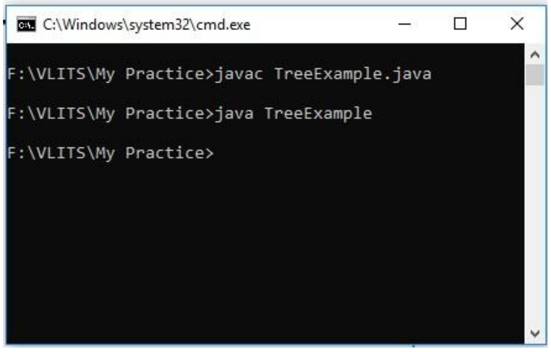
```
F:\VLITS\My Practice>javac BouncingBall.java

F:\VLITS\My Practice>java BouncingBall

F:\VLITS\My Practice>
```



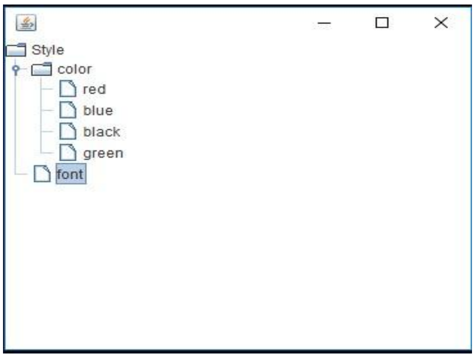
b). Write a JAVA program JTree as displaying a real tree upside down

```
import javax.swing.*;  import
javax.swing.tree.DefaultMutableTreeNode;
public class TreeExample extends JFrame {
JFrame f;
TreeExample(){
f=new JFrame();
DefaultMutableTreeNode style=new DefaultMutableTreeNode("Style");
DefaultMutableTreeNodecolor=new
DefaultMutableTreeNode("color");  DefaultMutableTreeNode
font=new DefaultMutableTreeNode("font");  style.add(color);
style.add(font);
DefaultMutableTreeNode red=new DefaultMutableTreeNode("red");
DefaultMutableTreeNode blue=new DefaultMutableTreeNode("blue");
DefaultMutableTreeNode black=new
DefaultMutableTreeNode("black");  DefaultMutableTreeNode
green=new DefaultMutableTreeNode("green");  color.add(red);
color.add(blue); color.add(black); color.add(green);      JTreejt=new
JTree(style);  f.add(jt);
f.setSize(200,200);
f.setVisible(true);
}
public static void main(String[] args) {
new TreeExample();
}}
```

Output:



```
F:\VLITS\My Practice>javac TreeExample.java

F:\VLITS\My Practice>java TreeExample

F:\VLITS\My Practice>
```



Style
color
red
blue
black
green
font