Lab Manual for

II B. Tech I Semester

Subject: OBJECT ORIENTED PROGRAMMING

Code: A0513

Academic Year 2021-22
Regulations: MR20

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MALLAREDDY ENGINEERING COLLEGE

(An UGC Autonomous Institution, Approved by AICTE and Affiliated to JNTUH Hyderabad,
Recognized under section 2(f) &12(B) of UGC Act 1956,
Accredited by NAAC with 'A'Grade(II Cycle) and NBA
Maisammaguda, Dhulapally (Post ViaKompally), Secunderabad-500 100

Website:www.mrec.ac.in E-mail:principal@mrec.ac.in

List of Programs:

1. Write Java Programs that implement the following.
   a) Constructor
   b) Parameterized constructor
   c) Method Overloading
   d) Constructor overloading
2. Write a Java program
   a) checks whether a given string is a palindrome or not.
   b) for sorting a given list of names in ascending order.
   c) that reads a line if integers and then displays each integer and the sum of all integers(use string tokenizer class of java.util).
3. Write Java programs that uses the following keywords…
   a) this
   b) super
   c) static
   d) final
4. Write a Java program to implement
   a) Method Overriding.
   b) dynamic method dispatch.
   c) multiple inheritance.
   d) access specifiers.
5. Write a Java program that
   a) reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
   b) reads a file and displays the file on the screen, with a line number before each line.
   c) displays the number of characters, lines and words in a test file.
6. Write a Java program for handling
   a) Checked exceptions.
   b) unchecked exceptions.
7. Write a Java program
   a) Creates three threads. First threads displays “Good Morning “for every one Second, the second thread displays “Hello” for every two seconds, the third thread Displays “Welcome” for every three seconds.
   b) that correctly implements producer consumer problem using concept of inter thread communication.
8. Write a Java program which demonstrates the use of following collection classes
   a) Array List
   b) Hash Set
   c) Deque
9. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +,-,*,/ operations. Add a text field to displaythe result.
10. Write a Java program for handling
    a) mouse events.
    b) key events.
11. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields num1 and num2. The division of num1 and num2 is displayed in the result field when the divide button is clicked. If num1 or num2 were not an integer, the program would throw number format exception. If num2 were zero, the program would throw an arithmetic exception and display the exception in the message dialogue box.
12. Write a Java program that
    a) Simulates traffic light. The program lets the user select one of three lights: red, yellow or green. When a radio button is selected, the light is turned on and only one light can be on at a time. No light is on when the program starts.
    b) Allows the user to draw lines rectangles and ovals.

1. Write Java Programs that implement the following..

a) Default Constructor **b) Parameterized constructor**

c) Method overloading **d) Constructor overloading**

a) Default Constructor

```
public class Hello {
   String name;
   //Constructor
Hello() {
      this.name = "BeginnersBook.com";
   }
   public static void main(String[] args) {
      Hello obj = new Hello();
System.out.println(obj.name);
   }
}
```

Output:

BeginnersBook.com

b)Parameterized constructor

```
class Example {
   //Default constructor
Example() {
System.out.println("Default constructor");
   }
   //Parameterized constructor with two integer arguments
Example(int i, int j) {
System.out.println("constructor with Two parameters");
   }
   //Parameterized constructor with three integer arguments
Example(int i, int j, int k) {
System.out.println("constructor with Three parameters");
   }
  //Parameterized constructor with two arguments, int and String
Example(int i, String name) {
System.out.println("constructor with int and String param");
   }
   public static void main(String args[]) {
      //This will invoke default constructor
      Example obj = new Example();
      //This will invoke the constructor with two int parameters
      Example obj2 = new Example(12, 12);
      // This will invoke the constructor with three int parameter
      Example obj3 = new Example(1, 2, 13);
      // This will invoke the constructor with int and String parameter
      Example obj4 = new Example(1, "BeginnersBook");
   }
}
```

Output:

```
Default constructor
constructor with Two parameters
constructor with Three parameters
constructor with int and String parameters
```

c)Method Overloading

```
import java.lang.*;
class Methodoverload
{
        public void demo()
        {
        System.out.println("No parameters");
        }
        public void demo(int a)
        {
        System.out.println("Integer value a is:"+a);
        }
        public void demo(int a,int b)
        {
        System.out.println("Two parameters: a and b values"+a+b);
        }
        public void demo(int a,intb,int c)
        {
        System.out.println("Three parameters: a , b and c values"+a+b+c);
        }
        public static void main(String[] args)
        {
        System.out.println("Method overloading");
        Methodoverload m=new Methodoverload();
        m.demo();
        m.demo(10);
        m.demo(10,20);
        m.demo(10,20,30);
        }
}
```

Output:

```
Method overloading
No parameters
Integer value a is:10
Two parameters: a and b values: 10   20
Three parameters: a, b and c values  10  20  30
```

d)Constructor overloading

```
import java.lang.*;
class Conoverload
{
        public Conoverload()
        {
        System.out.println("Default constructor");
        }
        public Conoverload(int x)
        {
        System.out.println("Area of one parameter :"+(x*x));
        }
        public Conoverload(double x)
        {
        System.out.println("Area of one parameter :"+(x*x));
        }
        public Conoverload(int x,int y)
        {
        System.out.println("Area of two parameters:"+(x*y));
        }
        public Conoverload(int x,double y)
        {
        System.out.println("Area of two parameters:"+(x*y));
        }
        public static void main(String args[])
        {
        System.out.println("constructor overloading");
        Conoverload c1=new Conoverload();
        Conoverload c2=new Conoverload(10);
        Conoverload c3=new Conoverload(20.3);
        Conoverload c4=new Conoverload(20,30);
        Conoverload c5=new Conoverload(40,50.6);
        }
}
```

Output:

```
Constructor overloading
Default constructor
Area of one parameter: 100
Area of one parameter: 12.09000000000003
Area of two parameters: 00
Area of two parameters: 2024.0
Area of two parameters: 4856.0
```

2. Write a Java program
a) checks whether a given string is a palindrome or not.
b) for sorting a given list of names in ascending order.
c) that reads a line of integers and then displays each integer and the sum of all integers (use string tokenizer class of java.util).

a)checks whether a given string is a palindrome or not.

```
import java.lang.*;
import java.io.*;
class StringPal
{
        public static void main(String args[])throws Exception
        {
            BufferedReaderbr=new BufferedReader(new InputStreamReader(System.in));
            System.out.println("Enter a string:");
            String s1=br.readLine();
            char arr1[]=s1.toCharArray();
            char arr2[]=new char[s1.length()];

            for(int i=s1.length()-1,j=0;i>=0;i--,j++)
            arr2[j]=arr1[i];
            String s2=String.valueOf(arr2);
            System.out.println("Reverse of a given string:"+ s2);

            if(s1.equals(s2))
            System.out.println("String entered is a palindrome");
            else
            System.out.println("String entered is not a palindrome");
        }
}
```

Output:

```
Enter a string:  malayalam
String entered is a palindrome
```

b) for sorting a given list of names in ascending order.

```
class sorting
{
   public static void main(String[] input) {
      int k = input.length;
      String temp = new String();
      String names[] = new String[k + 1];
      for (int i = 0; i< k; i++) {
         names[i] = input[i];
      }
      for (int i = 0; i< k; i++)
         for (int j = i + 1; j < k; j++) {
            if (names[i].compareTo(names[j]) < 0) {
               temp = names[i];
               names[i] = names[j];
               names[j] = temp;
            }
         }
System.out.println("Sorted order is");
      for (int i = 0; i< k; i++) {
System.out.println(names[i]);
      }
   }
}
```

Output:

```
Java sorting Harish Ramesh Mahesh Rakesh
Sorted order is
        Ramesh
        Rakesh
        Mahesh
        Harish

Java sorting sai hari teja ravi sandeep
Sorted order is
        teja
        sandeep
        sai
        ravi
        Hari
```

c) Java Program that reads a line of integers and then displays each integer and the sum of all integers (use string tokenizer class of java.util).

```
import java.util.*;
import java.io.*;
class StringTokenSum
{
	public static void main(String args[])
	{
	String str;
	int i,sum=0;
	try
	{
	DataInputStream dis=new DataInputStream(System.in);
	System.out.println("Enter line of Integers");
	str=dis.readLine();
	StringTokenizerst=new StringTokenizer(str);
	while(st.hasMoreTokens())
	{
	String key=st.nextToken();
	System.out.println(key);
	i=Integer.parseInt(key);
	sum=sum+i;
	}
	System.out.println("Sum is : " +sum);
	}
	catch(Exception e)
	{
	System.out.println(e);
	}
	}
}
```

Output:

```
Enter line of Integers
10 20 30 40
10
20
30
40
Sum is : 100
```

3. Write a Java program that uses the following keywords…

a) this **b) super**

c) static **d) final**

a) this

Program-1

```
import java.lang.*;
class ThisKeyword {
        int x;
        public void show(int x){
        x=x;
        }
        public void display(int x){
        this.x=x;
        }
        public static void main(String args[]){
            Thisdemo t1=new Thisdemo();
            t1.display(10);
            System.out.println("t1 value:"+t1.x);
            Thisdemo t2=new Thisdemo();
            t2.show(10);
            System.out.println("t2 value:"+t2.x);
        }
}
```

Output:

```
t1 value: 10
t2  value :0
```

Program-2

```
class Student {
   int rollno;
   String name;
   float fee;
Student(int rollno, String name, float fee) {
this.rollno = rollno;
      this.name = name;
this.fee = fee;
   }
   void display() {
System.out.println(rollno + " " + name + " " + fee);
   }
}
class TestThis2 {
   public static void main(String args[]) {
      Student s1 = new Student(111, "ankit", 5000 f);
      Student s2 = new Student(112, "sumit", 6000 f);
      s1.display();
      s2.display();
   }
}
```

Output:

```
111 ankit 5000
112 sumit 6000
```

b)Super

Program-1

```
import java.lang.*;
class A
{
    int x=10;
}

class B extends A
{
    int x=30;
    public void show()
    {
    System.out.println("x value in B class: "+x);
    System.out.println("x value in A class: "+super.x);
    System.out.println("x+super.x value is"+(x+super.x));
    }
    }

    class Supkeyvar
    {
    public static void main(String args[])
    {
    B1 obj=new B1();
    obj.show();
    }
}
```

Output:

```
x value in B class: 30
x value in A class: 10
x+super.x value is: 40
```

Program-2

```
import java.lang.*;

class A
{
        public void show() {
        System.out.println("A class show() method");
        }
}

class B extends A
{
        public void show() {
        super.show();
        System.out.println("B class  show() method");
        }
}
class Supkeymethod
{
        public static void main(String args[])
        {
        b B=new b();
        B.show();
        }
}
```

Output:

```
A class show() method
B class show() method
```

c) static

Program-1

```
import java.lang.*;
class Staticvar
{
        int x=10;
        static int y=20;
        static double z=30.4;

        public static void main(String args[])
        {
        System.out.println("y value is:"+y);
        Staticvariable S=new Staticvariable();
        System.out.println("x value is :"+S.x);
        System.out.println("z value is :"+S.z);
        }
}
```

Output:

```
y value is:20
x value is :10
z value is :30.4
```

Program-2

```
import java.lang.*;
class Staticmethod
{
        public static void demo()
        {
        System.out.println("Hello 1");
        }

        public void test()
        {
        System.out.println("Hello 2");
        }

        public static void main(String args[])
        {
        demo();
        Staticmethod s=new Staticmethod();
        s.test();
        }
}
```

Output:

```
Hello 1
Hello 2
```

d) final

Program-1

```
import java.lang.*;
class Finalval
{
        public static void main(String args[]) {
                int a=10;
                System.out.println("value of a is:"+a);
                a=30;
                System.out.println("value of a is:"+a);
                final int b=50;
                System.out.println("value of b is:"+b);
        }
}
```

Output:

```
value of a is: 10
value of a is: 30
value of b is: 50
```

Program-2

```
import java.lang.*;
    class A {
            public void show()
            {
            System.out.println("A class show() method");
            }
            final void demo()
            {
            System.out.println("A class demo() method");
            }
    }
    class B extends A {
            public void show()
            {
            System.out.println("B class show() method");
            }
     }
    class Finalmethod {
            public static void main(String args[])
            {
            B b=new B();
            b.show();
            b.demo();
            }
     }
```

Output:

```
B class show() method
A class demo() method
```

Program-3

```
import java.lang.*;
final class demo
{
	public void show()
	{
	System.out.println("Hello Final Class");
	}
}

public class Finalclass
{
	public static void main(String args[])
	{
	demo d=new demo();

	d.show();
	}
}
```

Output:

Hello Final Class

4. Write a Java program to implement
a) Method overloading. **b) dynamic method dispatch.**
c) multiple inheritance. **d) access specifiers.**

a) Method overloading

```
import java.lang.*;
class Methodoverload
{
	public void demo()
	{
	System.out.println("No parameters");
	}

	public void demo(int a)
	{
	System.out.println("Integer value a is:"+a);
	}

	public void demo(int a,int b)
	{
	System.out.println("Two parameters: a and b values"+a+b);
	}

	public void demo(int a,intb,int c)
	{
	System.out.println("Three parameters: a , b and c values"+a+b+c);
	}

	public static void main(String[] args)
	{
	System.out.println("Method overloading");
	Methodoverload m=new Methodoverload();
	m.demo();
	m.demo(10);
	m.demo(10,20);
	m.demo(10,20,30);
	}
}
```

Output:

```
Method overloading
No parameters
Integer value a is:10
Two parameters: a and b values: 10   20
Three parameters: a, b and c values  10  20  30
```

b) dynamic method dispatch.

```
import java.lang.*;
class A
{
    public void show()
    {
    System.out.println("A class method");
    }
}
class B extends A
{
    public void show()
    {
    System.out.println("B class method");
    }
}
class C extends A
{
    public void show()
    {
    System.out.println("C class method");
    }
}
class DynDispatch
{
    public static void main(String args[])
    {
    A a=new A();
    B b=new B();
    C c=new C();
    A r;
    r=a;   r.show();
    r=b;   r.show();
    r=c;   r.show();
    }
}
```

Output:

```
A class method
B class method
C class method
```

c) multiple inheritance

```
Import java.lang.*;
interface Printable
{
   void print();
}
interface Showable
{
   void show();
}
class A7 implements Printable, Showable
{
   public void print()
   {
System.out.println("Hello");
   }
   public void show()
   {
System.out.println("Welcome");
   }
   public static void main(String args[])
   {
      A7 obj = new A7();
obj.print();
obj.show();
   }
}
```

Output :

```
Hello
Welcome
```

d) access specifiers.

```
package xyz;
import abc.AccessDemo;
public class AccessExample {
   public static void main(String[] args) {
AccessDemo ad = new AccessDemo();
ad.testDemo();
   }
}
class AccessDemo {
   private int x = 56;
   public void showDemo() {
System.out.println("The Variable value is " + x);
   }
   private void testDemo() {
System.out.println("It cannot be accessed in another class");
   }
}
public class AccessExample {
   public static void main(String[] args) {
AccessDemo ad = new AccessDemo();
ad.testDemo(); // Private method cannot be used
ad.x = 5; // Private variable cannot be used
ad.showDemo(); // run properly
   }
}
package abc;
class AccessDemo {
   default int a = 4;
}
package xyz;
import abc.AccessDemo;
class AccessExample {
   public static void main(String[] args) {
AccessDemo ad = new AccessDemo();
ad.a = 67; //It is not possible.
   }
}
class AccessDemo {
   protected int x = 34;
   public void showDemo() {
System.out.println("The variable value is " + x);
   }
}
class ChildAccess extends AccessDemo {
   // child class which inherits the properties of AccessDemo class
}
public class AccessExample {
   public static void main(String[] args) {
ChildAccess ca = new ChildAccess();
ca.showDemo(); // run properly
ca.x = 45; // run properly  }  }
```

5. Write a Java program that
a) reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
b) reads a file and displays the file on the screen, with a line number before each line.
c) displays the number of characters, lines and words in a test file.

a) reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.

```
import java.io.*;
class FileDemo
{
        static void P(String s)
        {
        System.out.println(s);
        }

        public static void main(String args[])
        {
        File f=new File("FileDemo.java");
        P("File Name= "+ f.getName());
        P("Path Name= " + f.getPath());
        P("Absolute path= " + f.getAbsolutePath());
        P("Parent= " + f.getParent());
        P("File last modified =" + f.lastModified());
        P("File size in bytes=" + f.length());
        }
}
```

Output:

```
File Name= FileDemo.java
Path Name= FileDemo.java
Absolute path= D:\cse\FileDemo.java
Parent= null
File last modified =1334219624765
File size in bytes=406
```

b) reads a file and displays the file on the screen, with a line number before each line.

```
import java.io.*;
class LineNumber {
   public static void main(String args[]) {
      try {
System.out.println("Enter a file name");
         File f = new File(args[0]);
FileReaderfr = new FileReader(f);
LineNumberReaderlnr = new LineNumberReader(fr);
         String fline;
         while ((fline = lnr.readLine()) != null)
System.out.println(lnr.getLineNumber() + " " + fline);
lnr.close();
      } catch (Exception e) {
System.out.println("Invalid");
      }
   }
}
```

Output:

Program is displayed with line number.

```
1. import java.io.*;
2. class LineNumber
3. {
4.public static void main(String args[])
5. {
6. try
7. {
8. System.out.println("Enter a file name"); ……
```

c) displays the number of characters, lines and words in a test file.

```
import java.io.*;
class CountFile
{
        public static void main(String args[])
        {
        int c=0,w=0,l=0,b=0;
        try
        {
        System.out.println("Enter a file name");
        File f=new File(args[0]);
        FileInputStreamfis=new FileInputStream(f);
        DataInputStream d=new DataInputStream(fis);

        while(( b=fis.read())!=-1)
        {
        c++;
        switch((char) b)
        {
        case '\t' :  w++; break;
        case ' ' :   w++; break;
        case '\n' : l++; break;
        }
        }
        }
        catch(Exception e)
        {
        System.out.println("Invalid") ;
        }
        System.out.println(args[0] + "\nFile Contains\n "+ c +  " characters\n "+ w +
        " words\n" + l +" lines");
        }
 }
```

Output:

```
File Contains
        606 Characters
        64 words
        35 lines
```

6. Write a Java program for handling
a) Checked exceptions
b) unchecked exceptions

a) Checked exceptions.

```
class Ex1
{
	void div()
	{
		int b=10/0;
	}
	void div1()
	{
		div();
	}
	void div2()
	{
		try
		{
			div();
		}
		catch(Exception e)
		{
			System.out.println(e);
		}
	}
	public static void main(String[] args)
	{
		Ex1 e=new Ex1();
		e.div2();
	}
}
```

Output:

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

b) unchecked exceptions.

```
class MyException extends Exception
{
	public String toString()
	{
		return("myException");
	}
}
class MyExceptionDemo
{
	static void compute(int age) throws MyException
	{
		if(age<18)
		throw new MyException();
		else
		System.out.println("elgible");
	}
	public static void main(String[] args)
	{
		try
		{
			compute(24);
			compute(18);
		}
		catch(MyException e)
		{
			System.out.println(e);
		}
	}
}
```

Output:

```
MyException
```

7. Write a Java program

a) Creates three threads. First threads displays "Good Morning "for every one Second, the second thread displays "Hello" for every two seconds, the third thread Displays "Welcome" for every three seconds.

b) That correctly implements producer consumer problem using concept of inter thread communication

a) Creates three threads. First threads displays "Good Morning "for every one Second, the second thread displays "Hello" for every two seconds, the third thread Displays "Welcome" for every three seconds.

```
class A extends Thread
{
    synchronized public void run()
    {
   try
   {
      while(true)
      {
        sleep(1000);
        System.out.println("good morning");
      }
   }
   catch(Exception e)
   { }
    }
}
class B extends Thread
{
     synchronized public void run()
     {
   try
   {
      while(true)
      {
      sleep(2000);
      System.out.println("hello");
      }
      }
     catch(Exception e)
   { }
     }
}
class C extends Thread
{
    synchronized public void run()
    {
   try
   {
```

```
        while(true)
        {
            sleep(3000);
            System.out.println("welcome");
        }
    }
    catch(Exception e)
    { }
     }
}
class ThreadDemo
{
    public static void main(String args[])
    {
        A t1=new A();
        B t2=new B();
        C t3=new C();
        t1.start();
        t2.start();
        t3.start();
    }
}
```

Output:

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

b) That correctly implements producer consumer problem using concept of inter thread communication

```
class Q
{
	int n;
	boolean valueSet=false;
	synchronized int get()
	{
		if(!valueSet)
		try
		{
			wait();
		}
		catch(InterruptedException e)
		{
			System.out.println("Interrupted Exception caught");
		}
		System.out.println("Got:"+n);
		valueSet=false;
		notify();
		return n;
	}
	synchronized void put(int n)
	{
		if(valueSet)
		try
		{
			wait();
		}
		catch(InterruptedException e)
		{
			System.out.println("Interrupted Exception caught");
		}
		this.n=n;
		valueSet=true;
		System.out.println("Put:"+n);
		notify();
	}
}
class Producer implements Runnable
{
	Q q;
	Producer(Q q)
	{
		this.q=q;
		new Thread(this,"Producer").start();
	}
	public void run()
	{
		int i=0;
		while(true)
		{
			q.put(i++);
		}
```

```
        }
}
class Consumer implements Runnable
{
        Q q;
        Consumer(Q q)
        {
                this.q=q;
                new Thread(this,"Consumer").start();
        }
        public void run()
        {
                while(true)
                {
                        q.get();
                }
        }
}
class ProdCons
{
        public static void main(String[] args)
        {
                Q q=new Q();
                new Producer(q);
                new Consumer(q);
                System.out.println("Press Control-c to stop");
        }
}
```

Output:

```
Put:0
Press Control-c to stop
Got:0
Put:1
Got:1
Put:2
Got:2
Put:3
Got:3
Put:4
Got:4
Put:5
Got:5
Put:6
Got:6
```

8. Write a Java program which demonstrates the use of following collection classes
a) Array List
b) Hash Set
c) Deque

a) Array List

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

    // create ArrayList
    ArrayList<String> languages = new ArrayList<>();

    // Add elements to ArrayList
    languages.add("Java");
    languages.add("Python");
    languages.add("R Language");
    System.out.println("ArrayList: " + languages);
  }
}
```

Output:

ArrayList: Java Python R Language

b) Hash Set

```
import java.util.*;
class HashSetDemo
{
  public static void main(String args[])
  {
    // Creating HashSet
    HashSet<String> hs = new HashSet<String>();
    // Adding elements
    hs.add("Ramu");
    hs.add("Mohan");
    hs.add("Raju");
    hs.add("Mohan");
    // Displaying HashSet
    System.out.println("HashSet:"+ hs);
  }
}
```

Output:

HashSet: Ramu Mohan Raju Mohan

c) Deque

```
//Implementation of Deque in ArrayDeque Class
import java.util.Deque;
import java.util.ArrayDeque;

class DequeDemo
{
   public static void main(String[] args)
   {
      // Creating Deque using the ArrayDeque class
      Deque<Integer> numbers = new ArrayDeque<>();

      // add elements to the Deque
      numbers.offer(1);
      numbers.offerLast(2);
      numbers.offerFirst(3);
      System.out.println("Deque: " + numbers);

      // Access elements of the Deque
      int firstElement = numbers.peekFirst();
      System.out.println("First Element: " + firstElement);

      int lastElement = numbers.peekLast();
      System.out.println("Last Element: " + lastElement);

      // Remove elements from the Deque
      int removedNumber1 = numbers.pollFirst();
      System.out.println("Removed First Element: " + removedNumber1);

      int removedNumber2 = numbers.pollLast();
      System.out.println("Removed Last Element: " + removedNumber2);

      System.out.println("Updated Deque: " + numbers);
   }
}
```

Output:

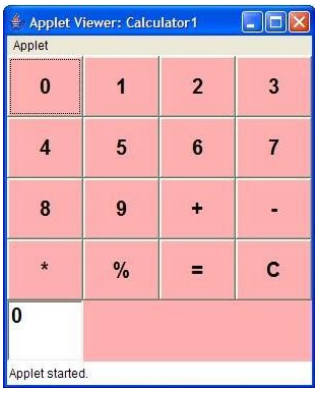
```
Deque: [3, 1, 2]
First Element: 3
Last Element: 2
Removed First Element: 3
Removed Last Element: 2
Updated Deque: [1]
```

9. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +,-,*,/ operations. Add a text field to display the result.

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet  code="Calculator1" width=300 height=300></applet>*/
public class Calculator1 extends Applet implements ActionListener
{
        TextField t;
        Button b[]=new Button[15];
        Button b1[]=new Button[6];
        String op2[]={"+","-","*","%","=","C"};
        String str1="";
        int p=0,q=0;
        String oper;
        public void init()
        {
                setLayout(new GridLayout(5,4));
                t=new TextField(20);
                setBackground(Color.pink);
                setFont(new Font("Arial",Font.BOLD,20));
                int k=0;
                t.setEditable(false);
                t.setBackground(Color.white);
                t.setText("0");
                for(int i=0;i<10;i++)
                {
                        b[i]=new Button(""+k);
                        add(b[i]);
                        k++;
                        b[i].setBackground(Color.pink);
                        b[i].addActionListener(this);
                }

                for(int i=0;i<6;i++)
                {
                        b1[i]=new Button(""+op2[i]);
                        add(b1[i]);
                        b1[i].setBackground(Color.pink);
                        b1[i].addActionListener(this);
                }
                add(t);
        }
        public void actionPerformed(ActionEvent ae)
        {

                String  str=ae.getActionCommand();

                if(str.equals("+")){      p=Integer.parseInt(t.getText());
                                          oper=str;
                                          t.setText(str1="");
                                   }
                else if(str.equals("-")){  p=Integer.parseInt(t.getText());
                                           oper=str;
                                           t.setText(str1="");
                                          }
```

```
                else if(str.equals("*")){  p=Integer.parseInt(t.getText());
                                           oper=str;
                                          t.setText(str1="");
                                        }
                else if(str.equals("%")){  p=Integer.parseInt(t.getText());
                                           oper=str;

                                           t.setText(str1="");
                                      }
                else if(str.equals("=")) { str1="";
                                           if(oper.equals("+"))  {
                                           q=Integer.parseInt(t.getText());
                                           t.setText(String.valueOf((p+q)));}

                                           else if(oper.equals("-")) {
                                           q=Integer.parseInt(t.getText());
                                           t.setText(String.valueOf((p-q))); }

                                           else  if(oper.equals("*")){
                                           q=Integer.parseInt(t.getText());
                                           t.setText(String.valueOf((p*q))); }

                                           else if(oper.equals("%")){
                                           q=Integer.parseInt(t.getText());
                                           t.setText(String.valueOf((p%q))); }
                                      }

                else if(str.equals("C")){  p=0;q=0;
                                           t.setText("");
                                           str1="";
                                           t.setText("0");
                                         }


        else{  t.setText(str1.concat(str));
                                 str1=t.getText();
                               }

                    }

}
```

Output:



10. Write a Java program for handling
a) Key Events. b) Mouse Events.

a) Key Events.

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="Keyevents"  width=300 height=400></applet>*/
public class Keyevents extends Applet implements KeyListener
{
        String msg=" ";
        int  x=10,y=20;
        public void init(){
                addKeyListener(this);
                requestFocus(true);
        }
        public void keyPressed(KeyEventke){
                showStatus(  "keyPressed");
                repaint();
        }
        public void keyReleased(KeyEventke){
        showStatus("keyReleased");
        repaint();
        }
        public void keyTyped(KeyEventke){
        char  ch=ke.getKeyChar();
          if(ch=='M'||ch=='m')
             msg="Good Morning";
          else if(ch=='A'||ch=='a')
             msg="Good Afternoon";
          else if(ch=='E'||ch=='e')
             msg="Good Evening";
        repaint();
        }
        public void paint(Graphics g){
        g.drawString(msg,x,y);
        }
}
```

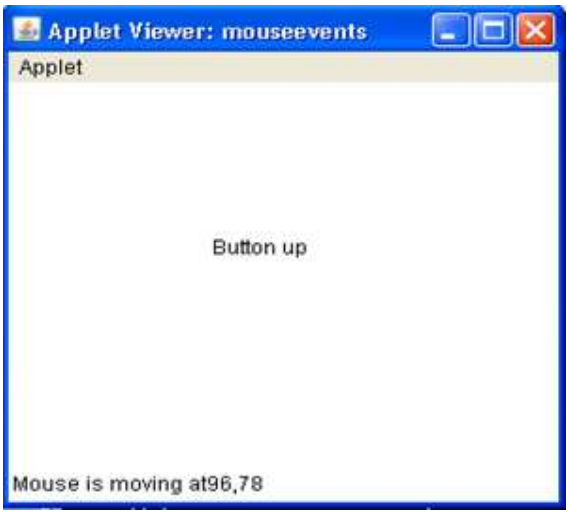
Output:

b) Mouse Events.

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/* <applet code="Mouseevents" width=300 height=400></applet>*/
public class Mouseevents extends Applet implements
MouseListener,MouseMotionListener
{
String msg=" ";
int x=0,y=0;
public void init()
{
addMouseListener(this);
addMouseMotionListener(this);
}
public void mouseEntered(MouseEvent me) {
x=0;y=10;
msg=" Mouse is Entered";
repaint();
}
public void mouseExited(MouseEvent me) {
x=0;y=10;
msg="Mouse is Exited";
repaint();
}
public void mousePressed(MouseEvent me) {
x=me.getX();
y=me.getY();
msg="Button Down";
repaint();
}
public void mouseReleased(MouseEvent me) {
x=me.getX();
y=me.getY();
msg="Button up";
repaint();
}
public void mouseClicked(MouseEvent me)
{
x=0;y=10;
msg="Mouse is Clicked";
repaint();
}
public void mouseMoved(MouseEvent me)
{
showStatus("Mouse is moving at" +me.getX()+","+me.getY());
repaint();
}
```

```
public void mouseDragged(MouseEvent me)
{
x=me.getX();
y=me.getY();
msg=" ***";
showStatus("Mouse isDragged at" +me.getX()+","+me.getY());
repaint();
}

public void paint(Graphics g)
{
g.drawString(msg,x,y);
}
}
```

Output:



11. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields num1 and num2. The division of num1 and num2 is displayed on the result field when the divide button is clicked. If num1 and num2 were not an integer, the program would throw number format exception. If num2 was zero , the program would throw an arithmetic exception and display the exception in message dialogue box.

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="Div"width=230 height=250>
</applet>*/
public class Div extends Applet implements ActionListener
{
String msg;
TextField num1,num2,res;Label l1,l2,l3;
Button div;
public void init()
{
l1=new Label("Number 1");
l2=new Label("Number 2");
l3=new Label("result");
num1=new TextField(10);
num2=new TextField(10);
res=new TextField(10);
div=new Button("DIV");
div.addActionListener(this);
add(l1);
add(num1);
add(l2);
add(num2);
add(l3);
add(res);
add(div);
}
public void actionPerformed(ActionEvent ae)
{
String arg=ae.getActionCommand();
if(arg.equals("DIV"))
{
String s1=num1.getText();
String s2=num2.getText();
int num1=Integer.parseInt(s1);
int num2=Integer.parseInt(s2);
if(num2==0)
{
try
{
System.out.println(" ");
}
catch(Exception e)
{
```

```
System.out.println("ArithematicException"+e);
}
msg="Arithemetic";
repaint();
}
else if((num1<0)||(num2<0))
{
try
{
System.out.println("");
}
catch(Exception e)
{
System.out.println("NumberFormat"+e);
}
msg="NumberFormat";
repaint();
}
else
{
int num3=num1/num2;
res.setText(String.valueOf(num3));
}
}
}
public void paint(Graphics g)
{
g.drawString(msg,30,70);
}
}
```

Output

12. Write a JAVA program that

a) Simulate traffic light. The program lets the user select one of three lights :red green or yellow. When a radio button is selected, the light is turned on and only one light can be on at a time. No light is on when the program starts.

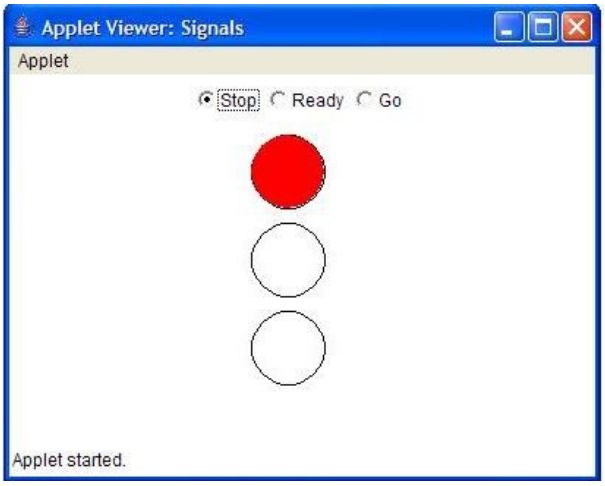
b) allows the user to draw lines rectangles and ovals.

a) Simulate traffic light. The program lets the user select one of three lights :red green or yellow. When a radio button is selected, the light is turned on and only one light can be on at a time. No light is on when the program starts.

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code="Signals" width=400 height=250></applet>*/
public class Signals extends Applet implements ItemListener
{
	String msg="";
	Checkbox  stop,ready,go;
	CheckboxGroup cbg;
	public void init()
	{
		cbg = new CheckboxGroup();
		stop = new Checkbox("Stop", cbg, false);
		ready = new Checkbox("Ready", cbg, false);
		go= new Checkbox("Go", cbg, false);
		add(stop);
		add(ready);
		add(go);
		stop.addItemListener(this);
		ready.addItemListener(this);
		go.addItemListener(this);

	}

	public void itemStateChanged(ItemEvent ie)
	{
		repaint();
	}

	public void paint(Graphics g)
	{

		msg=cbg.getSelectedCheckbox().getLabel();
		g.drawOval(165,40,50,50);
		g.drawOval(165,100,50,50);
		g.drawOval(165,160,50,50);



	if(msg.equals("Stop"))
		{
			g.setColor(Color.red);
			g.fillOval(165,40,50,50);
		}
		else if(msg.equals("Ready"))
		{
```

```
                    g.setColor(Color.yellow);
                    g.fillOval(165,100,50,50);
                }
                else
                {
                    g.setColor(Color.green);
                    g.fillOval(165,160,50,50);
                }

        }
}
```

Output:



b) allows the user to draw lines rectangles and ovals.

```
import java.awt.*;
import java.applet.*;
/*
<applet code="Sujith" width=200 height=200>
</applet>
*/
public class Sujith extends Applet
{
	public void paint(Graphics g)
	{
		for(int i=0;i<=250;i++)
		{
			Color c1=new Color(35-i,55-i,110-i);
			g.setColor(c1);
			g.drawRect(250+i,250+i,100+i,100+i);
			g.drawOval(100+i,100+i,50+i,50+i);
			g.drawLine(50+i,20+i,10+i,10+i);
		}
	}
}
```

Output: