History of Java

Java is a high-level programming language developed by Sun Microsystems in the year 1995. James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. Java was originally developed by ***James Gosling*** at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995. The small team of sun engineers called Green Team. Originally designed for small, embedded systems in electronic appliances like set-top boxes. It was called Oak and was developed as a part of the Green project. Oak is a symbol of strength and choosen as a national tree of many countries like U.S.A., France, Germany, Romania etc. In 1995, Oak was renamed as "Java" because it was already a trademark by Oak Technologies. Java is an island of Indonesia where first coffee was produced (called java coffee). Java is just a name not an acronym. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. Currently, Java is used in internet programming, mobile devices, games, e-business solutions etc.

Features of Java/Java Buzzwords

Simple

Java was designed to be easy for the professional programmer to learn and use effectively. Java inherits the C/C++ syntax and many of the object-oriented features of C++.

Object-Oriented

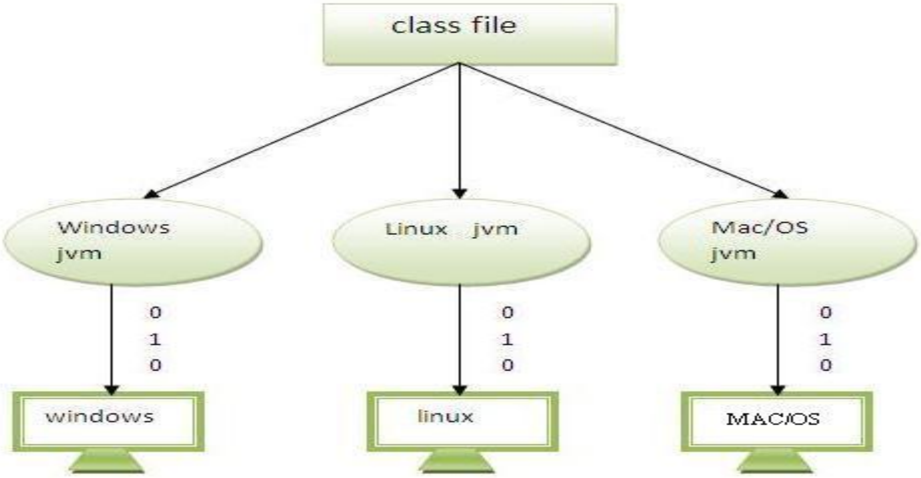
Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behaviour. Object-oriented programming(OOPs) is a methodology that simplify software development and maintenance by providing some rules.

Basic concepts of OOPs are:

- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

Platform Independent

java is platform independent. Java code can be run on multiple platforms e.g.Windows,Linux,Sun Solaris,Mac/OS etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere(WORA).



Secured

Java is secured because:

- No explicit pointer
- Programs run inside virtual machine.

Robust

Robust simply means strong. Java uses strong memory management. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. Because Java is a strictly typed language, it checks your code at compile time. All these points makes java robust.

Multithreaded

Java was designed to meet the real-world requirement of creating interactive networking programs. To accomplish this, Java supports multithreaded programming, which allows you to write programs that do many things simultaneously.

Interpreted and High Performance

Java enables the creation of cross-platform programs by compiling into an intermediate representation called Java bytecode. This code can be executed on any system that implements the Java Virtual Machine. Java bytecode was carefully designed so that it would be easy to translate directly into native machine code for very high performance by using a just-in-time compiler.

Distributed

Java is designed for the distributed environment of the Internet because it handles TCP/IP protocols. In fact, accessing a resource using a URL is not much different from accessing a file. Java also supports Remote

Method Invocation (RMI). This feature enables a program to invoke methods across a network.

Dynamic

Java programs carry with them substantial amounts of run-time type information that is used to verify and resolve accesses to objects at run time. This makes it possible to dynamically link code in a safe and expedient manner.

Question1

No:

Aim:

Write a program to find the distance between two points.

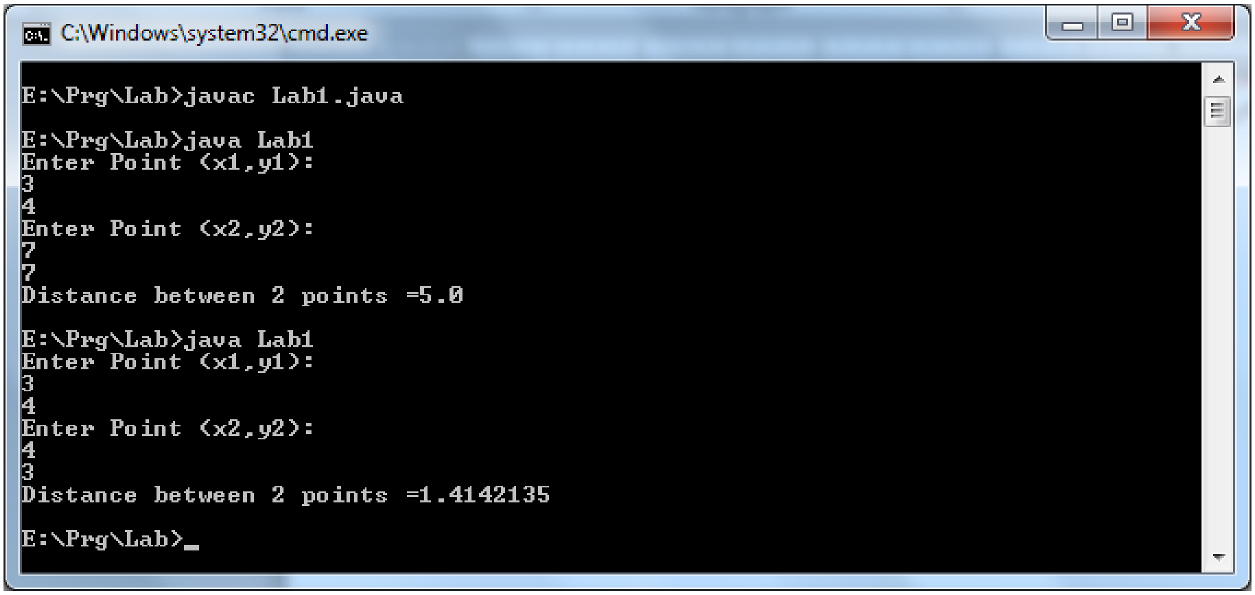
```
import java.util.Scanner;
class Lab1
{
public static void main(String args[])
{
 float   x1, x2, y1, y2,res;
 Scanner obj1=new Scanner(System.in);
 System.out.println("Enter Point (x1,y1): " );
 x1=obj1.nextInt();
 y1=obj1.nextInt();
System.out.println( "Enter Point (x2,y2): " );
 x2=obj1.nextInt();
 y2=obj1.nextInt();
 res = (float)(Math.sqrt(Math.pow((y2-y1),2) + Math.pow((x2-x1),2)));
 System.out.println("Distance between 2 points ="+res);
}
}
```

Paste Printed output on left side after class diagram

Class Diagram

Lab1
main(args[]:String)

Output



```
E:\Prg\Lab>javac Lab1.java

E:\Prg\Lab>java Lab1
Enter Point (x1,y1):
3
4
Enter Point (x2,y2):
7
7
Distance between 2 points =5.0

E:\Prg\Lab>java Lab1
Enter Point (x1,y1):
3
4
Enter Point (x2,y2):
4
3
Distance between 2 points =1.4142135

E:\Prg\Lab>_
```

Question2

No:

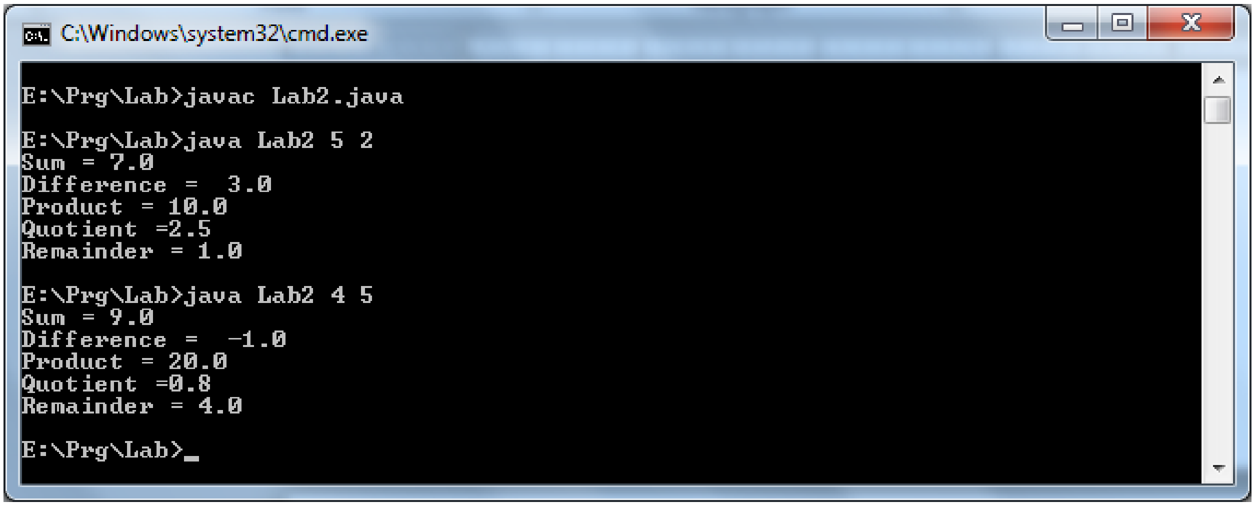
Aim: Write a program to find the sum, difference, product, quotient and remainder of two numbers passed as command line argument.

```
import java.util.Scanner;
class Lab2
{
public static void main(String args[])
{
 float   n1, n2, res;
 n1=Integer.parseInt(args[0]);
 n2=Integer.parseInt(args[1]);
 System.out.println("Sum = "+(n1+n2));
 System.out.println("Difference =  "+(n1-n2));
 System.out.println("Product = "+(n1*n2));
 System.out.println("Quotient ="+(n1/n2));
 System.out.println("Remainder = "+(n1%n2));
}
}
```

Paste Printed output on left side after class diagram

Class Diagram

Lab2
main(args[]:String)



```
E:\Prg\Lab>javac Lab2.java

E:\Prg\Lab>java Lab2 5 2
Sum = 7.0
Difference =  3.0
Product = 10.0
Quotient =2.5
Remainder = 1.0

E:\Prg\Lab>java Lab2 4 5
Sum = 9.0
Difference =  -1.0
Product = 20.0
Quotient =0.8
Remainder = 4.0

E:\Prg\Lab>_
```

Question3

No:

Aim: Write java program to display Fibonacci series up to a limit.

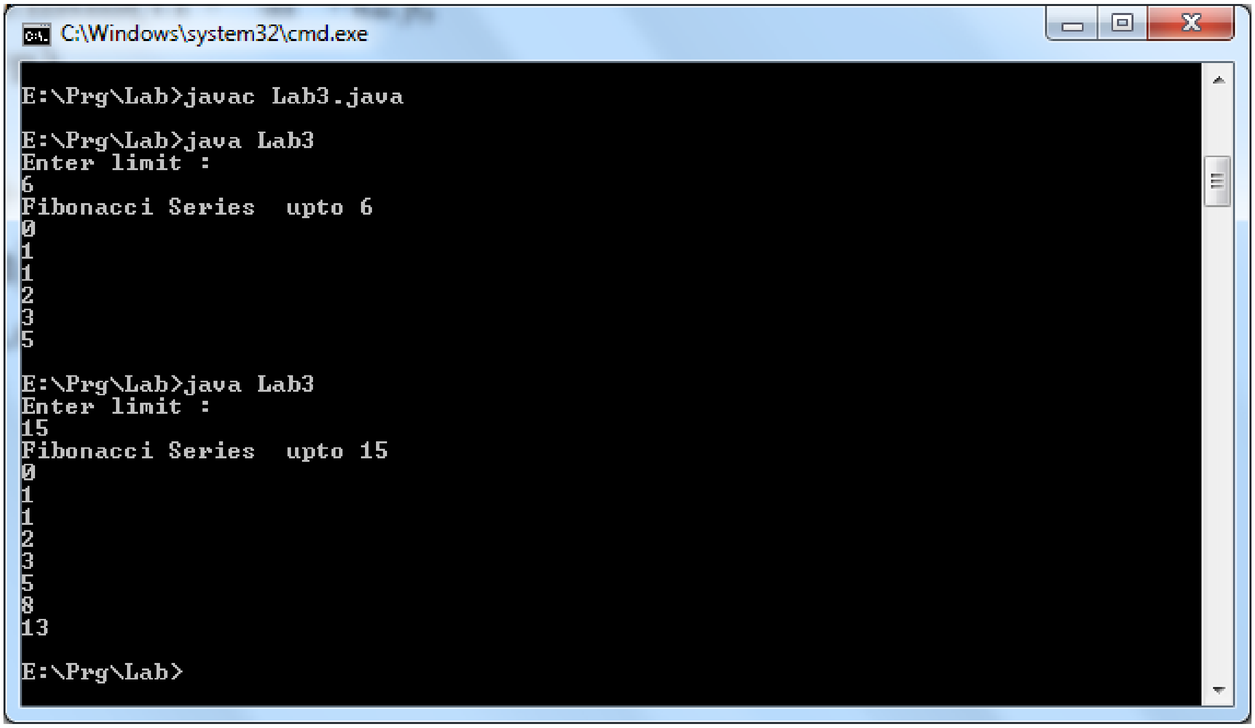
```
import java.util.Scanner;
class Fibonacci
{
int n, t1=0, t2=1,sum=0;
Scanner obj1=new Scanner(System.in);
void read()
{
System.out.println("Enter limit :");
n=obj1.nextInt();
}
void disp()
{
System.out.print("Fibonacci Series  upto " + n +"\n");
System.out.println(t1+"\n"+t2);
while(sum<=n)
        {
    sum = t1 + t2;
    if(sum<=n)
        System.out.println(sum);
    t1 = t2;
    t2 = sum;
        }
}
}
class Lab3
{
```

```
public static void main(String args[])
{
 Fibonacci f1=new Fibonacci();
f1.read();
f1.disp();
}
}
```

Fibonacci
t1:int t2:int sum:int

read()
disp()

Lab3
main(args[]:String)



```
E:\Prg\Lab>javac Lab3.java

E:\Prg\Lab>java Lab3
Enter limit :
6
Fibonacci Series  upto 6
0
1
1
2
3
5


E:\Prg\Lab>java Lab3
Enter limit :
15
Fibonacci Series  upto 15
0
1
1
2
3
5
8
13

E:\Prg\Lab>
```

Question 4

No:

Aim: Write java program to display Armstrong numbers within a range.

```
import java.util.Scanner;
class Armstrong
```

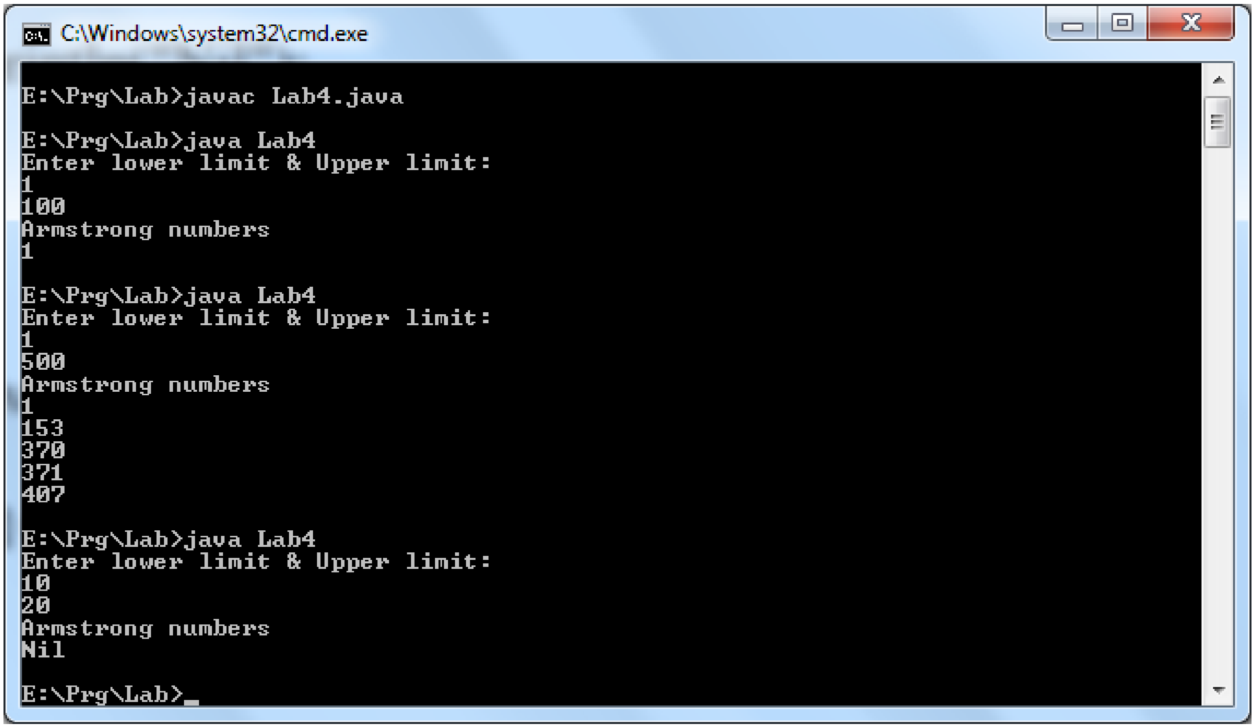
```
{
int n1,n2,n,d,i,temp,sum,count=0;
Scanner obj1=new Scanner(System.in);
void read()
{
System.out.println("Enter lower limit & Upper limit:");
n1=obj1.nextInt();
n2=obj1.nextInt();
}
void disp()
{
System.out.println("Armstrong numbers");
for(i=n1;i<=n2;i++)
{
n=i;
temp=n;
sum=0;
while(n>0)
{
d=n%10;
sum=sum+(int)Math.pow(d,3);
n=n/10;
}
if(temp==sum )
 {
 count=1;
 System.out.println(temp);
  }
}
if(count==0)
System.out.println("Nil");
}
}
class Lab4
{
```

```
public static void main(String args[])
{
Armstrong a1=new Armstrong();
a1.read();
a1.disp();
}
}
```

Armstrong
n1:int n2:int sum:int count:int

read()
disp()

Lab4
main(args[]:String)



```
E:\Prg\Lab>javac Lab4.java

E:\Prg\Lab>java Lab4
Enter lower limit & Upper limit:
1
100
Armstrong numbers
1

E:\Prg\Lab>java Lab4
Enter lower limit & Upper limit:
1
500
Armstrong numbers
1
153
370
371
407

E:\Prg\Lab>java Lab4
Enter lower limit & Upper limit:
10
20
Armstrong numbers
Nil

E:\Prg\Lab>_
```

Question 5

No:

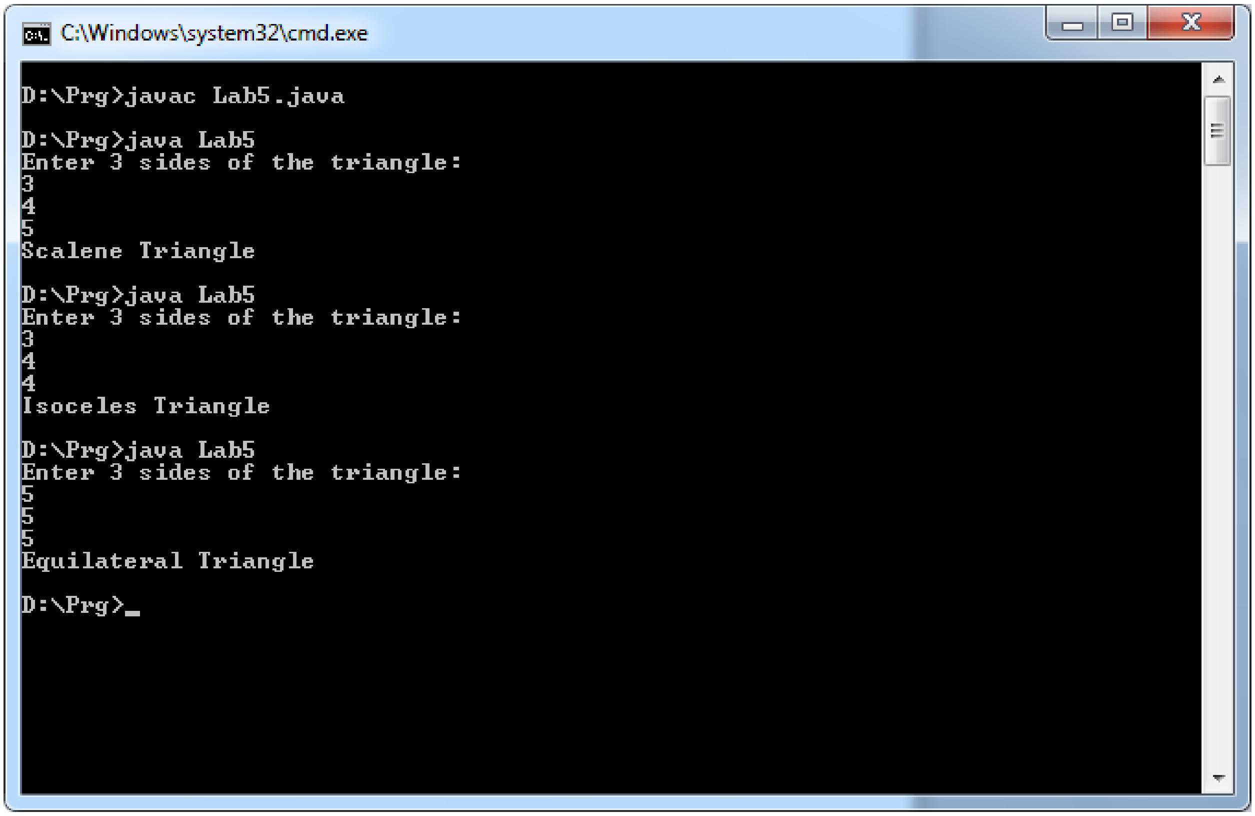
Aim: Given the sides of a triangle, write a program to check whether the triangle is equilateral, isosceles or scalene and find its area.

```
import java.util.Scanner;
class Triangle
{
int a,b,c;
Scanner obj1=new Scanner(System.in);
void read()
{
System.out.println("Enter 3 sides of the triangle:");
a=obj1.nextInt();
b=obj1.nextInt();
c=obj1.nextInt();
}
void check()
{
if (a == b && b == c )
        System.out.println("Equilateral Triangle");

  else if (a == b || b == c || c == a )
        System.out.println("Isoceles Triangle");

     else
        System.out.println("Scalene Triangle");
}
}
class Lab5
{
public static void main(String args[])
{
Triangle  t1=new Triangle();
t1.read();
t1.check();
}
}
```

Triangle
a:int b:int c:int

read()
check()

Lab5
main(args[]:String)



```
D:\Prg>javac Lab5.java

D:\Prg>java Lab5
Enter 3 sides of the triangle:
3
4
5
Scalene Triangle

D:\Prg>java Lab5
Enter 3 sides of the triangle:
3
4
4
Isoceles Triangle

D:\Prg>java Lab5
Enter 3 sides of the triangle:
5
5
5
Equilateral Triangle

D:\Prg>_
```

No:

Aim

Read an array of 10 or more numbers and write a program to find the

a) Smallest element in the array

b) Largest element in the array

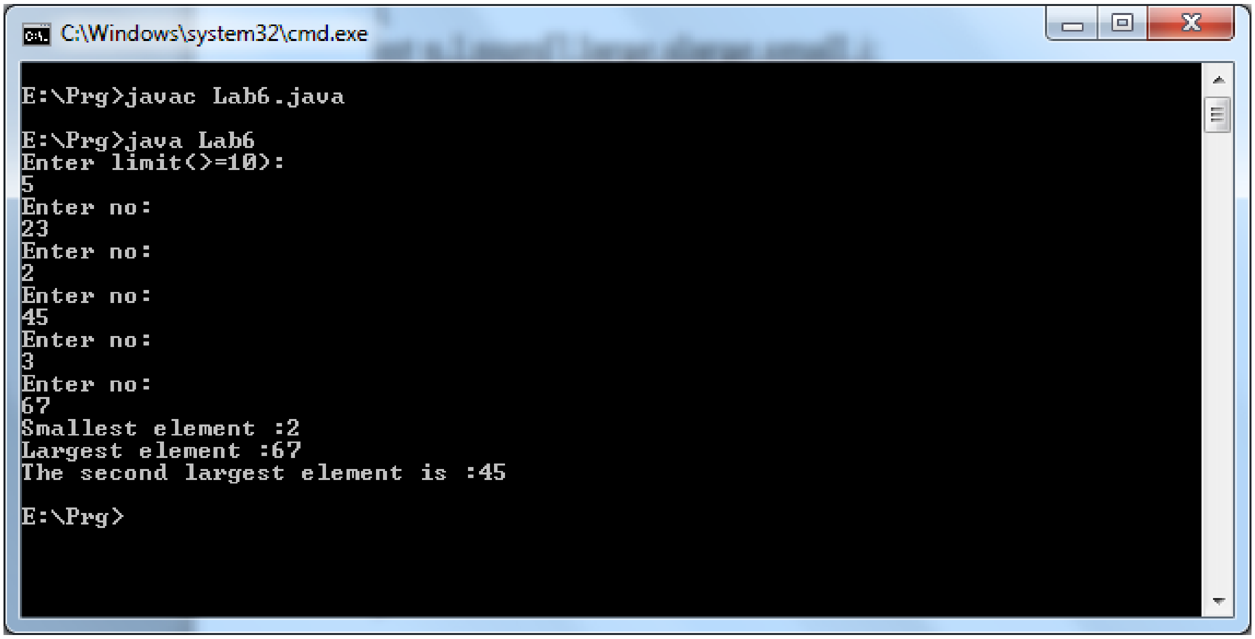
c) Second largest element in the array

```
import java.util.Scanner;
class Number
{
int n,num[],large,slarge,small,i;
Scanner obj1=new Scanner(System.in);
void read()
{
System.out.println("Enter limit(>=10):");
n=obj1.nextInt();
num=new int[n];
for(i=0;i<n;i++)
{
System.out.println("Enter no:");
num[i]=obj1.nextInt();
}
}
void check()
{
large=slarge=small=num[0];
for (i = 0; i < n; i++)
 {
if(num[i]<small)
  small=num[i];
     if (num[i] > large)
 {
            slarge=large;
            large=num[i];
 }
else if (num[i] > slarge && num[i] != large)
            slarge=num[i];
 }
System.out.println("Smallest element :"+small);
System.out.println("Largest element :"+large);
System.out.println("The second largest element is :"+ slarge);
}
}
class  Lab6
{
public static void main(String args[])
{
Number num1=new Number();
num1.read();
num1.check();
}
}
```

small:int large:int slarge:int num:int []

read()
check()

Lab6
main(args[]:String)



```
E:\Prg>javac Lab6.java

E:\Prg>java Lab6
Enter limit(>=10):
5
Enter no:
23
Enter no:
2
Enter no:
45
Enter no:
3
Enter no:
67
Smallest element :2
Largest element :67
The second largest element is :45

E:\Prg>
```

No:7

Aim : Write a program to perform base conversion

a) Integer to binary

b) Integer to Octal

c) Integer to Hexadecimal

```
import java.util.Scanner;
class Number
{
int n,num,count,arr[] = new int[20];
Scanner obj1=new Scanner(System.in);
void read()
   {
System.out.println("Enter no:");
```

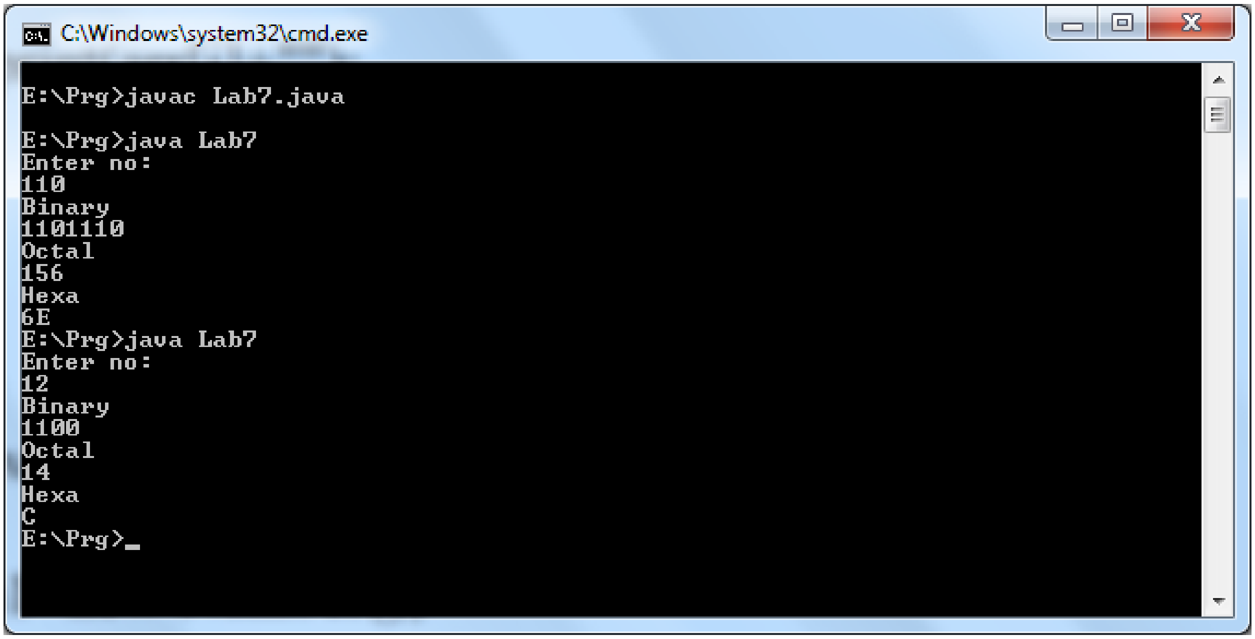
```
n=obj1.nextInt();
}
void intBinary()
{
count=0;
num=n;
while (num > 0)
 {
 arr[count] = num % 2;
  num = num / 2;
  count++;
}
System.out.println("Binary");
for (int i = count-1; i >= 0; i--)
 System.out.print(arr[i]+"");
}
void intOctal()
{
count=0;
num=n;
while (num > 0)
 {
 arr[count] = num % 8;
  num = num / 8;
  count++;
}
System.out.println("\nOctal");
for (int i = count-1; i >= 0; i--)
 System.out.print(arr[i]+"");
}
void intHexa()
{
count=0;
num=n;
while (num > 0)
 {
 arr[count] = num % 16;
  num = num / 16;
  count++;
}
```

```
System.out.println("\nHexa");
for (int i = count-1; i >= 0; i--)
{
if(arr[i]==10)
 System.out.print('A');
else if(arr[i]==11)
 System.out.print('B');
else if(arr[i]==12)
 System.out.print('C');
else if(arr[i]==13)
 System.out.print('D');
else if(arr[i]==14)
 System.out.print('E');
 else if(arr[i]==15)
 System.out.print('F');
else
 System.out.print(arr[i]+"");
}
} }
class  Lab7
{
public static void main(String args[])
{
Number num1=new Number();
num1.read();
num1.intBinary();
num1.intOctal();
num1.intHexa();
}
 }
```

Number
n:int num:int count:int arr:int []

read()
intBinary()
intOctal()
intHexa()

Lab7
main(args[]:String)



```
E:\Prg>javac Lab7.java

E:\Prg>java Lab7
Enter no:
110
Binary
1101110
Octal
156
Hexa
6E
E:\Prg>java Lab7
Enter no:
12
Binary
1100
Octal
14
Hexa
C
E:\Prg>_
```

No:

Aim: Write a program to merge two arrays.

```
import java.util.Scanner;
class NumArray
{
int a[],b[],c[],n1,n2,i,j,k;
Scanner obj1=new Scanner(System.in);
void read()
{
System.out.println("Enter size of Array1:");
n1=obj1.nextInt();
a=new int[n1];
System.out.println("Enter elements in sorted order:");
for(i=0;i<n1;i++)
{
a[i]=obj1.nextInt();
}
System.out.println("Enter size of Array2:");
```

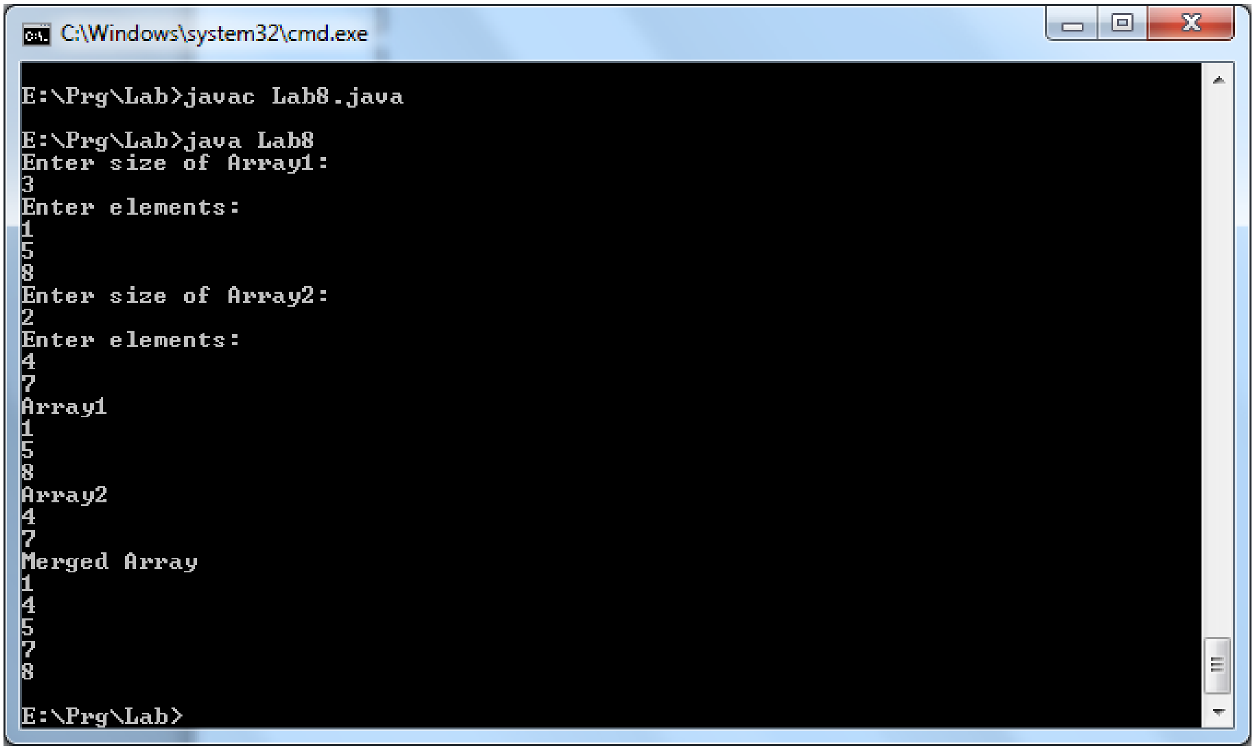
```
n2=obj1.nextInt();
b=new int[n2];
c=new int[n1+n2];
System.out.println("Enter elements in sorted order:");
for(i=0;i<n2;i++)
{
b[i]=obj1.nextInt();
}
}
void merge()
{
 i = 0;
 j = 0;
k=0;
while(i < n1 && j< n2)
    {
 if(a[i] < b[j])
 {
 c[k] = a[i];
 i++;
 }
else
{
c[k] = b[j];
j++;
}
k++;
}
while(i <n1)
  {
  c[k] = a[i];
  k++;
 i++;
}
while(j < n2)
  {
  c[k] = b[j];
 k++;
j++;
}
```

```
}
void disp()
{
System.out.println("Array1");
for(i=0;i<n1;i++)
{
System.out.println(a[i]+" ");
}
System.out.println("Array2");
for(i=0;i<n2;i++)
{
System.out.println(b[i]+" ");
}
System.out.println("Merged Array");
for(i=0;i<n1+n2;i++)
{
System.out.println(c[i]+" ");
}
}
}
class Lab8
{
public static void main(String args[])
{
NumArray num1=new NumArray();
num1.read();
num1.merge();
num1.disp();
}
}
```

NumArray
n1:int n2:int a:int[] b:int[] c:int[]

read()
merge()
disp()

Lab8
main(args[]:String)



```
E:\Prg\Lab>javac Lab8.java

E:\Prg\Lab>java Lab8
Enter size of Array1:
3
Enter elements:
1
5
8
Enter size of Array2:
2
Enter elements:
4
7
Array1
1
5
8
Array2
4
7
Merged Array
1
4
5
7
8

E:\Prg\Lab>
```

No:

Aim: Write a program to find the trace and transpose of a matrix.

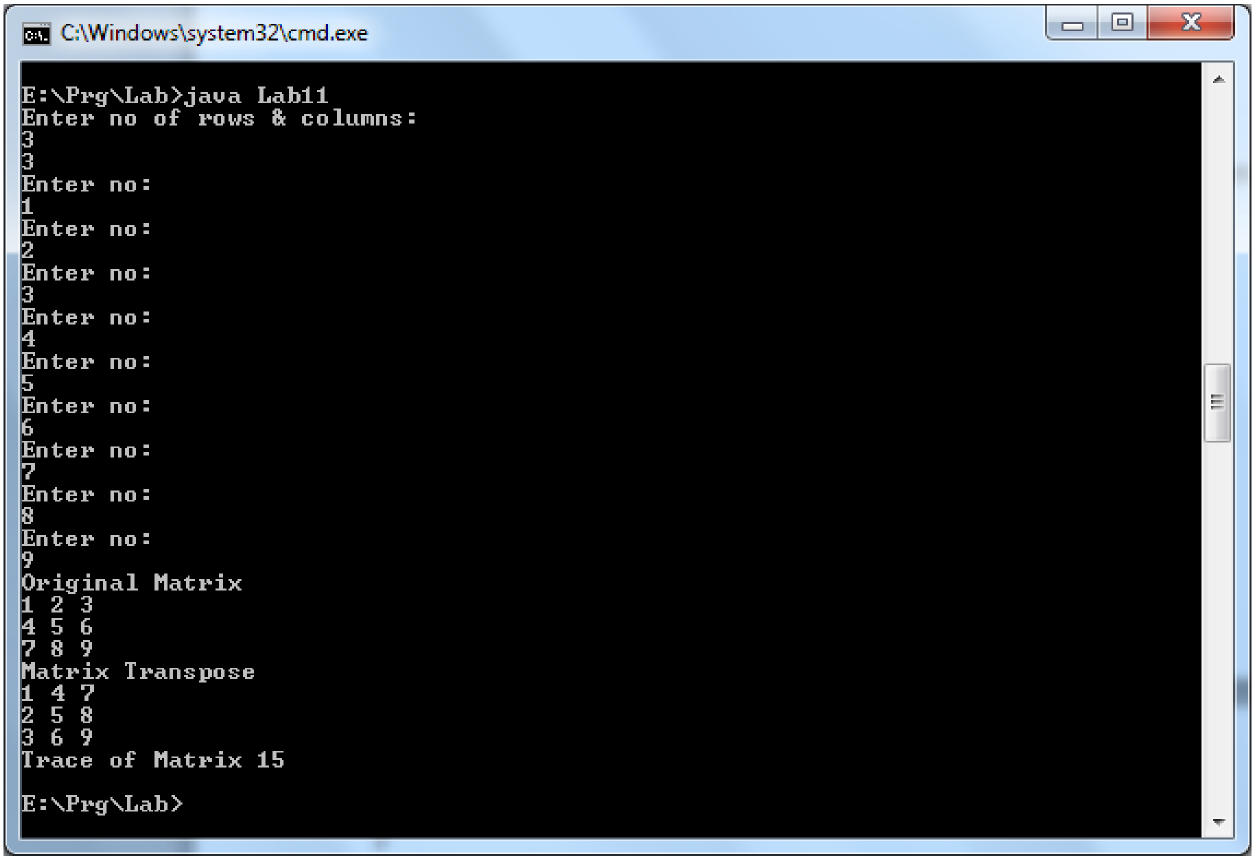
```
import java.util.Scanner;
import java.util.Scanner;
class Matrix
{
int a[][],i,j,r,c,trace;
Scanner obj1=new Scanner(System.in);
void read()
{
System.out.println("Enter no of rows & columns:");
r=obj1.nextInt();
c=obj1.nextInt();
a=new int[r][c];
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
System.out.println("Enter no:");
a[i][j]=obj1.nextInt();
}
}
}
void disp()
{
System.out.println("Original Matrix");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
System.out.print(a[i][j]+" ");
}
System.out.println();
}
}
void transpose()
{
System.out.println("Matrix Transpose");
for(i=0;i<c;i++)
{
for(j=0;j<r;j++)
{
System.out.print(a[j][i]+" ");
}
System.out.println();
}
}
void trace()
{
if(r==c)
{
for(i=0;i<c;i++)
{
for(j=0;j<r;j++)
{
if(i==j)
trace=trace+a[i][j];
}
}
System.out.println("Trace of Matrix "+trace);
}
```

```
else
System.out.println("Not a square matrix");
}
}
class Lab11
{
public static void main(String args[])
{
Matrix m1=new Matrix();
m1.read();
m1.disp();
m1.transpose();
m1.trace();
}
}
```

Matrix
a:int[][] r:int c:int sum:int trace:int

read()
disp()
transpose()
trace()

Lab11
main(args[]:String)



```
E:\Prg\Lab>java Lab11
Enter no of rows & columns:
3
3
Enter no:
1
Enter no:
2
Enter no:
3
Enter no:
4
Enter no:
5
Enter no:
6
Enter no:
7
Enter no:
8
Enter no:
9
Original Matrix
1 2 3
4 5 6
7 8 9
Matrix Transpose
1 4 7
2 5 8
3 6 9
Trace of Matrix 15

E:\Prg\Lab>
```

No:

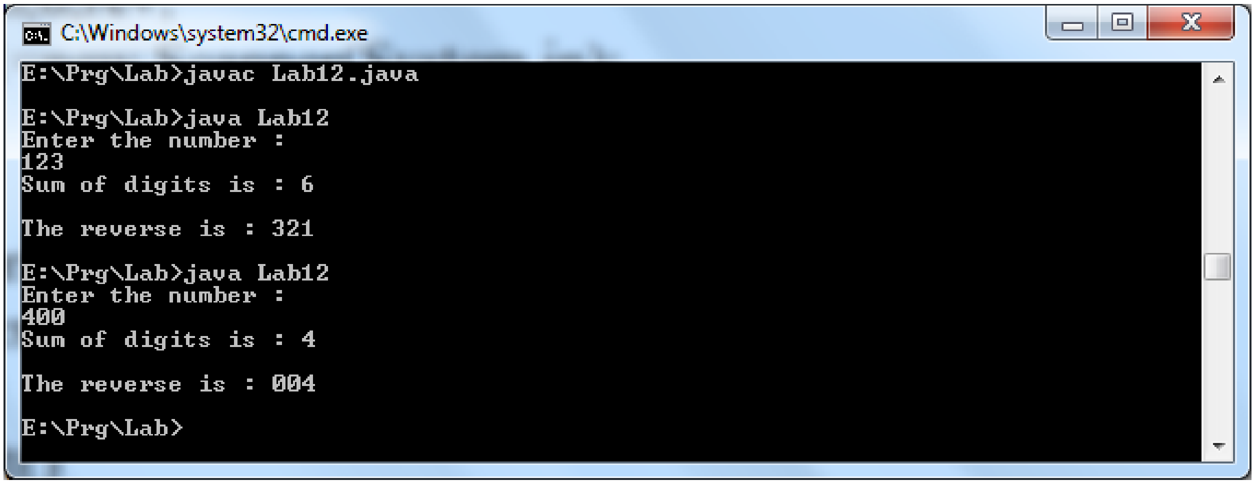
Aim : Write java program to find the sum of the digits and reverse of a given number using class and objects.

```
import java.util.Scanner;
class Number
{
int sum,n,num,d,rev;
Scanner obj1=new Scanner(System.in);
void read()
{
System.out.println("Enter the number :");
num=obj1.nextInt();
}
void sumdigit()
{
n=num;
while(n>0)
{ d=n%10;
  sum=sum+d;
  n=n/10;
 }
System.out.println("Sum of digits is : "+sum);
}
void reverse()
{
n=num;
System.out.print("\nThe reverse is : ");
while(n>0)
{
d=n%10;
rev=rev*10+d;
if(rev==0)
System.out.print(rev);
n=n/10;
}
System.out.println(rev);
}
}
class Lab12
{
public static void main(String args[])
{
Number n1=new Number();
n1.read();
n1.sumdigit();
n1.reverse();
}
}
```

Number
n:int sum:int rev:int d:int num:int

read()
sumdigit()
reverse()

Lab12
main(args[]:String)

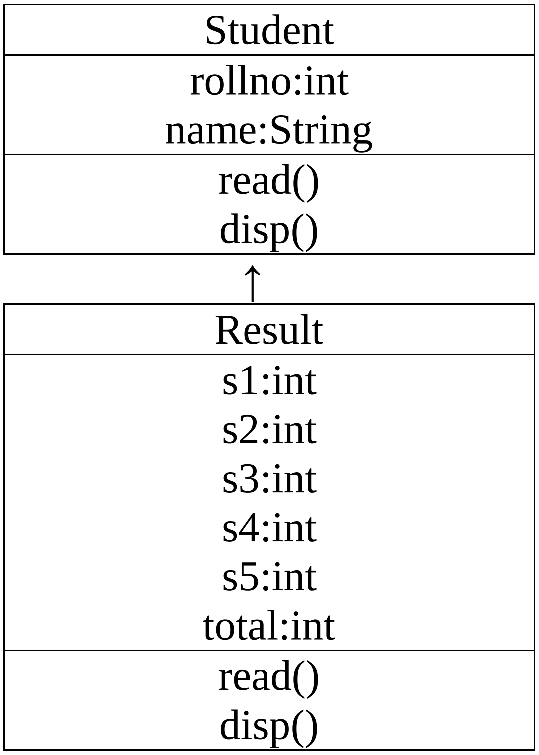


```
E:\Prg\Lab>javac Lab12.java

E:\Prg\Lab>java Lab12
Enter the number :
123
Sum of digits is : 6

The reverse is : 321

E:\Prg\Lab>java Lab12
Enter the number :
400
Sum of digits is : 4

The reverse is : 004

E:\Prg\Lab>
```

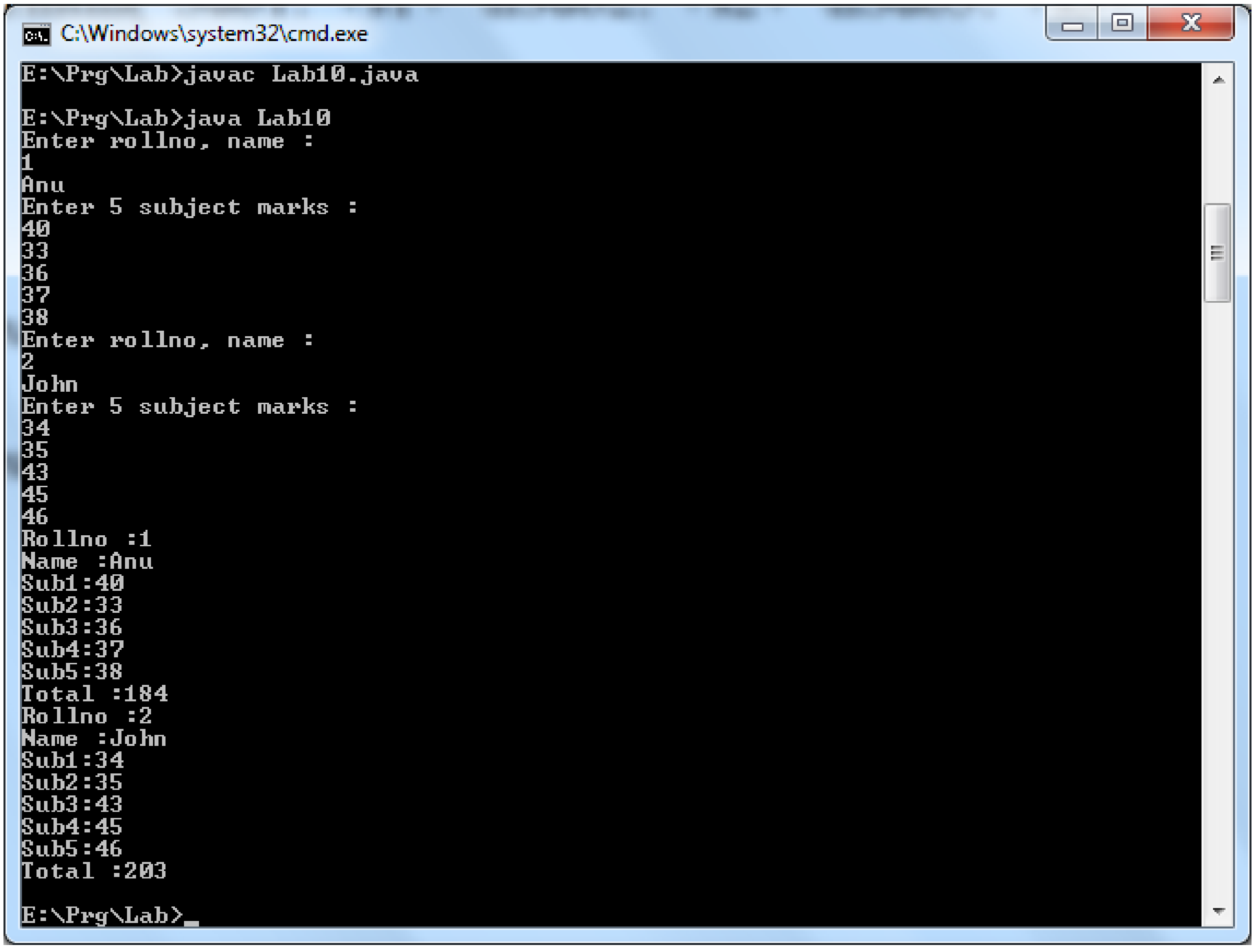
No:

Aim: Create a class student with methods to read and display the student details. Create a derived class result with methods to read marks of 5 subjects. Write a java program to display the total and grade of students, creating objects of class result.

```
import java.util.Scanner;
class Student
{
int rollno;
String name;
Scanner obj1=new Scanner(System.in);
void read()
{
System.out.println("Enter rollno, name :");
rollno=obj1.nextInt();
name=obj1.next();
}
void disp()
{
System.out.println("Rollno :"+rollno+"\nName :"+name);
}
}
class Result extends Student
{
int s1,s2,s3,s4,s5,total;
void read()
{
super.read();
System.out.println("Enter 5 subject marks :");
s1=obj1.nextInt();
s2=obj1.nextInt();
s3=obj1.nextInt();
s4=obj1.nextInt();
s5=obj1.nextInt();
total=s1+s2+s3+s4+s5;
}
void disp()
{
super.disp();
System.out.println("Sub1:"+s1+"\nSub2:"+s2+"\nSub3:"+s3+"\nSub4:"
+s4+"\nSub5:"+s5+"\nTotal :"+total);
}
}
class Lab10
{
public static void main(String args[])
{
Result r1=new Result(),r2=new Result();
r1.read();
r2.read();
r1.disp();
r2.disp();
}
}
```



Lab10
main(args[]:String)



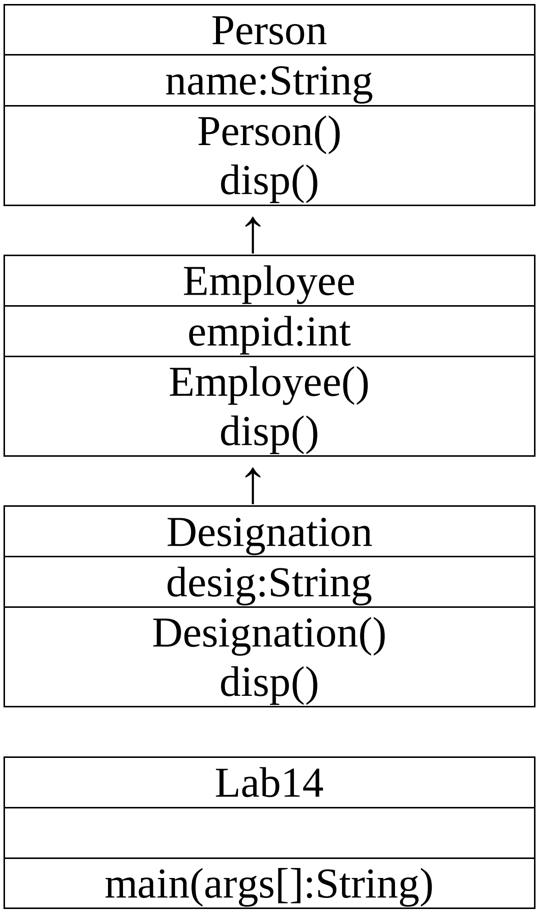
```
E:\Prg\Lab>javac Lab10.java

E:\Prg\Lab>java Lab10
Enter rollno, name :
1
Anu
Enter 5 subject marks :
40
33
36
37
38
Enter rollno, name :
2
John
Enter 5 subject marks :
34
35
43
45
46
Rollno :1
Name :Anu
Sub1:40
Sub2:33
Sub3:36
Sub4:37
Sub5:38
Total :184
Rollno :2
Name :John
Sub1:34
Sub2:35
Sub3:43
Sub4:45
Sub5:46
Total :203

E:\Prg\Lab>_
```

No:

Aim: Write a program to demonstrate the order in which constructors are invoked in multilevel inheritance

```
class Person
{
String name;
Person()
{
System.out.println("Person class constructor");
name="Anu";
}
void disp()
{
System.out.println("Name :"+name);
}
}
class Employee extends Person
```

```
{
int empid;
Employee()
{
System.out.println("Employee class constructor");
empid=1;
}
void disp()
{
System.out.println("Employee ID :"+empid);
super.disp();
}
}
class  Designation extends Employee
{
String desig;
Designation()
{
System.out.println("Designation class constructor");
desig="Manager";
}
void disp()
{
super.disp();
System.out.println("Designation:"+desig);
}
}
class Lab14
{
public static void main(String args[])
{
Designation d1=new Designation();
d1.disp();
}
}
```



Person

name:String

Person()
disp()

↑

Employee

empid:int

Employee()
disp()

↑

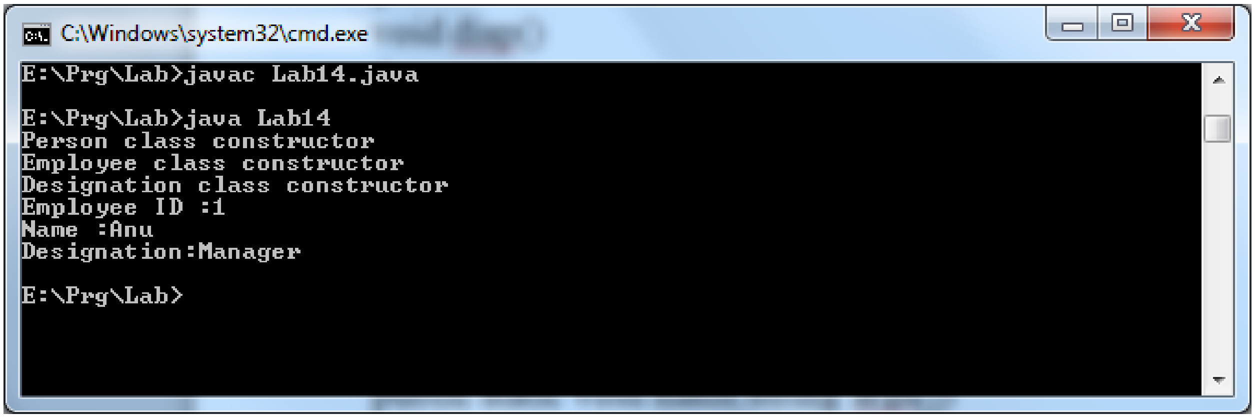
Designation

desig:String

Designation()
disp()

Lab14

main(args[]:String)



```
E:\Prg\Lab>javac Lab14.java

E:\Prg\Lab>java Lab14
Person class constructor
Employee class constructor
Designation class constructor
Employee ID :1
Name  :Anu
Designation:Manager

E:\Prg\Lab>
```

No:

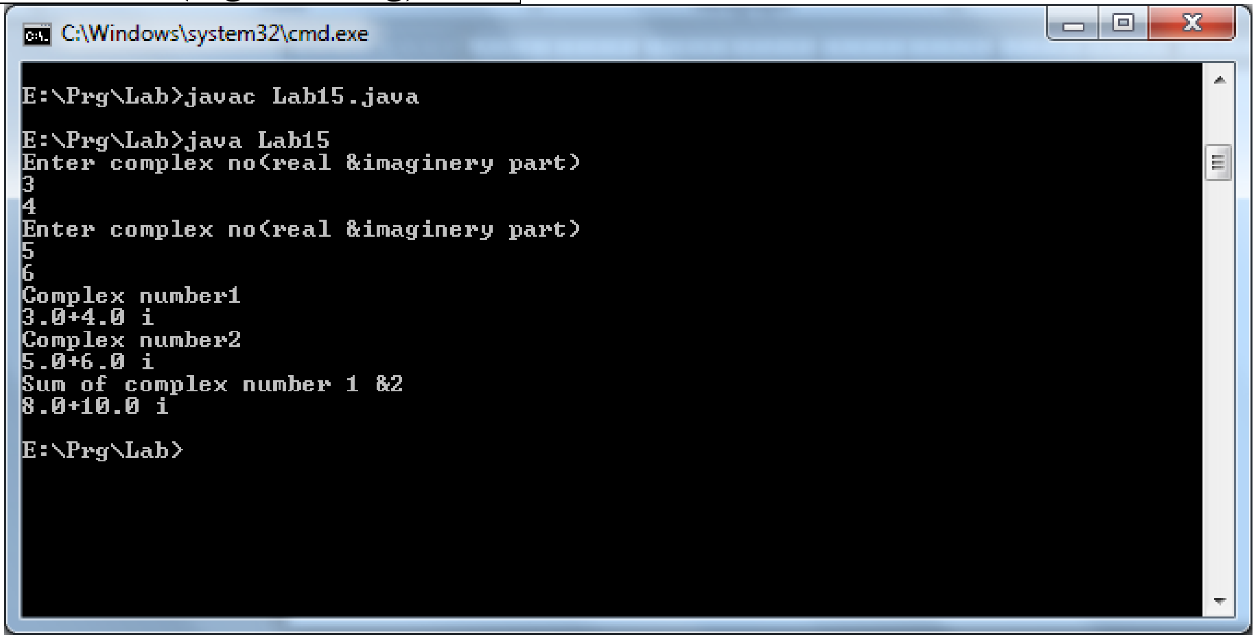
Aim: Using class and objects, write a java program to find the sum of two complex numbers (Hint: Use object as parameter to function).

```
import java.util.Scanner;
class Complex
{
float real,img;
Scanner obj1=new Scanner(System.in);
void read()
{
System.out.println("Enter complex no(real &imaginery part)");
real=obj1.nextFloat();
img=obj1.nextFloat();
}
void sum(Complex temp1,Complex temp2)
{
real=temp1.real+temp2.real;
img=temp1.img+temp2.img;
}
void disp()
{
System.out.println(real+"+"+img+" i");
}
}
class Lab15
{
public static void main(String args[])
{
Complex c1=new Complex();
Complex c2=new Complex();
Complex c3=new Complex();
c1.read();
c2.read();
c3.sum(c1,c2);
System.out.println("Complex number1");
c1.disp();
System.out.println("Complex number2");
c2.disp();
System.out.println("Sum of complex number 1 &2");
c3.sum(c1,c2);
c3.disp();
}
}
```

Complex
real:float img:float

read()
disp()
sum(temp1:Complex, temp2:Complex)

Lab15
main(args[]:String)



```
E:\Prg\Lab>javac Lab15.java

E:\Prg\Lab>java Lab15
Enter complex no(real &imaginery part)
3
4
Enter complex no(real &imaginery part)
5
6
Complex number1
3.0+4.0 i
Complex number2
5.0+6.0 i
Sum of complex number 1 &2
8.0+10.0 i

E:\Prg\Lab>
```

No:

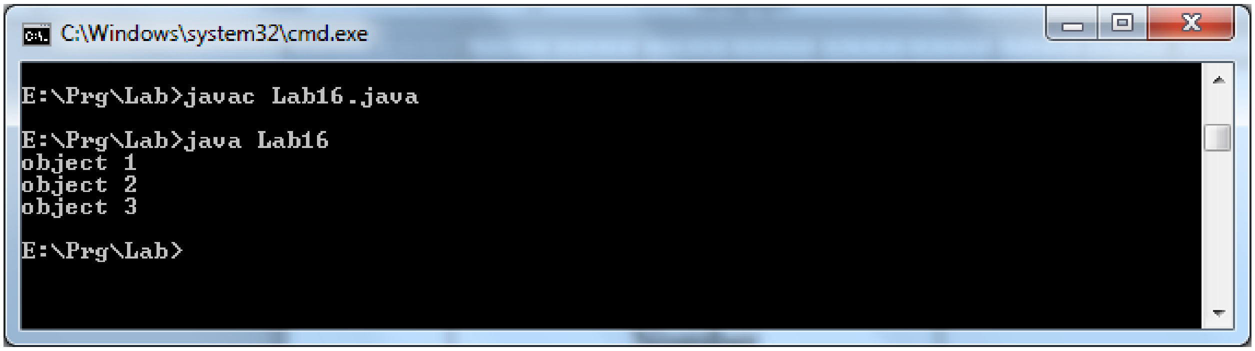
Aim : Write a program to count and display total number of objects created to a class (Hint: static members).

```
import java.util.Scanner;
class Number
{
static int count;
Scanner obj1=new Scanner(System.in);
Number()
{
count++;
}
void disp()
{
System.out.println("object "+count);
}
}
class Lab16
{
public static void main(String args[])
{
Number n1=new Number();
n1.disp();
Number n2=new Number();
n2.disp();
Number n3=new Number();
n3.disp();
}
}
```

Number
count:static int

Number()
disp()

Lab16
main(args[]:String)



```
E:\Prg\Lab>javac Lab16.java

E:\Prg\Lab>java Lab16
object 1
object 2
object 3

E:\Prg\Lab>
```

Q17

No:

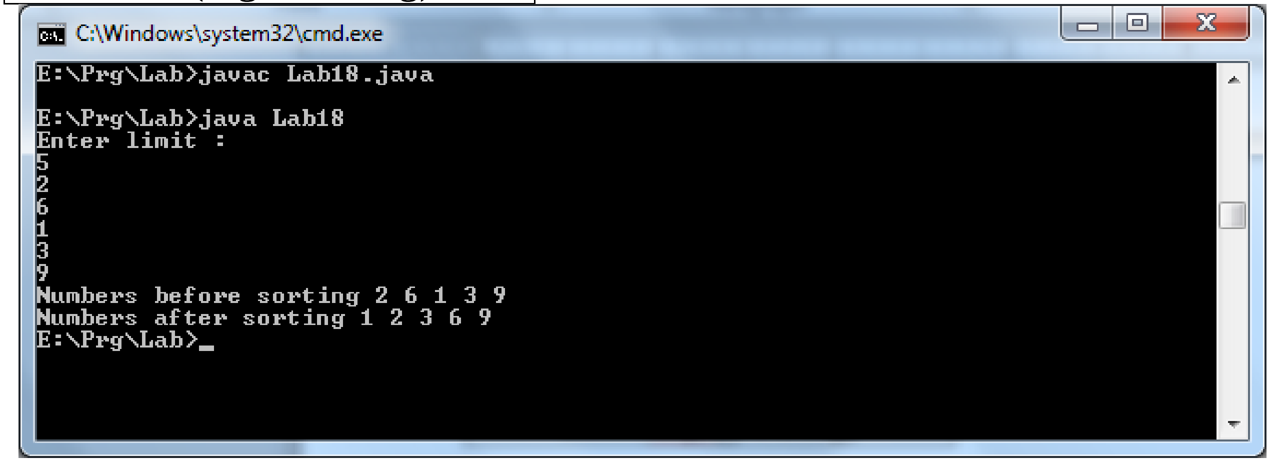
Aim: Write a program to sort a set of n numbers using a class.

```
import java.util.Scanner;
class Number
{
int a[],n;
Scanner obj1=new Scanner(System.in);
void read()
{
System.out.println("Enter limit :");
n=obj1.nextInt();
a=new int[n];
for(int i=0;i<n;i++)
{
a[i]=obj1.nextInt();
}
}
void sortasc()
{
int temp;
for(int i=0;i<n;i++)
{
for(int j=i+1;j<n;j++)
{
if(a[i]>a[j])
{
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
}
}
void disp()
{
for(int i=0;i<n;i++)
{
System.out.print(a[i]+" ");
}
}
}
class Lab18
{
public static void main(String args[])
{
Number n1=new Number();
n1.read();
System.out.print("Numbers before sorting ");
n1.disp();
n1.sortasc();
System.out.print("\nNumbers after sorting ");
n1.disp();
}  }
```

Number
a[]:int n:int

read()
sortasc()
disp()

Lab18
main(args[]:String)

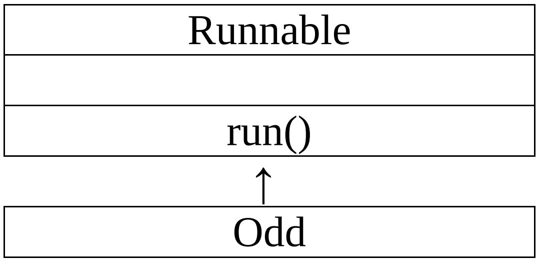


```
E:\Prg\Lab>javac Lab18.java

E:\Prg\Lab>java Lab18
Enter limit :
5
2
6
1
3
9
Numbers before sorting 2 6 1 3 9
Numbers after sorting 1 2 3 6 9
E:\Prg\Lab>_
```

No:

Aim: Write a multi thread java program for displaying odd numbers and even numbers up to a limit (Hint: Implement thread using Runnable interface).

```
import java.util.Scanner;
class Odd implements Runnable
{
Thread temp;
int i,n;
Odd(int t)
{
n=t;
temp=new Thread(this);
temp.start();
}
public void  run()
{
System.out.println("Odd numbers ");
for(i=1;i<=n;i=i+2)
{
System.out.println(i);
}
}
}
class Even extends Thread
{
Thread temp;
int i,n;
Even(int t)
{
n=t;
temp=new Thread(this);
temp.start();
}
public void  run()
{
System.out.println("Even numbers ");
for(i=2;i<=n;i=i+2)
{
System.out.println(i);
}
}
}
class Lab19
{
public static void main(String args[])
{
Scanner obj1=new Scanner(System.in);
System.out.println("Enter limit :");
int num=obj1.nextInt();
Odd o1=new Odd(num);
Even e1=new Even(num);
}
}
```



temp:Thread n:int i:int

Odd()
run()

Thread

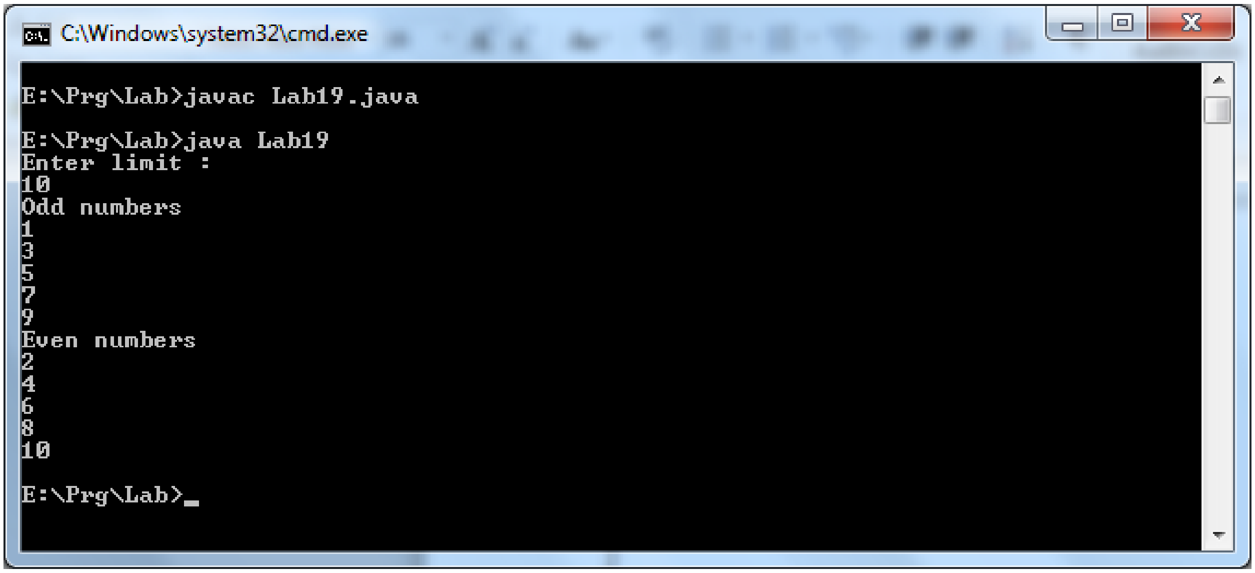
start()
run()

↑

Even
temp:Thread n:int i:int

Even()
run()

Lab19
main(args[]:String)

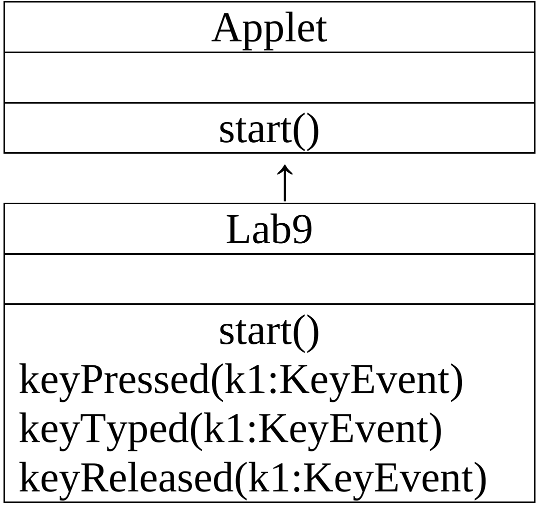


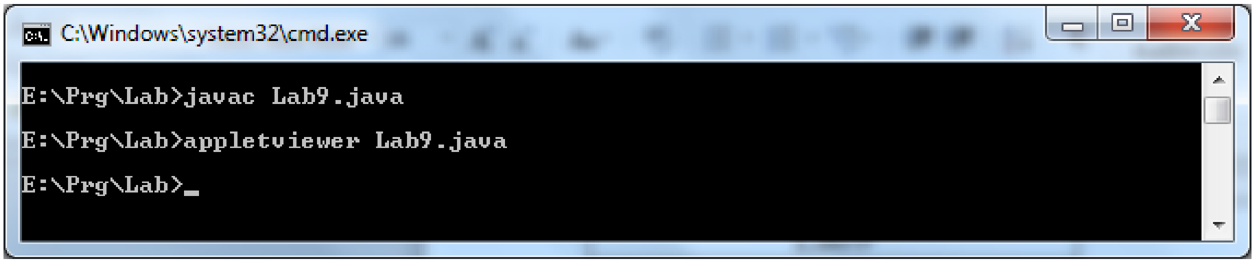
```
E:\Prg\Lab>javac Lab19.java

E:\Prg\Lab>java Lab19
Enter limit :
10
Odd numbers
1
3
5
7
9
Even numbers
2
4
6
8
10

E:\Prg\Lab>_
```

No:

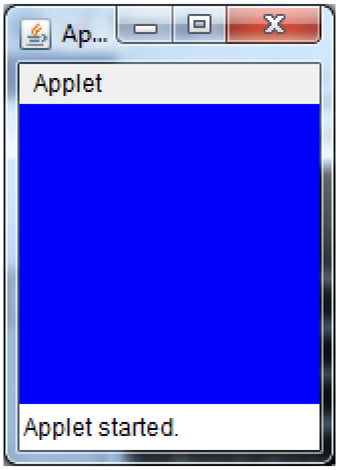
Aim: Write a program to track keyboard events on an applet.

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class Lab9  extends Applet implements KeyListener
{
public void start()
{
addKeyListener(this);
}
public void keyTyped(KeyEvent k1) // keys like CAPS Lock,Shift
{
 setBackground(Color.green);
}
public void keyPressed(KeyEvent k1)
{
 setBackground(Color.blue);
 }
public void keyReleased(KeyEvent k1)
{
 setBackground(Color.cyan);
}
 }
/* <applet code="Lab9" height="150" width="150"></applet>*/
```

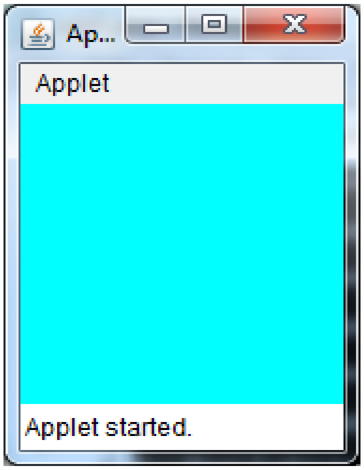




```
E:\Prg\Lab>javac Lab9.java

E:\Prg\Lab>appletviewer Lab9.java

E:\Prg\Lab>_
```



Applet started.



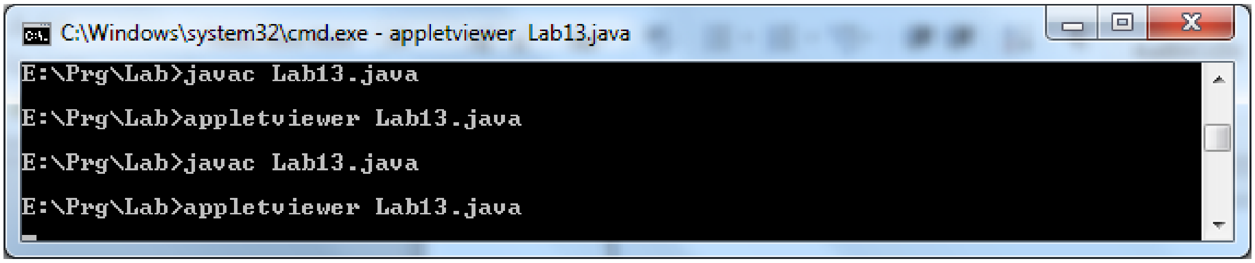
Applet started.

No:

Aim: Create an AWT application to add, remove items in a list

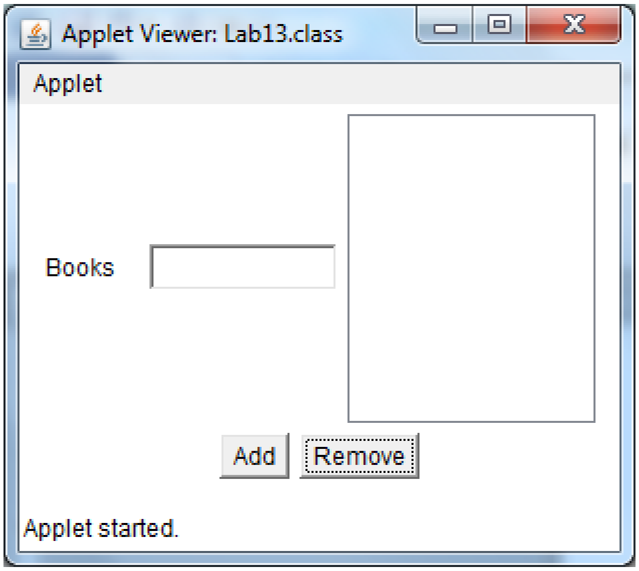
```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class Lab13 extends Applet implements ActionListener
{
Label l1;
TextField t1;
List c1,c2;
Button b1,b2;
public void init()
{
l1=new Label("Books");
t1=new TextField(10);
c1=new List(10,true);
b1=new Button("Add");
b2=new Button("Remove");
 }
public void start()
{
add(l1);
add(t1);
add(c1);
add(b1);
add(b2);
b1.addActionListener(this);
b2.addActionListener(this);
}
public void actionPerformed(ActionEvent a1)
{
```

```
if(a1.getSource()==b1)
{
c1.add(t1.getText());
t1.setText("");
}
else
{
String na[]=c1.getSelectedItems();
for(int i=0;i<na.length;i++)
  c1.remove(na[i]);
}
}
}
/*<applet code="Lab13.class" height="200" width="300"></applet>*/
```



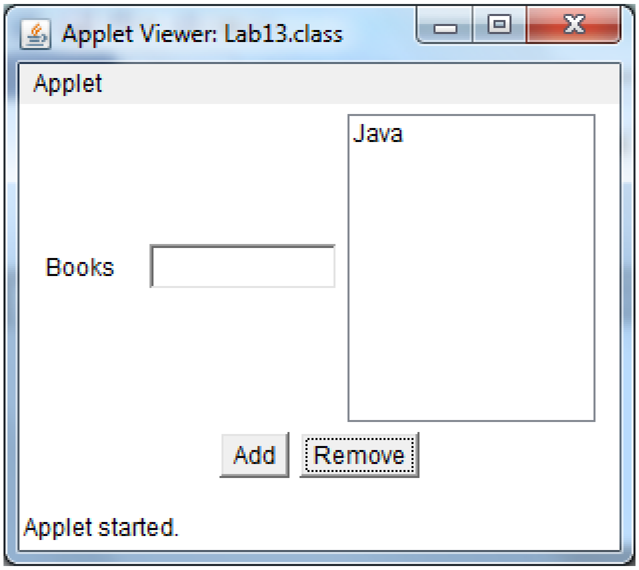
```
E:\Prg\Lab>javac Lab13.java

E:\Prg\Lab>appletviewer Lab13.java

E:\Prg\Lab>javac Lab13.java

E:\Prg\Lab>appletviewer Lab13.java
```

User interface



Applet started.

output



Applet started.