1) Write a Java program find the area of circle.

```
import java.util.*;

class pra1

{

public static void main(String args[])

{

Scanner in=new Scanner(System.in);

double pi=3.14;

double circle;

int r;

System.out.print("enter the value of r:");

r=in.nextInt();

circle=pi*r*r;

System.out.println("Area of circle="+circle);

}

}
```

Output

```
enter the value of r:4

Area of circle=50.24
```

2) Write a Java program that will display factorial of the given number.

```
import java.util.*;

class pra2

{
```

```
public static void main(String args[])
{
Scanner in=new Scanner(System.in);

int i,f=1,n;
System.out.print("enter the value of n:");

n=in.nextInt();
for(i=1;i<=n;i++)
{
f=f*i;
}
System.out.print("Factorial of number is:="+f);
}
}
```

Output

```
enter the value of n:5
Factorial of number is:=120
```

3) Write a java program that will find largest no from the given two nos.

```
import java.util.*;
class pra3
{
public static void main(String args[])
{
Scanner in=new Scanner(System.in);
```

```
int a,b;
System.out.print("enter the value of A:");
a=in.nextInt();
System.out.print("enter the value of B:");
b=in.nextInt();
if(a>b)
{

System.out.println("a is greater"+a);
}
else
{

System.out.println("b is greater"+b);
}
}
}
```

Output

```
enter the value of A:5
enter the value of B:12
b is greater
```

4) Write a java program that will find largest no from the given three nos.

```
import java.util.*;
class pra4
{
```

```
public static void main(String args[])
{
Scanner in=new Scanner(System.in);
int a,b,c,max=0;
System.out.print("enter the value of A:");
a=in.nextInt();
System.out.print("enter the value of B:");
b=in.nextInt();
System.out.print("enter the value of C:");
c=in.nextInt();
max=a;
if(max<a)
{
max=a;
}
if(max<b)
{
max=b;
}
if(max<c)
{
max=c;
}
System.out.print("max no is="+max);
}
```

```
}
```

Output

```
enter the value of A:6

enter the value of B:12

enter the value of C:5

max no is=12
```

5) Write a Java program that shows the use of switch Statement.

```
import java.util.*;

class pra5

{

public static void main(String args[])

{

Scanner in=new Scanner(System.in);

int color;

System.out.print("enter the color code:");

color=in.nextInt();


switch(color)

{

case 1:System.out.print("red");

break;

case 2:System.out.print("green");

break;

case 3:System.out.print("blue");
```

```
break;
case 4:System.out.print("black");
break;
case 5:System.out.print("white");
break;
}


}
}
```

Output

```
enter the color code:2
green
```

6) Write a java program to find the sum of the digits of given number.

```
import java.util.*;
class pra6
{
public static void main(String args[])
{
Scanner in=new Scanner(System.in);
int a,n,sum=0;
System.out.print("enter the no:");
n=in.nextInt();
while(n>0)
{
a=n%10;
```

```
n=n/10;
sum=sum+a;
}
System.out.print("sum of digit="+sum);
}
}
```

Output

```
enter the no:521
sum of digit=8
```

7) Wtrie a java program that will display the sum of 1+1/2+1/3….1/n.

```
import java.util.*;
class pra7
{
public static void main(String args[])
{
Scanner in=new Scanner(System.in);
float sum=0;
int n;
System.out.print("enter the value of n:");
n=in.nextInt();
for(int i=1;i<=n;i++)
{
sum=sum+(float)1/i;
}
System.out.print(+sum);
```

```
}
}
```

Output

```
enter the value of n:2
1.5
```

8) Write a java program that check weather the given no is prime or not.

```
import java.util.*;
class pra8
{
public static void main(String args[])
{
int n,i,flag=0;
Scanner in = new Scanner(System.in);
System.out.println("Enter No");
n=in.nextInt();
for(i=2;i<=n/2;i++)
{
if(n%i==0)
{
flag=1;
}
}
if(flag==0)
{
System.out.println("no is prime");
```

```
}
else
{
System.out.println("no is not prime");
}
}
}
```

Output

```
Enter No 6
no is not prime
```

9) Write a java program that implements the use of break statement.

```
 class p9 {
public static void main(String[] args) {
   for(int i=0;i<10;i++)
   {
       if(i==5)
       {
              System.out.println("terminating the loop");
              break;
       }
       System.out.println("Still in the loop");
   }
 }
}
```

Output

```
Still in the loop
Still in the loop
Still in the loop
Still in the loop
Still in the loop
terminating the loop
```

10) write a java program that implements the use of continue statement.

```
class PR10
{
```

```
   public static void main(String[] args)
        {
     for(int i=1;i<=15;i++)
    {
        if(i==10)
                continue;
                System.out.println(" "+i);
    }
  }
}
```

Output

```
1
 2
 3
 4
 5
 6
 7
 8
 9
 11
 12
 13
 14
 15
```

11) Write a java program that will accept Command -line Argument and display the same.

```
    public class PR11
{
```

```
   public static void main(String args[])
         {
      for(int i=0;i<args.length;++i)
      {
        System.out.println("Argument["+i+"]:"+args[i]);

      }
   }
}
```

Output

Process completed.

12) Write a java program to sort the elements of an array in Ascending Order.

```
        class PR12
{


   public static void main(String[] args)
         {
      int arr[]={1,8,5,16,19,10,12};
      int temp;
      System.out.println("Original Array");
      for(int i=0;i<arr.length;i++)
         System.out.println(arr[i]+" ");
      for(int i=0;i<arr.length;i++)
      {
         for(int j=0;j<i;j++)
         {
                  if(arr[i]<=arr[j])
                  {
                           temp=arr[i];
                           arr[i]=arr[j];
                           arr[j]=temp;

                  }
         }
      }
```

```
                System.out.println("\nAscending Array");
                for(int i=0;i<arr.length;i++)
                System.out.print(arr[i]+" ");
    }
}
```

Output

```
Original Array
1
8
5
16
19
10
12

Ascending Array
1 5 8 10 12 16 19
```

13) Write a java program to create a student class and generate a result of student (Total,Per,Grade)

```
import java.util.*;
 class PR13
{

   public static void main(String[] args)
         {

     int total=510,sub=6,per;
     per=total/sub;
      if(per>=70)
      {
          System.out.println("Distriction Class");
      }
      else if(per>=60 && per<=69)
      {
          System.out.println("First Class");
      }
      else if(per>=50 && per<=59)
      {
          System.out.println("Second Class");
```

```
      }
      else if(per>=35 && per<=49)
      {
          System.out.println("Pass Class");
      }
      else
      {
          System.out.println("Fail");
      }
     System.out.println("Total= "+total);
     System.out.println("Subject= "+sub);
     System.out.println("Percentage= "+per);
   }
}
```

Output

```
Distriction Class
Total= 510
Subject= 6
Percentage= 85
```

14) Write a java program to create employee class and generate Salary sleep for the employee.

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

   public static void main(String[] args)
         {
                Scanner in = new Scanner(System.in);
                int basic;
                double  da,hra,pf,gross,net,ma=300;
                System.out.println("enter the basic");
                basic = in.nextInt();

                da=basic*0.10;
                hra=basic*0.075;
                pf=basic*0.12;
                gross=basic+da+hra+ma;
                net=gross-pf;

                System.out.println("Gross= "+gross);
                System.out.println("Net Salary= "+net);
   }
}
```

Output

enter the basic
5000
Gross= 6175.0
Net Salary= 5575.0

15) Write a java program show the use of static member.

```
class sample
{
    int x,y;
    static int count=1;
    sample()
    {
        x=0;
        y=0;
    }
     void display()
    {
        System.out.println("total object is created="+count);
        count++;
    }
}
class pra15
{
    public static void main(String args[])
    {
```

```
                sample s=new sample();
                sample s1=s;
                sample s2=s1;

                s.display();
                s1.display();
                s2.display();

        }
}
```

Output

total object is created=1

total object is created=2

total object is created=3

16) Write a java program which shows the nesting of methods.

```
import java.util.*;
class nest
{
        int x,y;
        nest(int a,int b)
        {
                x=a;
                y=b;
        }
        int mul()
```

```
	{
		return(x*y);
	}
	void display()
	{
		System.out.println("multiplication="+mul());
	}
}
class pra16
{
	public static void main(String args[])
	{
		nest n=new nest(10,20);
		n.display();
	}
}
```

output

```
multiplication=200
```

17) Write a java program which shows the use of Method Overloading.

```
class overload
{
	String name;
	int no;

	void setname(String n)
```

```
		{
			name=n;
		}
	void setname(String n,int i)
		{
			name=n;
			no=i;

		}
	void showvalues()
		{
	System.out.println("Your Name Is : "+name);
	System.out.println("Your No Is : "+no);
		}
}
public class PR17
{

   public static void main(String[] args)
   {
      overload obj=new overload();
			obj.setname("Mitesh",3);
			obj.showvalues();
			obj.setname("Bhavik",4);
			obj.showvalues();
```

```
    }
}
```

Output

```
Your Name Is : Mitesh
Your No Is : 3
Your Name Is : Bhavik
Your No Is : 4
```

18) Write a java program which implements Default Constructors.

```
    import java.util.*;
class constra
{
        int x,y;
        constra()                   //defalt constractor
        {
                x=0;y=0;
        }
        void display()
        {
                System.out.println("x="+x);
                System.out.println("y="+y);

        }
```

```
}
class pra18
{
	public static void main(String args[])
	{
		constra c=new constra();
		c.display();
	}
}
```

Output

```
x=0
y=0
```

19)Write a java program which implements parameterized Constructors.

```
import java.util.*;
class constra
{
	int x,y;
	constra(int a,int b)		//perameterized constractor
	{
		x=a;y=b;
	}
	void display()
```

```
	{
		System.out.println("x="+x);
		System.out.println("y="+y);

	}
}
class pra19
{
	public static void main(String args[])
	{
		constra c=new constra(100,800);
		c.display();
	}
}
```

Output

```
x=100
y=800
```

20) Write a java program which implements the Overloading of Constructors.

```
import java.util.*;
class constra
{
	int x,y;
```

```
	constra()			//default constractor
	{
		x=5;
		y=10;
	}
	constra(int a,int b)		//perameterized constractor
	{
		x=a;
		y=b;
	}
	void display()
	{
		System.out.println("x="+x);
		System.out.println("y="+y);


	}
}
class pra20
{
	public static void main(String args[])
	{
		constra c=new constra();
		constra c1=new constra(100,800);
		System.out.println("Default constractor");
```

```
                c.display();
                System.out.println("Perameterized constractor");
                c1.display();
        }
}
```

Output

```
x=5
y=10
Perameterized constractor
x=100
y=800
```

21) Write a java program which implements the concept of Single inheritance.

```
class student
{
	int no;
	student(int x)
	{
		no=x;
	}
	void display()
	{
		System.out.println("Student no:"+no);
	}
}
class faculty extends student
{
	int sub1,sub2;
	faculty(int x,int y,int z)
	{
		super(x);
		sub1=y;
		sub2=z;
	}
	void show()
	{
```

```
                display();
                System.out.println("sub1="+sub1);
                System.out.println("sub2="+sub2);
        }

}
class pra21
{
        public static void main(String args[])
        {
                faculty f=new faculty(1,70,80);
                f.show();
        }
}
```

Student no:1

sub1=70

sub2=80 Output

22) Write a java program which implements the concept of Multilevel inheritance.

```
class student
{
        int no;
        student(int x)
```

```
	{
		no=x;
	}
	void display()
	{
		System.out.println("Student no:"+no);
	}
}
class faculty extends student
{
	int sub1,sub2;
	faculty(int x,int y,int z)
	{
		super(x);
		sub1=y;
		sub2=z;
	}
	void show()
	{
		display();
		System.out.println("sub1="+sub1);
		System.out.println("sub2="+sub2);
	}
```

```
}
class principal extends faculty
{
	int total;
	principal(int x,int y,int z)
	{
		super(x,y,z);
	}
	void print()
	{
		total=sub1+sub2;
		System.out.println("total="+total);
	}
}
class pra22
{
	public static void main(String args[])
	{
		principal p=new principal(1,70,80);
		p.show();
		p.print();
	}
}
```

Output

Student no:1

sub1=70

sub2=80

total=150

23) Write a java program which implements the concept of Hierarchical inheritance.

```
class student
{
	int no;
	student(int x)
	{
		no=x;
	}
	void display()
	{
		System.out.println("Student no:"+no);
	}
}
class faculty extends student
{
	int sub1,sub2;
	faculty(int x,int y,int z)
	{
```

```
                super(x);
                sub1=y;
                sub2=z;
        }
        void show()
        {
                display();
                System.out.println("sub1="+sub1);
                System.out.println("sub2="+sub2);
        }

}
class principal extends student
{
        int a,b;
        principal(int x,int y,int z)
        {
                super(x);
                a=y;
                b=z;
        }
        void print()
        {
                System.out.println("a="+a);
```

```
                System.out.println("b="+b);
        }
}
class pra23
{
        public static void main(String args[])
        {
                principal p=new principal(1,70,80);
                faculty f=new faculty(1,100,200);
                f.show();
                p.display();
                p.print();
        }
}
```

Output

```
Student no:1
sub1=100
sub2=200
Student no:1
a=70
b=80
```

24) Write a java program which shows the Method Overriding.

```
class sample
{
        int  x;
        int y;
        sample(int a,int b)
        {
                int c;
                x=a;
                y=b;

        }
        void show()
        {
                System.out.println(+x);
                System.out.println(+y);
        }
}
class example extends sample
{
        int z;
        example(int a,int b,int c)
        {
                super(a,b);
                z=c;
```

```
        }
        void show()
        {
                System.out.println(+x);
                System.out.println(+y);
                System.out.println(+z);
        }
}
class pra24
{
        public static void main(String args[])
        {
                example e=new example(10,20,30);
                sample s=new sample(50,90);
                s.show();
                e.show();
        }
}
```

Output

```
50
90
10
20
30
```

25) Write a java program to implement final class and final method

```
class point
{
	int x,y;


}
class colored_point extends point
{
	int color;
}
final class colored_3dpoint extends colored_point
{
	int z;
final void display()
{
	System.out.println("Final Method");
}
}
final class demo
{
	public static void main (String[] args)
	{
		colored_3dpoint o=new colored_3dpoint();
		o.x=5;
		o.y=8;
```

```
                o.z=1;
                o.color=10;
                o.display();
                System.out.println("X = "+o.x);
                System.out.println("Y = "+o.y);
                System.out.println("Z = "+o.z);
                System.out.println("Color = "+o.color);
        }
}
```

Output

```
Final Method
X = 5
Y = 8
Z = 1
Color = 10
```

26) Write a java program to implement abstract class and abstract method.

```
import java.util.*;
abstract class demo
{
        int x;
        abstract void display();


}
 abstract class demo1 extends demo
{
        abstract void display();
        {
                System.out.println("hello");
        }



}
class pra26
{
        public static void main(String args[])
        {
          demo1 d = new demo1()
          {
        void display()
                {
```

```
            System.out.println("hi");
        }
    };
            d.display();



    }
}
```

Output

hello

hi

27) Write a Java program which implements interface

```
interface inheritance
{
    int roll=1;
    public void display(String s);
}
class my_inheritance implements inheritance
{
    public void display(String s)
    {
        System.out.println("Name is "+s);
    }
}
class MainClass
```

```
{
public static void main (String args[])
        {
                my_inheritance m1=new my_inheritance();
                System.out.println("Your No. Is "+m1.roll);
                m1.display("Pratap");
        }
}
```

Output

```
Your No. Is 1
Name is Pratap
```

28) Write a java program which implements Multiple Interfaces.

```
interface my_con
{
        int num1=10;
        int num2=20;
        int num3=30;
        String value="Marks ";
}
interface my_method
{
        public void display_value();
        public void display_marks();
        public void display_total();
```

```
}
class operation implements my_con,my_method
{
	public void display_value()
	{
		System.out.println("The Value Are "+value);
	}
	public void display_marks()
	{
		System.out.println("Marks1 : "+num1);
		System.out.println("Marks2 : "+num2);
		System.out.println("Marks3 : "+num3);
	}
	public void display_total()
	{
		int total=num1+num2+num3;
		System.out.println("Total Is : "+total);
	}
}
public class PR28
{
	 public static void main(String[] args)
	{
       operation o1=new operation();
```

```
        o1.display_value();
        o1.display_marks();
        o1.display_total();

    }
}
```

Output

The Value Are Marks

Marks1 : 10

Marks2 : 20

Marks3 : 30

Total Is : 60

29) Write a java program which shows importing of classes from other packages.

ClassA.java

```
package package1;
public class ClassA
{

        public void displayA()
        {
                System.out.println("Class A");
        }

}
```

Pr29.java

```
import package1.ClassA;
public class Pr29
{
   public static void main(String[] args)
   {
      ClassA a=new ClassA();
      a.displayA();
   }
}
```

Output

```
Class A
Process completed
```

30) Write a java program to implement the methods of Math Class.

```
import java.util.*;
class pra30
{
        public static void main(String args[])
```

```
	{
		double x;
		x=Math.max(10,20);
		System.out.println("The Maximum among Two is"+x);
		x=Math.sqrt(64);
		System.out.println("The sqrt of 64 is"+x);
		x=Math.abs(-55);
		System.out.println("The absolute value of -55 is"+x);
		x=Math.pow(16,3);
		System.out.println("The Qube of 16 is equal to"+x);
		x=Math.min(10,20);
		System.out.println("The Minimum among Two is"+x);
		x=Math.log(3);
		System.out.println("The Log of 3 is"+x);
	}
}
```

Output

```
The Maximum among Two is20.0
The sqrt of 64 is8.0
The absolute value of -55 is55.0
The Qube of 16 is equal to4096.0
The Minimum among Two is10.0
The Log of 3 is1.0986122886681098
```

31) Write a java program to implement the methods of string class.

```
class pra31
{
	public static void main(String args[])
	{
		StringBuffer str = new StringBuffer("object language");
		System.out.println("original string:" +str);
		System.out.println("Length of String:"+str.length());
			for(int i=0; i<str.length(); i++)
		{
			int p=i+1;
			System.out.println("Char at Position" +p+ "is"+str.charAt(i));
		}
		String astring = new String(str.toString());
		int pos = astring.indexOf("language");
		System.out.println("Modified String"+str);
		str.setCharAt(6,'-');
		System.out.println("String now"+str);
		str.append("improves security.");
		System.out.println("Append String"+str);
	}
}
```

Output

```
original string:object language
Length of String:15
```

Char at Position1iso

Char at Position2isb

Char at Position3isj

Char at Position4ise

Char at Position5isc

Char at Position6ist

Char at Position7is

Char at Position8isl

Char at Position9isa

Char at Position10isn

Char at Position11isg

Char at Position12isu

Char at Position13isa

Char at Position14isg

Char at Position15ise

Modified Stringobject language

String nowobject-language

Append Stringobject-languageimproves security

32) Write a java program to implement the methods of vector class.

```
import java.util.*;
class pra32
{
	public static void main(String args[])
	{
```

```
Vector v = new Vector(1, 1);
System.out.println("Initial size: " + v.size());
System.out.println("Initial capacity: " +
v.capacity());
v.addElement(new Integer(1));
v.addElement(new Integer(2));
v.addElement(new Integer(3));
v.addElement(new Integer(4));

System.out.println("Capacity after four additions: " +
v.capacity());
v.addElement(new Double(5.45));
System.out.println("Current capacity: " +
v.capacity());
v.addElement(new Double(6.08));
v.addElement(new Integer(7));
System.out.println("Current capacity: " +
v.capacity());
v.addElement(new Float(9.4));
v.addElement(new Integer(10));
System.out.println("Current capacity: " +
v.capacity());
v.addElement(new Integer(11));
v.addElement(new Integer(12));
System.out.println("First element: " +(Integer)v.firstElement());
```

```
            System.out.println("Last element: " +(Integer)v.lastElement());
            if(v.contains(new Integer(3)))
            System.out.println("Vector contains 3.");

            Enumeration vEnum = v.elements();
            System.out.println("\\nElements in vector:");
            while(vEnum.hasMoreElements())
            System.out.print(vEnum.nextElement() + " ");
            System.out.println();
        }
}
```

Output

Initial size: 0

Initial capacity: 1

Capacity after four additions: 4

Current capacity: 5

Current capacity: 7

Current capacity: 9

First element: 1

Last element: 12

Vector contains 3.

\nElements in vector:

1 2 3 4 5.45 6.08 7 9.4 10 11 12

33) Write a java program to implement the methods of Stack class

```
import java.util.*;
public class PR33
{

   public static void main(String args[])
   {
     Stack st=new Stack();
     st.push("Java");
     st.push("Latest");
     st.push("Edition");
     st.push("Six");
     st.search("");

     System.out.println("The Elements In The Stack : "+st);
     System.out.println("The Elements At The Top : "+st.peek());
     System.out.println("The Elements Poped Out OF The Stack : "+st.pop());
     System.out.println("The Elements In A Stack After Pop Out And Element : "+st);
     System.out.println("The Result Of Searching : "+st.search("Edition"));
   }
}
```

Output

```
The Elements In The Stack : [Java, Latest, Edition, Six]
The Elements At The Top : Six
The Elements Poped Out OF The Stack : Six
The Elements In A Stack After Pop Out And Element : [Java, Latest, Edition]
```

The Result Of Searching : 1

34) Write a java program which read a text and count all occurrences of a particular word.

```
class Program34{
	public static void main(String args[]){
		String text[] = { "this", "computer", "is", "new", "this", "is", "computer"};
		System.out.print("The String is : ");
					for(int i=0; i<text.length;i++)
							System.out.print(text[i] + " ");
					System.out.println("\n");
		int c = 0;
		for(int i = 0;i<text.length;i++)
		{
			c = 0;
			for(int j =0 ;j<text.length;j++)
			{
				if(text[i] == text[j])
				{
					c++;
				}
			}
			System.out.println(text[i] + " => repeats " + c + " time(s).");
		}
	}
}
```

Output

The String is : this computer is new this is computer

this => repeats 2 time(s).

computer => repeats 2 time(s).

is => repeats 2 time(s).

new => repeats 1 time(s).

this => repeats 2 time(s).

is => repeats 2 time(s).

computer => repeats 2 time(s).

35) write a java program which read a string and rewrite it in the alphabetical order

eg The word "STRING" should be written a "GINRST".

```
import java.io.*;
import java.util.*;
class pra35
{
	String alphaOrder(String str)
	{
	char[] charArray=str.toCharArray();
	Arrays.sort(charArray);
	String aString=new String(charArray);
	return aString;
	}
	public static void main(String[] args)throws IOException
	{
```

```
        System.out.println("Enter the String->");
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        String inputString=br.readLine();
        pra35 obj=new pra35();
        String alphaString=obj.alphaOrder(inputString);
        System.out.println("String in the Alphabetic Order :" +alphaString);
        }
}
```

Output

```
Enter the String->
STRING
String in the Alphabetic Order :GINRST
```

36. Write a java program which creates threads using the thread class

```
class ClassA extends Thread
{
    public void run()
    {
        for(int i=1;i<=5;i++)
        {
            System.out.println("Class A : " + i);
        }
    }

}
class ClassB extends Thread
```

```
{
	public void run()
	{
		for(int i=1;i<=5;i++)
		{
			System.out.println("Class B : " + i);
		}
	}

}
class ClassC extends Thread
{
	public void run()
	{
		for(int i=1;i<=5;i++)
		{
			System.out.println("Class C : " + i);
		}
	}

}

class Program36
{
	public static void main(String args[])
```

```
    {
        ClassA objA = new ClassA();
        ClassB objB = new ClassB();
        ClassC objC = new ClassC();

        objA.start();
        objB.start();
        objC.start();

    }
}
```

Output

```
Class A : 1
Class B : 1
Class A : 2
Class A : 3
Class A : 4
Class A : 5
Class B : 2
Class B : 3
Class C : 1
Class B : 4
Class C : 2

Class B : 5
```

Class C : 3

Class C : 4

Class C : 5

37) Write a java program which shows the use of yield() ,stop() and sleep() methods.

```
class ClassA extends Thread
{
    public void run()
    {
        for(int i=1;i<=5;i++)
        {
            if(i==4)
            {
                try
                {
                    System.out.println("Class A : " + i);
                    Thread.sleep(5000);
                }

                catch(InterruptedException e)

                {



                }
            }
```

```
                else

                    System.out.println("Class A : " + i);


        }
        }

}
class ClassB extends Thread
{
    public void run()
    {
        for(int i=1;i<=5;i++)
        {
            if(i==2)
                Thread.yield();
            else
                System.out.println("Class B : " + i);
        }
    }

}
class ClassC extends Thread
```

```
{
    public void run()
    {
        for(int i=1;i<=5;i++)
        {
            if(i==3)
                stop();
            else
                System.out.println("Class C : " + i);
        }
    }

}

class Program37
{
    public static void main(String args[])
    {
        ClassA objA = new ClassA();
        ClassB objB = new ClassB();
        ClassC objC = new ClassC();

        objA.start();
        objB.start();
        objC.start();
```

```
    }
}
```

Output

```
Class A : 1
Class A : 2
Class A : 3
Class A : 4
Class B : 1
Class C : 1
Class B : 3
Class C : 2
Class B : 4
Class B : 5
Class A : 5
```

38).Write a java program which shows the Priority in Threads.

```
class Thread_A extends Thread
{
    public void run()
    {
    for(int i=1;i<=5;i++)
    {
      System.out.println("Class A : "+i);
    }
    }
```

```
}
class Thread_B extends Thread
{            public void run()
             {
             for(int j=1;j<=5;j++)
             {
              System.out.println("Class B : "+j);
             }
             }
}
class Thread_C extends Thread
{            public void run()
             {
             for(int k=1;k<=5;k++)
             {
              System.out.println("Class C : "+k);
             }
             }
}
public class PR38 {
   public static void main(String[] args) {
             Thread_A a=new Thread_A();
      Thread_B b=new Thread_B();
      Thread_C c=new Thread_C();
                  a.setPriority(Thread.NORM_PRIORITY);
```

```
        b.setPriority(Thread.NORM_PRIORITY);
        c.setPriority(Thread.NORM_PRIORITY);
            a.start();
            b.start();
            c.start();

  }
}
```

Output

```
Class C : 1
Class C : 2
Class C : 3
Class B : 2
Class C : 4
Class B : 3
Class C : 5
Class B : 4
Class B : 5
```

39) Write a java program which use of Runnable Interface.

```
class demo implements Runnable //step1
{
  public void run()   //step2
  {
```

```
     for(int i=1;i<=10;i++)
     {
       System.out.println("\t Threadx"+i);
     }
     System.out.println("End Of Threadx");
   }
}
class pra39
{
   public static void main(String args[])
   {
     demo d = new demo();
     Thread th = new Thread(d);   //step3
     th.start();                  //step4
     System.out.println("End of main thread");
    }
}
```

Output

```
End of main thread
    Threadx1
    Threadx2
    Threadx3
    Threadx4
    Threadx5
    Threadx6
```

```
    Threadx7
    Threadx8
    Threadx9
    Threadx10
End Of Threadx
```

40) Write a java program which uses try and catch for Exception Handling.

```
class Program40{
    public static void main(String args[])
    {
        int a,b,c;
        int x,y;
        a = 10;
        b = 5;
        c = 5;

        try
        {

            x = a /(b-c);

            System.out.println("X : " + x);
        }
        catch (ArithmeticException e)
        {
            System.out.println("\nError !! Can not divided by Zero");
```

```
            }
            y=a/(b+c);
            System.out.println("y ="+y);
        }
}
```

Output

```
Error !! Can not divided by Zero
y =1
```

41) write a java program which uses Multiple catch blocks.

```
class Pr41
{
	public static void main(String args[])
	{
		int a[] = {5,10};
		int b = 5;



		try
		{
		 //int  x = a[2]/b-a[1];
		    //int x = a[1]/0;


		}
		catch(ArithmeticException e)
		{
			System.out.println("Not Division By Zero");
		}
		catch(ArrayIndexOutOfBoundsException e)
		{
			System.out.println("Array Index Error");
```

```
            }


            int y  = a[1]/a[0];
            System.out.println("Y==>" + y);
      }
}
```

Output

Array Index Error

Y==>2

42) Write a java program which uses finally Statement.

```
class TestFinallyBlock
{
public static void main(String args[])
{
            try
{
            float data=25/5;
            System.out.println(data);
}
```

```
        catch(NullPointerException e)
{
        System.out.println(e);
}
        finally
{
        System.out.println("finally block is always executed");
}
}
}
```

Output

5.0

finally block is always executed

43) Write a java program which uses Nested try Statements.

```
class p43 {


   public static void main(String[] args) {
        try
        {
                int a=2,b=4,c=2,x=7,z;
                int p[]={2};
                p[3]=33;
                try
                {
```

```
                z=x/(b*b)-(4*a*c);
            //      z=a+b;
            //z=x/0;
                System.out.println("The value is ="+z);


            }
            catch(ArithmeticException e)
            {
                System.out.println("Division by Zero in Arithmetic expresion");


            }
        }

    catch(ArrayIndexOutOfBoundsException e)
    {
        System.out.println("Array index is out-of-bounds");


    }
  }
}
```

Output

Array index is out-of-bounds

44) Write a java program which shows throwing our own Exception.

```
class MyException extends Exception
{
```

```
    MyException(String str)
    {
        System.out.println(str);
    }
}
class Program44
{
    public static void main(String str[])
    {
        int x = 50,y = 10,z;

        try
        {
            z = x/y;
            if(z<10)
            {
                throw new MyException("ArithmeticException Found");
            }
        }
        catch(MyException e)
        {
            System.out.println("Exception Occure");
        }
        finally
        {
```

```
                System.out.println("I am always here.....");
            }
        }
}
```

Output

ArithmeticException Found

Exception Occure

I am always here.....

45) Create an Applet program that print Hello Applet.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

Pra45.html

```
<html>
        <head>
        </head>
        <body bgcolor="000000">
                <center>
                        <applet
                                code    = "pra45.class"
                                width   = "500"
                                height  = "300"
                                >
                        </applet>
                </center>
        </body>
</html>
```

Pra45.java

```
import java.applet.*;

import java.awt.*;


public class pra45 extends Applet

{

   public void paint(Graphics g)

   {

     g.drawString("Hello Applet",10,100);

   }

}
```

Output

Hello Applet

46).Create an applet that use init(),start(),stop() and destroy() methods of applet.

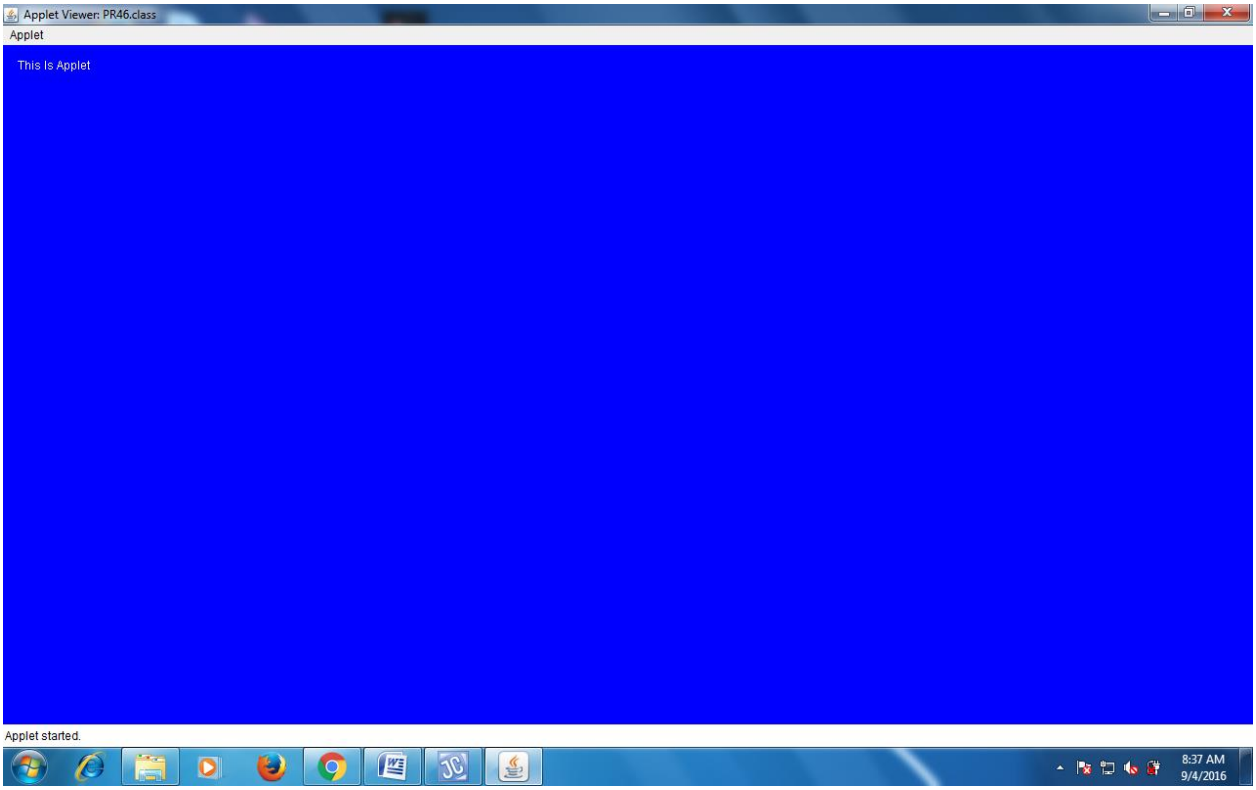
Pr46.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>
        <head>
        </head>
        <body bgcolor="000000">
                <center>
                        <applet
                                code    = "PR46.class"
                                width   = "500"
```

```
                                        height    = "500"
                                        >
                                </applet>
                        </center>
                </body>
</html>
```

Pr46.java

```
import java.awt.*;

import java.awt.Graphics;


public class PR46 extends java.applet.Applet
{

    String text="This Is Applet";

    public void init()
    {
        text="This Is Applet";
        setBackground(Color.blue);
    }
    public void start()
    {
            System.out.println("Starting....");
    }
    public void stop()
    {
            System.out.println("Stopping...");
    }
    public void distroy()
    {
            System.out.println("Priority To Unload");
    }


    public void paint(Graphics g)
    {
      System.out.println("Paint");
      g.setColor(Color.white);
      g.drawString(text,15,25);
    }
}
```



47) Write an applet program to implement the concept of passing parameter to applet.

```
import java.awt.*;

import java.applet.*;

public class p47 extends Applet {

              String str;

            public void init() {

                        str=getParameter("String");

      if(str==null)

            str="Java";

            str="Hello"+str;

            }

            public void paint(Graphics g) {

             g.drawString(str,10,100);
```

```
        }
}
```

Output

HelloJava

48).Write a applet program to implement various methods of Graphics class.

PR48.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
        <head>
        </head>
        <body bgcolor="000000">
                <center>
                        <applet
                                code    = "PR48.class"
                                width   = "1000"
                                height  = "800">
                        </applet>
                </center>
        </body>
</html>
```

PR48.java

```
import java.awt.*;
import java.applet.*;
public class PR48 extends java.applet.Applet
{
  public void paint(Graphics g)
    {
```

```
        g.setColor(Color.red);
                        g.drawString("Welcome",50, 50);
                        g.drawLine(20,30,20,300);
                        g.drawRect(70,100,30,30);
                        g.fillRect(170,100,30,30);
                        g.drawOval(70,200,30,30);
                        g.setColor(Color.orange);
                        g.fillOval(170,200,30,30);
                        g.drawArc(90,150,30,30,30,270);
                        g.fillArc(270,150,30,30,0,180);

    }
}
```