

TIRUMALA ENGINEERING COLLEGE

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

An ISO 9001:2015 Certified Institution

Jonnalagadda(V), Narasaraopet 522601. Guntur District.



II B. Tech., CSE I I- Semester

LABORATORY MANUAL

For

JAVA PROGRAMMING LAB [R20]

Prepared by

Mr. A.BALAJI, Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course Objectives:

The aim of this lab is to

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

Course Outcomes:

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

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

Exercise - 1 (Basics)

- a) Write a JAVA program to display default value of all primitive data type of JAVA
- b) Write a java program that display the roots of a quadratic equation $ax^2+bx=0$ Calculate the discriminate D and basing on value of D, describe the nature of root.
- c) Five Bikers Compete in a race such that they drive at a constant speed which may or maynot be the same as the other. To qualify the race, the speed of a racer must be more than theaverage speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers.

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

- a) Write a JAVA program to search for an element in a given list of elements using binary search mechanism.
- b) Write a JAVA program to sort for an element in a given list of elements using bubble sort
- c) Write a JAVA program to sort for an element in a given list of elements using merge sort.
- d) Write a JAVA program using StringBuffer to delete, remove character.

Exercise - 3 (Class, Objects)

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

Exercise - 4 (Methods)

- a) Write a JAVA program to implement constructor overloading.
- b) Write a JAVA program implement method overloading.

Exercise - 5 (Inheritance)

- a) Write a JAVA program to implement Single Inheritance
- b) Write a JAVA program to implement multi level Inheritance
- c) Write a java program for abstract class to find areas of different shapes

Exercise - 6 (Inheritance - Continued)

- a) Write a JAVA program give example for “super” keyword.
- b) Write a JAVA program to implement Interface. What kind of Inheritance can be achieved?

Exercise - 7 (Exception)

- a) Write a JAVA program that describes exception handling mechanism
- b) Write a JAVA program Illustrating Multiple catch clauses

Exercise – 8 (Runtime Polymorphism)

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

Exercise – 9 (User defined Exception)

- a) Write a JAVA program for creation of Illustrating throw
- b) Write a JAVA program for creation of Illustrating finally
- c) Write a JAVA program for creation of Java Built-in Exceptions
- d) d)Write a JAVA program for creation of User Defined Exception

Exercise – 10 (Threads)

- a) Write a JAVA program that creates threads by extending Thread class .First thread display “Good Morning “every 1 sec, the second thread displays “Hello “every 2 seconds and the third display “Welcome” every 3 seconds ,(Repeat the same by implementing Runnable)
- b) Write a program illustrating isAlive and join ()
- c) Write a Program illustrating Daemon Threads.

Exercise - 11 (Threads continuity)

- a) Write a JAVA program Producer Consumer Problem

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

Exercise – 12 (Packages)

a) Write a JAVA program illustrate class path

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

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

Exercise - 13 (Applet)

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

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

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

Exercise - 14 (Event Handling)

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

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

EXPERIMENT - I

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

Source Code:

```
classDefaultValues

{

byte b;

short s;

int i;

long l;

float f;

double d;

char c;

booleanbl;

public static void main(String[] args)

{

DefaultValues DV=new DefaultValues();

System.out.println("The default values of primitive data types are:");

System.out.println("Byte :"+DV.b);

System.out.println("Short :"+DV.s);

System.out.println("Int :"+DV.i);

System.out.println("Long :"+DV.l);

System.out.println("Float :"+DV.f);

System.out.println("Double :"+DV.d);

System.out.println("Char :"+DV.c);

System.out.println("Boolean :"+DV.bl);
```

```
}  
  
}
```

OutPut:

F:\Java Lab CSE 2-2>javac DefaultValues.java

F:\Java Lab CSE 2-2>java DefaultValues

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

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.

Source Code:

```
import java.util.*;  
  
class Quadratic  
{  
  
    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:

F:\Java Lab CSE 2-2>javac Quadratic.java

F:\Java Lab CSE 2-2>java Quadratic

Given quadratic equation: $ax^2 + bx + c$

Enter a:24

Enter b:6

Enter c:2

Roots are imaginary

F:\Java Lab CSE 2-2>java Quadratic

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

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

Source Code:

```
import java.util.*;
```



```
class Bikerace
{
    public static void main(String[] args)
    {
        float s1,s2,s3,s4,s5,average;

        Scanner s = new Scanner(System.in);

        System.out.println("Enter speed of first racer:");

        s1 = s.nextFloat();

        System.out.println("Enter speed of second racer:");

        s2 = s.nextFloat();

        System.out.println("Enter speed of third racer:");

        s3 = s.nextFloat();

        System.out.println("Enter speed of fourth racer:");

        s4 = s.nextFloat();

        System.out.println("Enter speed of fifth racer:");

        s5 = s.nextFloat();

        average=(s1+s2+s3+s4+s5)/5;

        if(s1>average)

            System.out.println("First racer is qualify racer:");

        if(s2>average)

            System.out.println("Second racer is qualify racer:");

        if(s3>average)

            System.out.println("Third racer is qualify racer:");

        if(s4>average)

            System.out.println("Fourth racer is qualify racer:");
```

```
if(s5>average)
System.out.println("Fifth racer is qualify racer:");
}
}
```

OutPut:

F:\Java Lab CSE 2-2>javac Bikerace.java

F:\Java Lab CSE 2-2>java Bikerace

Enter speed of first racer:

10

Enter speed of second racer:

90

Enter speed of third racer:

120

Enter speed of fourth racer:

100

Enter speed of fifth racer:

90

Second racer is qualify racer:

Third racer is qualify racer:

Fourth racer is qualify racer:

Fifth racer is qualify racer:

EXPERIMENT - II

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

Source Code:

```
import java.util.Scanner;

class Binarysearch
{
    public static void main(String args[])
    {
        int n, i, num, first, last, middle, flag;
        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;
        flag = 0;

        while (first <= last)
        {
            middle = (first + last) / 2;

            if (a[middle] == num)
```

```
{  
flag=1;  
System.out.println("The search element is : "+a[middle]);  
break;  
}  
if (a[middle] < num )  
first = middle + 1;  
if(a[middle]>num)  
last = middle - 1;  
}  
if ( flag==1)  
{  
System.out.println( "The Number is found ");  
}  
else  
System.out.println("The Number is not found");  
}  
}
```

output:

F:\Java Lab CSE 2-2>javac Binarysearch.java

F:\Java Lab CSE 2-2>java Binarysearch

Enter total number of elements:

5

JAVA LAB MANUAL

Enter elements in sorted order:

7

8

10

20

40

Enter the search value:

10

The search element is : 10

The Number is found

F:\Java Lab CSE 2-2>java Binarysearch

Enter total number of elements:

5

Enter elements in sorted order:

1

2

3

4

5

Enter the search value:

8

The Number is not found

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

Source Code:

```
import java.util.Scanner;

class Bubblesort
{
    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 = n - 1; i > 0; 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(a[i]+"\\t");  
  
}  
  
}
```

Output:

F:\Java Lab CSE 2-2>javac Bubblesort.java

F:\Java Lab CSE 2-2>java Bubblesort

Enter total number of elements:

10

Enter elements:

24

45

10

23

67

15

24

76

34

29

The sorted elements are:

10 15 23 24 24 29 34 45 67 76

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

Source Code:

```
import java.util.*;

class Mergesort
{
    void mergesort(int arr[], int l, int r)
    {
        if(l < r)
        {
            int m = (l+r)/2;
            mergesort(arr, l, m);
            mergesort(arr, m+1, r);
            merge(arr, l, m, r);
        }
    }

    void merge(int arr[], int l, int m, int r)
    {
        int n1 = m - l + 1;
        int n2 = r - m;
        int L[] = new int[n1];
        int M[] = new int[n2];

        // fill the left and right array
        for (int i = 0; i < n1; i++)
            L[i] = arr[l + i];
```



```
for (int j = 0; j < n2; j++)
```

```
    M[j] = arr[m + 1 + j];
```

```
int i, j, k;
```

```
i = 0;
```

```
j = 0;
```

```
k = l;
```

```
while (i < n1 && j < n2)
```

```
{
```

```
    if (L[i] <= M[j])
```

```
    {
```

```
        arr[k] = L[i];
```

```
        i++;
```

```
    }
```

```
    else
```

```
    {
```

```
        arr[k] = M[j];
```

```
        j++;
```

```
    }
```

```
    k++;
```

```
}
```

```
while (i < n1)
```

```
{
```

```
    arr[k] = L[i];
```

```
    i++;
```

```
    k++;
```

```
}  
  
while (j < n2)  
{  
    arr[k] = M[j];  
    j++;  
    k++;  
}  
  
public static void main(String []args)  
{  
    int n,i;  
    int arr[]=new int[50];  
    Scanner S=new Scanner(System.in);  
    Mergesort M=new Mergesort();  
    System.out.println("Enter total number of elements");  
    n=S.nextInt();  
    System.out.println("Enter array elements");  
    for(i=0;i<n;i++)  
        arr[i]=S.nextInt();  
    M.mergesort(arr,0,n-1);  
    System.out.println("The sorted elements");  
    for(i=0;i<n;i++)  
        System.out.print(arr[i]+"\\t");  
}  
}
```

JAVA LAB MANUAL

Output:

F:\Java Lab CSE 2-2>javac Mergesort.java

F:\Java Lab CSE 2-2>java Mergesort

Enter total number of elements

10

Enter array elements

9

7

10

3

4

1

5

7

2

8

The sorted elements

1 2 3 4 5 7 7 8 9 10

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

Source Code:

```
import java.util.*;
```

```
class StringBuffer
```

```
{
```

```
public static void main(String[] args)
```

```
{  
Scanner S=new Scanner(System.in);  
StringBuffer str1=new StringBuffer();  
int n;  
System.out.println("Enter a string");  
str1.append(S.next());  
System.out.println("Enter how many character will delete");  
n=S.nextInt();  
str1.delete(0,n);  
System.out.println(str1);  
StringBuffer str2=new StringBuffer();  
System.out.println("Enter a string");  
str2.append(S.next());  
System.out.println("Enter character position to delete");  
n=S.nextInt();  
str2.deleteCharAt(n);  
System.out.println(str2);  
}  
}
```

Output:

F:\Java Lab CSE 2-2>javac StringBuffer.java

F:\Java Lab CSE 2-2>java StringBuffer

Enter a string

GopalaKrishna

Enter how many character will delete

6

Krishna

Enter a string

RamaSSwamy

Enter character position to delete

5

RamaSwamy

EXPERIMENT - III

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

Source Code:

```
import java.util.*;
class BankAccount
{
    String name;
    static int AcID=0;
    int ID;
    double bal=0.0, dep=0.0, wd;
    Scanner S=new Scanner(System.in);
    BankAccount(String name)
    {
        AcID++;
        ID=AcID;
        this.name=name.toUpperCase();
    }
    void deposit()
    {
        System.out.println("Enter Deposit Amount");
        dep=S.nextDouble();
        bal=bal+dep;
        System.out.println("Deposit Amount Successfully");
        checkbal();
    }
    void withdraw()
    {
        System.out.println("Enter Withdraw Amount");
        wd=S.nextDouble();
        bal=bal-wd;
        System.out.println("Withdraw Amount Successfully");
        checkbal();
    }
    void checkbal()
    {
        System.out.println("Current Balance: "+bal);
    }
}
```

```
{
System.out.println(" Account ID:"+ID);
System.out.println(" Account Owner name :"+name);
System.out.println(" Available Balance :"+bal);
}
}
class Main
{
public static void main(String args[])
{
intch;
BankAccountJhon=new BankAccount("Jhon Abraham");
System.out.println("1.Deposit\n2.Withdraw\n3.CheckBalance");
System.out.println("Enter your choice");
Scanner SC=new Scanner(System.in);
ch=SC.nextInt();
switch(ch)
{
case 1:
    Jhon.deposit();
    break;
case 2:
    Jhon.withdraw();
    break;
case 3:
    Jhon.checkbal();
    break;
default:
    System.out.println("Wrong Choice");
}
}
}
```

output:

```
D:\Balaji>javac Main.java
```

```
D:\Balaji>java Main
```

```
1.Deposit
```

```
2.Withdraw
```

```
3.CheckBalance
```

```
Enter your choice
```

```
3
```

```
Account ID:1
```

```
Account Owner name :JHON ABRAHAM
```

```
Available Balance :0.0
```

```
D:\Balaji>java Main
```

```
1.Deposit
```

```
2.Withdraw
```

```
3.CheckBalance
```

```
Enter your choice
```

```
1
Enter Deposit Amount
10000
Deposit Amount Successfully
Account ID:1
Account Owner name :JHON ABRAHAM
Available Balance :10000.0
```

```
D:\Balaji>java Main
1.Deposit
2.Withdraw
3.CheckBalance
Enter your choice
2
Enter Withdraw Amount
5000
withdraw Amount Successfully
Account ID:1
Account Owner name :JHON ABRAHAM
Available Balance :-5000.0
```

B) Write a JAVA program to implement constructor.

Source Code:

```
class Box
{
double width;
double height;
double depth;
Box(double w, double h, double d)
{
width=w;
height=h;
depth=d;
}
double volume()
{
return (width * height * depth);
}
}
class Volume
{
public static void main(String args[])
{
Box B1 = new Box(5,6,8);
Box B2 = new Box(3,4,2);
System.out.println("Using Constructor...");
double vol1 = B1.volume();
System.out.println("Volume of Box1 :: "+vol1);
double vol2 = B2.volume();
System.out.println("Volume of Box2 :: "+vol2);
}
}
```

Output:

```
D:\Balaji>javac Volume.java
```

```
D:\Balaji>java Volume
Using Constructor...
Volume of Box1 :: 240.0
Volume of Box2 :: 24.0
```

EXPERIMENT - IV

A) Write a JAVA program to implement constructor overloading

Source Code:

```
class Box
{
    double width, height, depth;

    Box(double w, double h, double d)
    {
        width = w;
        height = h;
        depth = d;
    }

    Box()
    {
        width = height = depth = 0;
    }

    Box(double len)
    {
        width = height = depth = len;
    }

    double volume()
    {
        return width * height * depth;
    }
}
class ConstructorOverloading
{
    public static void main(String args[])
    {
        Box mybox1 = new Box(10, 20, 15);
        Box mybox2 = new Box();
        Box mycube = new Box(7);

        double vol;

        vol = mybox1.volume();
        System.out.println(" Volume of mybox1 is " + vol);
```



```
vol = mybox2.volume();
System.out.println(" Volume of mybox2 is " + vol);

vol = mycube.volume();
System.out.println(" Volume of mycube is " + vol);
    }
}
```

Output:

```
D:\Balaji>javac ConstructorOverloading.java
```

```
D:\Balaji>java ConstructorOverloading
Volume of mybox1 is 3000.0
Volume of mybox2 is 0.0
Volume of mycube is 343.0
```

B) Write a JAVA program implement method overloading.

Source Code:

```
class Method
{
float add(float a,float b)
{
return(a+b);
}
float add(int a, float b)
{
return(a+b);
}
float add(float a, int b)
{
return(a+b);
}
int add(int a, int b)
{
return(a+b);
}
float add(int a)
{
return(a+a);
}
float add(float b)
{
return(b+b);
}
}
```

```
class Overloading
{
public static void main(String args[])
{
Method M=new Method();
System.out.println("The addition =" +M.add(10.5f,12.7f));
System.out.println("The addition =" +M.add(10,12.5f));
System.out.println("The addition =" +M.add(10.5f,12));
System.out.println("The addition =" +M.add(10,12));
System.out.println("The addition =" +M.add(10,12));
System.out.println("The addition =" +M.add(10));
System.out.println("The addition =" +M.add(10.6f));
}
}
```

Output:

```
D:\Balaji>java Overloading
The addition =23.2
The addition =22.5
The addition =22.5
The addition =22
The addition =22
The addition =20.0
The addition =21.2
```

EXPERIMENT – V

A) Write a JAVA program to implement Single Inheritance

Source Code:

```
import java.util.*;
class Father
{
String fname, fjob, faddress;
int fage;
Scanner S=new Scanner(System.in);
void fdetails()
{
System.out.println("Enter Father name");
fname=S.nextLine();
System.out.println("Enter Father job");
fjob=S.nextLine();
System.out.println("Enter Father Address");
faddress=S.nextLine();
System.out.println("Enter Father Age");
fage=S.nextInt();
}
void fdisplay()
{
System.out.println("*****Father Details*****");
}
```

JAVA LAB MANUAL

```
System.out.println("Father name  :"+fname);
System.out.println(" Father job  :"+fjob);
System.out.println("Father address  :"+faddress);
System.out.println("Father Age    :"+fage);
}
}
class Child extends Father
{
String cname, cjob, caddress;
int cage;
Scanner S=new Scanner(System.in);
void cdetails()
{
System.out.println("Enter Chaild name");
cname=S.nextLine();
System.out.println("Enter Chaild job");
cjob=S.nextLine();
System.out.println("Enter ChaildAdress");
caddress=S.nextLine();
System.out.println("Enter Chaild Age");
cage=S.nextInt();
}
void cdisplay()
{
System.out.println("*****Chaild Details*****");
System.out.println("Chaild name  :"+cname);
System.out.println(" Chaild job   :"+cjob);
System.out.println("Chaild address  :"+caddress);
System.out.println("Chaild Age    :"+cage);
}
}
class Single
{
public static void main(String args[])
{
Child Son=new Child();
Son.fdetails();
Son.cdetails();
Son.fdisplay();
Son.cdisplay();
}
}
```

Output:

F:\Java Lab CSE 2-2>javac Single.java

F:\Java Lab CSE 2-2>java Single

Enter Father name

SambasivaRao

Enter Father job

Farmar

Enter Father Adress

Narasarao pet

```
Enter Father Age
52
Enter Chaild name
Abhiram
Enter Chaild job
Software
Enter ChaildAdress
Hyderabad
Enter Chaild Age
26
*****Father Details*****
Father name :SambasivaRao
Father job :Farmer
Father address :Narasarao pet
Father Age :52
*****Chaild Details*****
Chaildname :Abhiram
Chaildjob :Software
Chaild address :Hyderabad
Chaild Age :26
```

B) Write a JAVA program to implement multi-level Inheritance

Source Code

```
import java.util.*;
class GrandFather
{
String gname, gjob, gaddress;
int gage;
Scanner S=new Scanner(System.in);
void gdetails()
{
System.out.println("Enter GrandFather name");
gname=S.nextLine();
System.out.println("Enter GrandFather job");
gjob=S.nextLine();
System.out.println("Enter GrandFatherAdress");
gaddress=S.nextLine();
System.out.println("Enter GrandFather Age");
gage=S.nextInt();
}
void gdisplay()
{
System.out.println("*****GrandFather Details*****");
System.out.println("GrandFather name :"+gname);
System.out.println("GrandFather job :"+gjob);
System.out.println("GrandFather address :"+gaddress);
System.out.println("GrandFather Age :"+gage);
}
}
```

```
class Father extends GrandFather
{
String fname, fjob, faddress;
intfage;
Scanner S=new Scanner(System.in);
voidfdetails()
{
System.out.println("Enter Father name");
fname=S.nextLine();
System.out.println("Enter Father job");
fjob=S.nextLine();
System.out.println("Enter Father Adress");
faddress=S.nextLine();
System.out.println("Enter Father Age");
fage=S.nextInt();
}
voidfdisplay()
{
System.out.println("*****Father Details*****");
System.out.println("Father name  :"+fname);
System.out.println(" Father job   :"+fjob);
System.out.println("Father address  :"+faddress);
System.out.println("Father Age    :"+fage);
}
}
class Child extends Father
{
String cname, cjob, caddress;
int cage;
Scanner S=new Scanner(System.in);
voidcdetails()
{
System.out.println("Enter Chaild name");
cname=S.nextLine();
System.out.println("Enter Chaild job");
cjob=S.nextLine();
System.out.println("Enter ChaildAdress");
caddress=S.nextLine();
System.out.println("Enter Chaild Age");
cage=S.nextInt();
}
voidcdisplay()
{
System.out.println("*****Chaild Details*****");
System.out.println("Chaild name   :"+cname);
System.out.println(" Chaild job    :"+cjob);
System.out.println("Chaild address  :"+caddress);
System.out.println("Chaild Age     :"+cage);
}
}
class Multilevel
{
public static void main(String args[])
{

```

```
Child Son=new Child();
Son.gdetails();
Son.fdetails();
Son.cdetails();
Son.gdisplay();
Son.fdisplay();
Son.cdisplay();
}
}
```

output:

```
F:\Java Lab CSE 2-2>javac Multilevel.java
```

```
F:\Java Lab CSE 2-2>java Multilevel
Enter GrandFathername
Apparao
Enter GrandFatherjob
Clerk
Enter GrandFatherAdress
Narasarao pet
Enter GrandFather Age
89
Enter Father name
Venkateswarao
Enter Father job
Business
Enter Father Adress
Narasarao pet
Enter Father Age
45
Enter Chaild name
Prabhash
Enter Chaild job
Actor
Enter ChaildAdress
Hyderabad
Enter Chaild Age
26
*****GrandFather Details*****
GrandFathername :Apparao
GrandFatherjob :Clerk
GrandFather address :Narasarao pet
GrandFather Age :89
*****Father Details*****
Father name :Venkateswarao
Father job :Business
Father address :Narasarao pet
Father Age :45
*****Chaild Details*****
Chaildname :Prabhash
Chaildjob :Actor
Chaild address :Hyderabad
Chaild Age :26
```

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

Source Code:

```
import java.lang.Math;
abstract class Shape
{
    abstract void area();
    double area;
}
class Rectangle extends Shape
{
    double w=50,h=25;
    void area()
    {
        area = w*h;
        System.out.println("Area of Rectangle : "+area);
    }
}
class Square extends Shape
{
    double a=40;
    void area()
    {
        area = (a*a);
        System.out.println("Area of Square : "+area);
    }
}
class Circle extends Shape
{
    double r=7;
    void area()
    {
        area = Math.PI * r * r;
        System.out.println("Area of Circle : "+area);
    }
}
class Area
{
    public static void main(String [] args)
    {
        Square sq= new Square();
        Rectangle rc =new Rectangle();
        Circle cr =new Circle();
        sq.area();
        rc.area();
        cr.area();
    }
}
```

Output:

```
F:\Java Lab CSE 2-2>javac Area.java
```

```
F:\Java Lab CSE 2-2>java Area
```

```
Area of Square : 1600.0
```

```
Area of Rectangle : 1250.0
```

```
Area of Circle : 153.93804002589985
```

EXPERIMENT - VI

A) Write a JAVA program give example for "super" keyword.

Source Code:

```
class Rectangle
{
    private double length;
    private double width;
    public Rectangle()
    {
        length = 0;
        width = 0;
    }
    public Rectangle(double length, double width)
    {
        this.length = length;
        this.width = width;
    }
    public double getArea()
    {
        return length * width;
    }
}
class Box extends Rectangle
{
    private double height;
    public Box()
    {
        super();
        height = 0;
    }
    public Box(double length, double width, double height)
    {
        super(length, width);
        this.height = height;
    }

    public double getVolume()
    {
        return getArea() * height;
    }
}
```



```
}
public class Area
{
    public static void main(String[] args)
    {
        Box myBox1 = new Box();
        System.out.println("Default Constructor");
        System.out.println("Volume: " + myBox1.getVolume());
        Box myBox2 = new Box(12.2, 3.5, 2.0);
        System.out.println("Parameterized Constructor");
        System.out.println("Volume: " + myBox2.getVolume());
    }
}
```

output:

```
D:\Balaji>javac Area.java
```

```
D:\Balaji>java Area
Default Constructor
Volume: 0.0
Parameterized Constructor
Volume: 85.39999999999999
```

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

Source Code

```
interface Father
{
    double HT=6.2;
    void height();
}
interface Mother
{
    double HT=5.8;
    void color();
}
class Child implements Father, Mother
{
    public void height()
    {
        double ht=(Father.HT+Mother.HT)/2;
        System.out.println("Child's Height= "+ht);
    }
    public void color()
    {
        System.out.println("Child Color= brown");
    }
    public static void main(String[] args)
    {
        Child c=new Child();
        c.height();
    }
}
```

```
c.color();  
}  
}
```

output:

```
F:\Java Lab CSE 2-2>javac Child.java
```

```
F:\Java Lab CSE 2-2>java Child  
Child's Height= 6.0  
Child Color= brown
```

EXPERIMENT - VII

A) Write a JAVA program that describes exception handling mechanism

Source Code:

```
class Division  
{  
public static void main(String []args)  
{  
try  
{  
System.out.println("WELCOME");  
int a=5;  
int b=0;  
int c=a/b;  
System.out.println("The Division is "+c);  
}  
catch(ArithmeticExceptionae)  
{  
System.out.println("Division with zero is not possible");  
}  
finally  
{  
System.out.println("LOGOUT");  
}  
}  
}
```

Output:

```
F:\Java Lab CSE 2-2>javac Division.java
```

```
F:\Java Lab CSE 2-2>java Division  
WELCOME  
Division with zero is not possible  
LOGOUT
```

B) Write a JAVA program Illustrating Multiple catch clauses

Source Code:

```
import java.util.*;
class Multiple
{
    public static void main(String[ ] args)
    {
        try
        {
            System.out. println("WELCOME");
            Scanner sc=new Scanner(System.in);
            System.out. print("Enter a value: ");
            int a=sc.nextInt();
            System.out. print("Enter  b  value: ");
            int b=sc . nextInt();
            int c=a/b;
            System.out.println("The Division is "+c);
        }
        catch(InputMismatchExceptionae)
        {
            System.out.println("Wrong Input");
        }
        catch(ArithmeticExceptionae)
        {
            System.out.println("Division with zero is not possible");
        }
        finally
        {
            System.out.println("LOGOUT");
        }
    }
}
```

Output:

F:\Java Lab CSE 2-2>javac Multiple.java

F:\Java Lab CSE 2-2>java Multiple

WELCOME

Enter a value: 3

Enter b value: k

Wrong Input

LOGOUT

F:\Java Lab CSE 2-2>java Multiple

WELCOME

Enter a value: 2

Enter b value: 0

Division with zero is not possible

LOGOUT

F:\Java Lab CSE 2-2>java Multiple

WELCOME

```
Enter a value: 8
Enter b value: 2
The Division is 4
LOGOUT
```

EXPERIMENT - VIII

A) Write a JAVA program that implements Runtime polymorphism.

Source Code:

```
class Bank
{
    float interest()
    {
        return 0;
    }
}
class SBI extends Bank
{
    float interest()
    {
        return 8.4f;
    }
}
class AXIS extends Bank
{
    float interest()
    {
        return 7.3f;
    }
}
class RuntimePoly
{
    public static void main(String args[])
    {
        Bank b1=new SBI();
        System.out.println("SBI Rate of Interest: "+b1.interest());
        Bank b2=new AXIS();
        System.out.println("Axis Rate of Interest: "+b2.interest());
    }
}
```

Output:

```
F:\Java Lab CSE 2-2>javac RuntimePoly.java
```

```
F:\Java Lab CSE 2-2>java RuntimePoly
SBI Rate of Interest: 8.4
Axis Rate of Interest: 7.3
```

```
F:\Java Lab CSE 2-2>
```

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

Polymorphism:-

Polymorphism is the process of performing the single task in different ways. Polymorphism is the Greek word. In which 'poly' means many and 'morphs' means forms. It means many forms.

Polymorphism is classified into two types they are-

1. Static or Compile time Polymorphism
2. Dynamic or Run time Polymorphism

Runtime Polymorphism:-

Runtime Polymorphism is also called as Dynamic polymorphism. It is the process of calling the Overriding methods in runtime rather than in compile time.

In this process the overriding methods are called with reference of super class.

Explanation:-

In this previous example we have three classes Bank, SBI and AXIS. Bank is a parent class and SBI and AXIS are child classes. The child classes are overriding the method interest() of parent class. In this previous example we have child class object assigned to the parent class reference so in order to determine which method would be called, the type of the object would be determined at run-time. It is the type of object that determines which version of the method would be called (not the type of reference)

EXPERIMENT – IX

A) Write a JAVA program for creation of Illustrating throw.

Source Code:

```
import java.util.*;
class ThrowException
{
    static void checkAge(int age)
    {
        if (age < 18)
        {
            throw new ArithmeticException("Access denied - You must be at least 18
            years old.");
        }
    }
}
```

```
    }
else
{
System.out.println("Access granted - You are 18 years or above old !");
}
}

public static void main(String[] args)
{
Scanner s=new Scanner(System.in);
int age;
System.out.println("Enter your age");
age=s.nextInt();
checkAge(age); // Set age to 15 (which is below 18...)
}
}
```

Output:

```
F:\Java Lab CSE 2-2>javac ThrowException.java

F:\Java Lab CSE 2-2>java ThrowException
Enter your age
16
Exception in thread "main" java.lang.ArithmeticException: Access denied -
You must be at least 18 years old.
atThrowException.checkAge(ThrowException.java:8)
atThrowException.main(ThrowException.java:22)

F:\Java Lab CSE 2-2>java ThrowException
Enter your age
20
Access granted - You are 18 years or aboveold !
```

B) Write a JAVA program for creation of Illustrating finally

Source Code:

```
import java.util.*;
class Finally
{

public static void main(String[] args)
{
try
{
int n1,n2;
Scanner s=new Scanner(System.in);
System.out.println("WELCOME TO TRY BLOCK");
}
```

JAVA LAB MANUAL

```
System.out.println("Enter Two numbers for Division Operation");
n1=s.nextInt();
n2=s.nextInt();

        // Throw an Arithmetic exception
System.out.println("Reminder: "+(n1 /n2));
    }

    // catch an Arithmetic exception
catch (ArithmeticException e) {

System.out.println(
        "catch : exception handled.");
    }

    // Always execute
finally {

System.out.println("finally : i execute always.");
    }
}
```

Output:

```
F:\Java Lab CSE 2-2>javac Finally.java

F:\Java Lab CSE 2-2>java Finally
WELCOME TO TRY BLOCK
Enter Two numbers for Division Operation
6
3
Reminder: 2
finally : i execute always.

F:\Java Lab CSE 2-2>java Finally
WELCOME TO TRY BLOCK
Enter Two numbers for Division Operation
14
0
catch : exception handled.
finally : i execute always.
```

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

Source Code:

```
classBuildIn
{
public static void main(String args[])
{
try {
int a[] = new int[5];
```

```
a[6] = 9; // accessing 7th element in an array of
          // size 5
    }
    catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("Array Index is Out Of Bounds");
        System.out.println("accessing 7th element in an array of size 5");
    }
}
}
```

Output:

```
F:\Java Lab CSE 2-2>javac BuildIn.java

F:\Java Lab CSE 2-2>java BuildIn
Array Index is Out Of Bounds
accessing 7th element in an array of size 5
```

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

Source Code:

```
import java.util.*;
class MyException extends Exception
{
    MyException(String str)
    {
        super(str);
    }
}
class UserException
{
    public static void main(String args[])
    {
        try
        {
            double bal;
            System.out.println("WELCOME TO BANK");
            Scanner s=new Scanner(System.in);
            System.out.println("ENTER WITHDRAW AMOUNT");
            bal=s.nextDouble();
            if(bal>2500)
            {
                MyException me=new MyException("YOU ARE SUCCEED DAILY LIMIT");
                throw me;
            }
            System.out.println("BALANCE IS WITHDRAWN SUCCESSFULLY");
        }
        catch (MyException me)
        {
            System.out.println(me);
        }
    }
}
```


JAVA LAB MANUAL

```
finally
{
System.out.println("THAN YOU WITH US");
}
}
}
```

Output:

F:\Java Lab CSE 2-2>javac UserException.java

F:\Java Lab CSE 2-2>java UserException
WELCOME TO BANK
ENTER WITHDRAW AMOUNT
3000
MyException: YOU ARE SUCCEED DAILY LIMIT
THAN YOU WITH US

F:\Java Lab CSE 2-2>java UserException
WELCOME TO BANK
ENTER WITHDRAW AMOUNT
2000
BALANCE IS WITHDRAWN SUCCESSFULLY
THAN YOU WITH US

EXPERIMENT - X

- 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 thethird display "Welcome" every 3 seconds , (Repeat the same by implementing Runnable)**

Source Code:

```
classGoodMorning extends Thread
{
public void run()
{
for(int i=0;i<10;i++)
{
try
{
Thread.sleep(1600);
}
catch(Exception e){}
System.out.println("GoodMorning");
}
}
}
class Hello extends Thread
{
```

```
public void run()
{
for(int i=0;i<10;i++)
{
try{
Thread.sleep(2000);
}
catch(Exception e){}
System.out.println("Hello");
}
}
}
class Welcome extends Thread
{
public void run()
{
for(int i=0;i<10;i++)
{
try{
Thread.sleep(3000);
}
catch(Exception e){}
System.out.println("Welcome");
}
}
}
class ThreadClass
{
public static void main(String[] args)
{
GoodMorninggm=new GoodMorning();
Thread t1=new Thread(gm);
Hello hl=new Hello();
Thread t2=new Thread(hl);
Welcome wc=new Welcome();
Thread t3=new Thread(wc);
t1.start();
t2.start();
t3.start();
}
}
```

Output:

```
F:\Java Lab CSE 2-2>javac ThreadClass.java
```

```
F:\Java Lab CSE 2-2>java ThreadClass
GoodMorning
Hello
Welcome
GoodMorning
Hello
GoodMorning
Welcome
```

Hello
GoodMorning
GoodMorning
Hello
Welcome
GoodMorning
Hello
GoodMorning
Welcome
Hello
GoodMorning
Hello
GoodMorning
Welcome
Hello
GoodMorning
Welcome
Hello
Hello
Welcome
Welcome
Welcome
Welcome

B) Write a program illustrating isAlive and join ()

Source Code

```
class MyThread extends Thread
{
    public void run()
    {
        System.out.println("r1 ");
        try {
            Thread.sleep(500);
        }
        catch (InterruptedException e) { }
        System.out.println("r2 ");
    }
    public static void main(String[] args)
    {
        MyThread t1=new MyThread();
        MyThread t2=new MyThread();
        t1.start();
        t2.start();
        System.out.println(t1.isAlive());
        System.out.println(t2.isAlive());
    }
}
```

output:

F:\Java Lab CSE 2-2>javac MyThread.java

```
F:\Java Lab CSE 2-2>java MyThread
true
r1
r1
true
r2
r2
```

C) Write a Program illustrating Daemon Threads.

Source Code:

```
class DaemonThread extends Thread
{
    String s;
    public DaemonThread(String name){
        s=name;
    }

    public void run()
    {
        // Checking whether the thread is Daemon or not
        if(Thread.currentThread().isDaemon())
        {
            System.out.println(s + " is Daemon Thread");
        }

        else
        {
            System.out.println(s + " is User Thread");
        }
    }

    public static void main(String[] args)
    {

        DaemonThread thread1 = new DaemonThread("thread1");
        DaemonThread thread2 = new DaemonThread("thread2");
        DaemonThread thread3 = new DaemonThread("thread3");

        thread1.setDaemon(true); // set user thread1 to Daemon

        thread1.start(); // starting thread1
        thread2.start(); // starting thread2

        thread3.setDaemon(true); // set user thread1 to Daemon
        thread3.start(); // starting thread3
    }
}
```

Output:

```
F:\Java Lab CSE 2-2>javac DaemonThread.java
```

```
F:\Java Lab CSE 2-2>java DaemonThread
thread3 is Daemon Thread
thread1 is Daemon Thread
thread2 is User Thread
```

EXPERIMENT - XI

A) Write a JAVA program Producer Consumer Problem

Source Code:

```
import java.util.*;
class Producer extends Thread
{
    StringBuffer sb=new StringBuffer();
    public void run()
    {
        synchronized(sb)
        {
            for(int i=1;i<=10;i++)
            {
                sb.append(i+" : ");
                System.out.println("Appending");
                try{
                    Thread.sleep(100);
                }
                catch (InterruptedException e)
                { }
            }
            sb.notify();
        }
    }
}
class Consumer extends Thread
{
    Producer prod;
    Consumer (Producer prod)
    {
        this.prod=prod;
    }
    public void run()
    {
        synchronized (prod.sb)
        {
            try{
                prod.sb.wait();
            }
            catch (Exception e)
            {}
            System.out.println("Data is: "+prod.sb);
        }
    }
}
```

```
}  
classCommunciate  
{  
public static void main(String[] args)  
{  
    Producer p=new Producer();  
    Consumer c=new Consumer(p);  
    Thread t1=new Thread(p);  
    Thread t2=new Thread(c);  
    t2.start(); //Consumer thread will start first  
    t1.start();  
}  
}
```

Output:

```
F:\Java Lab CSE 2-2>javac Communciate.java
```

```
F:\Java Lab CSE 2-2>java Communciate
```

```
Appending
```

```
Appending
```

```
Appending
```

```
Appending
```

```
Appending
```

```
Appending
```

```
Appending
```

```
Appending
```

```
Appending
```

```
Appending
```

```
Data is: 1 : 2 : 3 : 4 : 5 : 6 : 7 : 8 : 9 : 10 :
```

```
F:\Java Lab CSE 2-2>
```

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

Synchronization:

When thread is already acting on an object, preventing any other thread from acting on the same object is called Thread Synchronization or thread safe. Thread synchronization is recommended when multiple threads are used on the same object.

In the above Producer-Consumer program we are synchronization on StringBuffer Object. In the Producer Thread sb.notify() method is sending a notification to the

Consumer thread that the StringBuffer object sb is available, and it can be used now.

Meanwhile, what the Consumer thread is doing? It is waiting for the notification that theStringBuffer object sb (of Producer class) is available. Here, there is no need of using sleep() method to go into sleep for some time wait () method stops waiting as soon as it receives the notification. So there is no time delay to receive the data from the Producer.

EXPERIMENT - XII

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

Setting CLASSPATH:

The CLASSPATH is an environment variable that tells the Java compiler where to look for class files to import. CLASSPATH is generally set to a directory or a JAR (Java Archive) file.

» To see what is there in currently in CLASSPATH variable in your system. You can type the command in windows.

```
echo %CLASSPATH%
```

» Suppose, preceding command has displayed class path as:

```
c:\rnr;.
```

> This means the current class path is set to rnr directory in C: \ and also to the current directory represented by dot (.). Our package pack does not exist in either rar or current directory. Our package exists in D:\sub, as:

```
set CLASSPATH=D:\sub;.;%CLASSPATH%
```

B) Write a JAVA program that import and use the defined your package in the previous Problem Package source code:

Source Code:

```
package mathematical;
public class Arithmetic
{
    public void add(int a, int b)
```

```
{
int c;
c=a+b;
System.out.println("THE ADDITION OPERATION :"+c);
}
public void mul(int a, int b)
{
int c;
c=a*b;
System.out.println("THE MULTIPLICATION OPERATION :"+c);
}
public void sub(int a, int b)
{
int c;
c=a-b;
System.out.println("THE SUBTRACTION OPERATION :"+c);
}
public void division(int a, int b)
{
float c;
c=a/b;
System.out.println("THE DIVISION OPERATION :"+c);
}
public void reminder(int a, int b)
{
float c;
c=a%b;
System.out.println("THE REMINDER OPERATION :"+c);
}
}
```

compile and create the package using below command.

```
F:\Java Lab CSE 2-2>javac -d . Arithmetic.java
```

import the user defined package to our program

source code:

```
import mathematical.Arithmetic;
class Userdefine
{
public static void main(String args[])
{
Arithmetic A=new Arithmetic();
A.add(10,5);
A.sub(10,5);
A.mul(10,5);
A.division(10,5);
A.reminder(10,5);
}
}
```

Output:


```
F:\Java Lab CSE 2-2>javac Userdefine.java
```

```
F:\Java Lab CSE 2-2>java Userdefine
```

```
THE ADDITION OPERATION :15
THE SUBTRACTION OPERATION :5
THE MULTIPLICATION OPERATION :50
THE DIVISION OPERATION :2.0
THE REMINDER OPERATION :0.0
```

EXERCISE-13

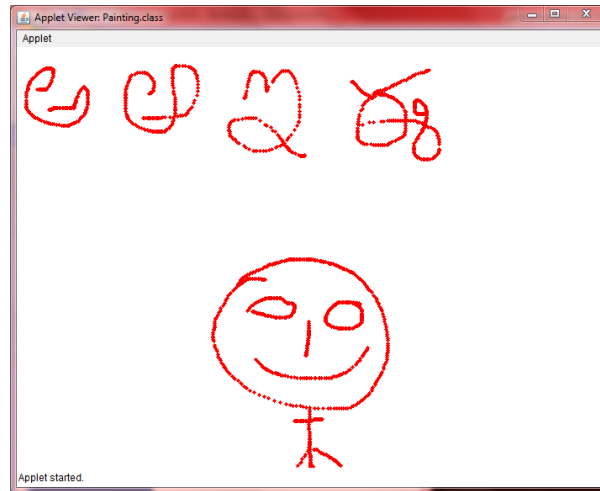
A) Write a JAVA program to paint like paintbrush in applet.

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

```
javac Painting.java
```

```
<html>
<applet code="Painting.class" width=700 height=500 >
</applet>
</html>
appletviewer Painting.html
```

Output:



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;
}

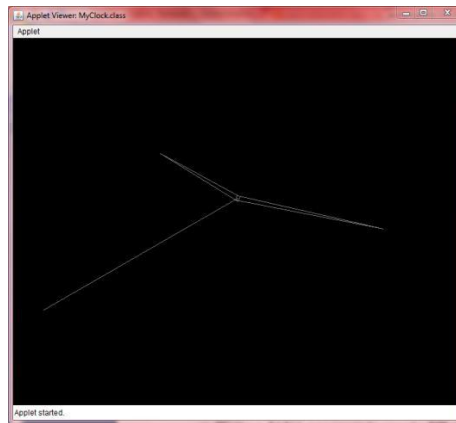
public void run()
{
    try
    {
        while (true)
        {
            Calendar cal =
            Calendar.getInstance();hours = cal.get(
            Calendar.HOUR_OF_DAY );if ( hours > 12 )
            hours-= 12;
            minutes = cal.get( Calendar.MINUTE
            );seconds=cal.get(Calendar.SECOND);

            if ( threadSuspended )
            {synchronized(this){
                while ( threadSuspended )
                {wait();
                }
            }
            }
            repaint();t.sleep(
            1000);
        }
    }
    catch (Exception e) { }
}
```

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

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

public void paint( Graphics g )
{
    g.setColor(Color.gray );
    drawWedge( 2*Math.PI * hours / 12, width/5, g
    );drawWedge( 2*Math.PI * minutes / 60, width/3, g
    );drawHand( 2*Math.PI * seconds / 60, width/2, g
    );g.setColor( Color.white );
}
}
```

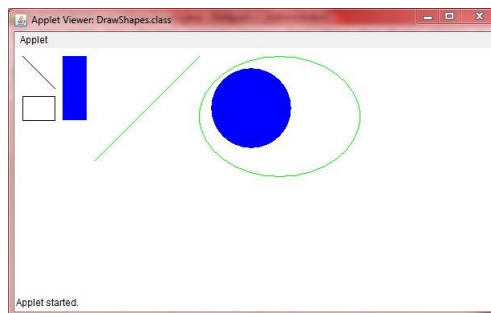


Output:

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

```
import java.awt.*; import java.  
applet.*;  
public class DrawShapes extends Applet  
{  
    public void paint(Graphics g)  
    {  
        g.drawLine(10,10,50,50);g.drawRect  
        (10,60,40,30);g.setColor(Color.blu  
        e);g.fillRect(60,10,30,80);g.setCo  
        lor(Color.green);g.drawLine(100,14  
        0,230,10);g.drawOval(230,10,200,15  
        0);  
        g.setColor(Color.blue);g.fillOval(  
        245,25,100,100);  
    }  
}
```

Output:

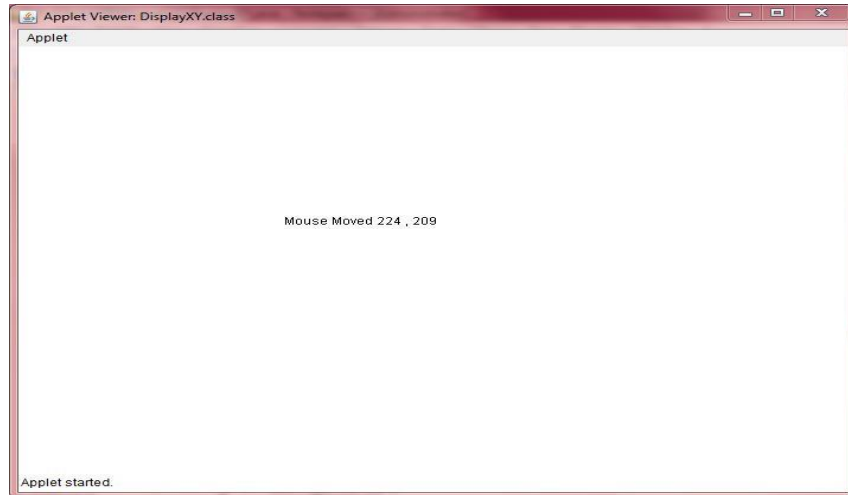


EXCERCISE-14

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

```
import java.awt.*; import
java.awt.event.*; import
java.applet.*;
public class DisplayXY extends Applet implements MouseMotionListener
{
    int x,y; String str
    ="";
    public void init()
    {
        addMouseMotionListener(this);
    }
    public void mouseDragged(MouseEvent me)
    {
        x = me.getX();
        y = me.getY();
        str = "Mouse Dragged "+x+" ,
        "+y; repaint();
    }
    public void mouseMoved(MouseEvent me)
    {
        x = me.getX();
        y = me.getY();
        str = "Mouse Moved "+x+" ,
        "+y; repaint();
    }
    public void paint(Graphics g)
    {
        g.drawString(str, x, y);
    }
}
```

Output:



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

```
import java.awt.*; import
java.awt.event.*; import
java.applet.*;
public class Key extends Applet implements KeyListener
{
    int X=20,Y=30;
    String
    msg=""; public void init()
    {
        addKeyListener(this); request
        Focus();
    }
    public void keyPressed(KeyEvent k)
    {
        showStatus("Key
        Pressed"); msg="Key
        Pressed"; repaint();
    }
    public void keyReleased(KeyEvent k)
    {
        showStatus("Key
        Up"); msg="Key Up"; repaint();
    }
    public void keyTyped(KeyEvent k)
    {
        showStatus("Key
        Typed"); msg="Key
        Typed"; repaint();
    }
}
```

```
    }  
    public void paint(Graphics g)  
    {  
        g.drawString(msg,X,Y);  
    }  
}
```

Output:

