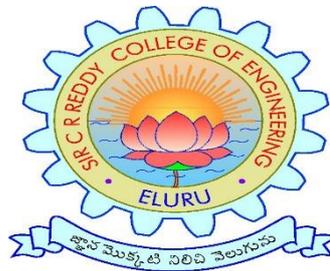


SIR C R REDDY COLLEGE OF ENGINEERING, ELURU

DEPARTMENT OF INFORMATION TECHNOLOGY

JAVA PROGRAMMING LAB

R19



Sri.V.Gopinath.,Asst.Prof

SUBJECT: JAVA PROGRAMMING LAB

CLASS 2/4 B.TECH IT II SEMISTER A.Y 2020-21

University program List

S.No	Program Statement
1	<p>Exercise - 1 (Basics)</p> <p>a). Write a JAVA program to display default value of all primitive data type of JAVA.</p> <p>b). Write a java program that display the roots of a quadratic equation $ax^2+bx=0$. Calculate thediscriminate D and basing on value of D, describe the nature of root.</p> <p>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.</p> <p>d) Write a case study on public static void main(250 words)</p>
2	<p>Exercise - 2 (Operations, Expressions, Control-flow, Strings)</p> <p>a). Write a JAVA program to search for an element in a given list of elements using binary search mechanism.</p> <p>b). Write a JAVA program to sort for an element in a given list of elements using bubble sort</p> <p>(c). Write a JAVA program to sort for an element in a given list of elements using merge sort.</p> <p>(d) Write a JAVA program using StringBufferto delete, remove character.</p>
3	<p>Exercise - 3(Class, Objects)</p> <p>a). Write a JAVA program to implement class mechanism. – Create a class, methods and invoke them inside main method.</p> <p>b). Write a JAVA program to implement constructor.</p>
4	<p>Exercise - 4 (Methods)</p> <p>a). Write a JAVA program to implement constructor overloading.</p> <p>b). Write a JAVA program implement method overloading.</p>
5	<p>Exercise - 5 (Inheritance)</p> <p>a). Write a JAVA program to implement Single Inheritance</p> <p>b). Write a JAVA program to implement multi level Inheritance</p> <p>c). Write a java program for abstract class to find areas of different shapes</p>
6	<p>Exercise - 6 (Inheritance - Continued)</p> <p>a). Write a JAVA program give example for “super” keyword.</p> <p>b). Write a JAVA program to implement Interface. What kind of Inheritance can be achieved?</p>
7	<p>Exercise - 7 (Exception)</p> <p>a).Write a JAVA program that describes exception handling mechanism</p> <p>b).Write a JAVA program Illustrating Multiple catch clauses</p>
8	<p>Exercise – 8 (Runtime Polymorphism)</p> <p>a). Write a JAVA program that implements Runtime polymorphism</p>

	b). Write a Case study on run time polymorphism, inheritance that implements in above problem
9	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). Write a JAVA program for creation of User Defined Exception
10	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.
11	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
12	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
13	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.
14	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.
15	Exercise - 15 (Swings) a). Write a JAVA program to build a Calculator in Swings b). Write a JAVA program to display the digital watch in swing tutorial
16	Exercise – 16 (Swings - Continued) a). Write a JAVA program that to create a single ball bouncing inside a JPanel. b). Write a JAVA program JTree as displaying a real tree upside down.

Content beyond the syllabus:

The programs 1.d), 15, 16 are intended to enhance the student skills which are not included in JNTU Kakinada curriculum.

1) Write a JAVA Program to display default value of all primitive data types of JAVA**Aim: Displaying default value of all primitive data types of JAVA**

Description:

Primitive Data Types: The Java programming language is statically-typed, which means that all variables must first be declared before they can be used.

The eight primitive data types supported by the Java programming language are:

Data Type	Default Value (for fields)
Byte	0
Short	0
Int	0
Long	0L
Float	0.0f
Double	0.0d
Char	'\u0000'
String (or any object)	null
Boolean	false

Program:

```
class DefaultValues
{
    static byte b;

    static short s;

    static int i;

    static long l;

    static float f;

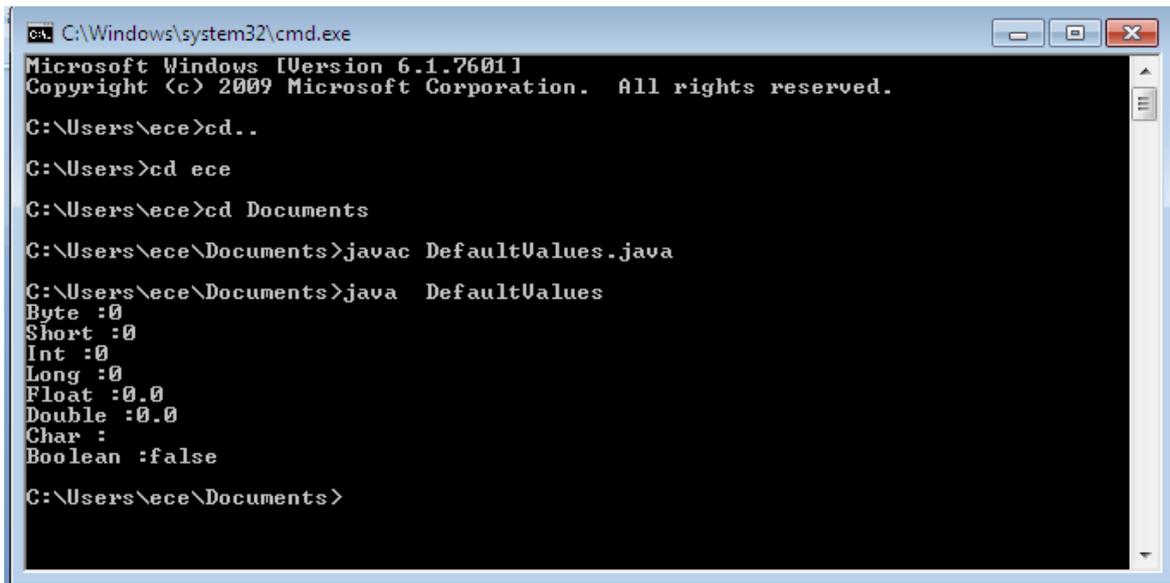
    static double d;

    static char c;

    static boolean bl;

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

```
System.out.println("Byte :"+b);  
System.out.println("Short :"+s);  
System.out.println("Int :"+i);  
System.out.println("Long :"+l);  
System.out.println("Float :"+f);  
System.out.println("Double :"+d);  
System.out.println("Char :"+c);  
System.out.println("Boolean :"+bl);  
}  
}
```

Output:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\ece>cd..
C:\Users>cd ece
C:\Users\ece>cd Documents
C:\Users\ece\Documents>javac DefaultValues.java
C:\Users\ece\Documents>java DefaultValues
Byte :0
Short :0
Int :0
Long :0
Float :0.0
Double :0.0
Char :
Boolean :false

C:\Users\ece\Documents>
```

Sample Viva questions:

- 1) What is Precision?
- 2) What is Unicode?
- 3) What is the use of unicode?
- 4) What is the minimum maximum value of each primitive data type?
- 5) Fill the blank with apt datatype.
byte a=10,b=5,c; c=(_____)a+b;
- 6) what are the reference types in java?
- 7) what is type casting?
- 8) What is the size of char & short data type?
- 9) What is the size of int & long data type?
- 10) What is the size of float & double data type?
- 11) What are possible values of Boolean variables in JAVA?

2) Write a JAVA program to display the roots of quadratic equation $ax^2+bx+c=0$. Calculate the discriminant D and basing on the value of D, Describe the nature of roots

Aim: To display roots of quadratic equation

Description: A quadratic equation is any equation having the form $ax^2+bx+c=0$ where x represents an unknown, and a, b, and c represent numbers such that a is not equal to 0.

Program:

```
import java.io.*;

class Quadratic

{

    public static void main(String args[])throws IOException

    {

        double x1,x2,disc,a,b,c;

        InputStreamReader obj=new InputStreamReader(System.in);

        BufferedReader br=new BufferedReader(obj);

        System.out.println("enter a,b,c values");

        a=Double.parseDouble(br.readLine());

        b=Double.parseDouble(br.readLine());

        c=Double.parseDouble(br.readLine());

        disc=(b*b)-(4*a*c);

        if(disc==0)

        {

            System.out.println("roots are real and equal ");

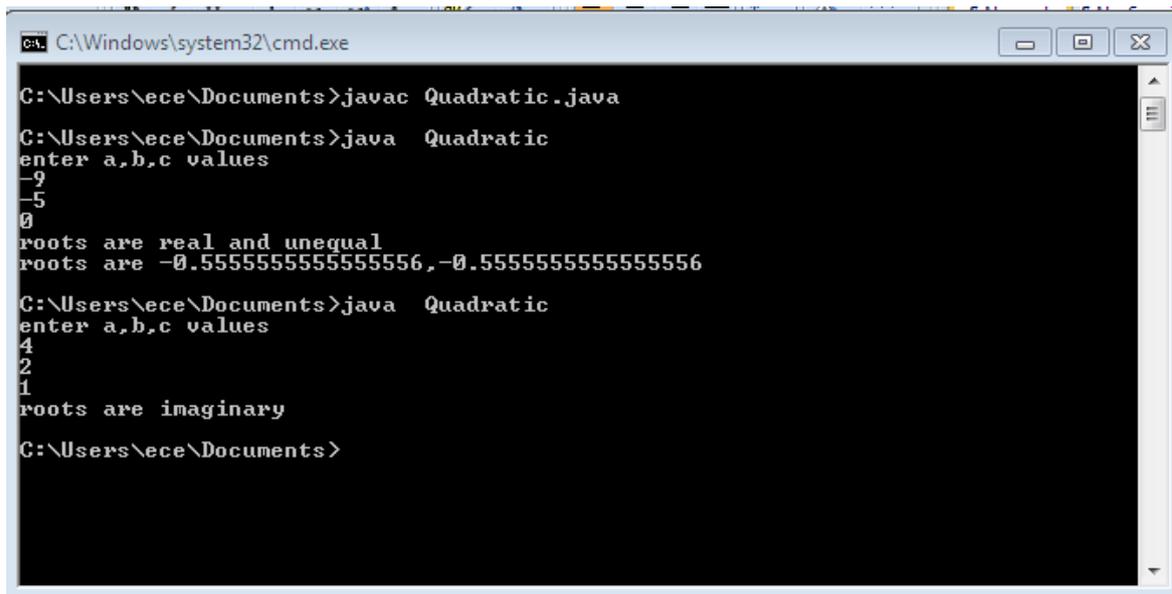
            x1=x2=-b/(2*a);

            System.out.println("roots are "+x1+", "+x2);

        }

        else if(disc>0)
```

```
        {  
            System.out.println("roots are real and unequal");  
            x1=(-b+Math.sqrt(disc))/(2*a);  
            x2=(-b+Math.sqrt(disc))/(2*a);  
            System.out.println("roots are "+x1+", "+x2);  
        }  
    else  
    {  
        System.out.println("roots are imaginary");  
    }  
}  
}
```

Output:

```
ca: C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>javac Quadratic.java
C:\Users\ece\Documents>java Quadratic
enter a,b,c values
-9
-5
0
roots are real and unequal
roots are -0.5555555555555556,-0.5555555555555556
C:\Users\ece\Documents>java Quadratic
enter a,b,c values
4
2
1
roots are imaginary
C:\Users\ece\Documents>
```

Sample viva questions:

- 1) What is WORA
- 2) Which class is used for mathematical related operations?
- 3) List out the ten useful methods for mathematical operations
- 4) Which package is imported by default to java program?
- 5) What do you mean by imaginary roots?
- 6) What is a JIT compiler?
- 7) What is a JRE?
- 8) Is java both compiled and interpreted language?
- 9) What is the difference between #include and import statement?
- 10) Name few conditional statements available in JAVA?
- 11) What are the arithmetical operators available in JAVA?

3) 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.

Aim: Displaying the speed of qualifying racers.

Description:

Calculate the average speed of all the racers and identify the speed of qualifying racers.

Speed=distance X time

Program:

```
import java.io.*;
```

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

```
class Race1
```

```
{
```

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

```
    {
```

```
        int i;
```

```
        int flag=0 ;
```

```
        float avg = 0.0f;
```

```
        DataInputStream dis=new DataInputStream(System.in);
```

```
        System.out.println("enter the speed of five racers");
```

```
        int a[]=new int[5];
```

```
        for(i=0;i<5;i++)
```

```
        {
```

```
            a[i]=Integer.parseInt(dis.readLine());
```

```
            avg += a[i];
```

```
        }
```

```
        System.out.println(" the speeds of five racers are follows");
```

```
        for(i=0;i<5;i++)
```

```
        {
```

```
            System.out.println("The speed of racer" +(i+1)+ "is "+a[i]);
```

```
        }
```

```
for(i=0;i<4;i++)
{
    if(a[i] != a[i+1])
    {
        flag=1;
        break;
    }
}
if(flag!=0)
{
    avg=avg/5.0f;
    System.out.println("average is"+avg);
    for(i=0;i<5;i++)
    {
        if(a[i]>avg)
        {
            System.out.println("The winner of the racer is player
"+(i+1));
        }
    }
}
else
    System.out.println("All the Racers are at same Speed \n No one will
be the winner!!!");
}
}
```

Output:

```
C:\Users\cse1ab3\Documents>java Race
enter the speed of five racers
1
0
5
8
9
The winner of the racer is player 2
C:\Users\cse1ab3\Documents>
```

Sample viva questions

- 1) Name iterative statements in JAVA?
- 2) Differentiate between pre test and post test loops?
- 3) What are the relational operators available in JAVA?
- 4) What are the logical operators available in JAVA?
- 5) List few packages available in JAVA library?
- 6) What is an infinite Loop? How infinite loop is declared?
- 7) What is the difference between double and float variables in Java?
- 8) Which class is used to take console input?
- 9) Which methods are used to read data from console?

4) Write a case study on public static void main (250 words)**Aim: To illustrate the keywords of main function in java.****Explanation:****Public static void main(String args[]) { }**

The main part of the statement is the method. That is, it is the block of code that is actually going to display our output. Therefore, it needs to be included within a class. Even though this is the main method, it may be hard to notice since the keyword is buried in the other keywords. Each and every keyword present in this method must be known.

public static void main(String args[]) – Java program processing starts from the main() method which is a mandatory part of every Java program.

Public

The first word in the statement, public, means that any object can use the main method.

The first word could also be static, but public static is the standard way. Still, public means that it is truly the main method of the program and the means by which all other code will be used in the program.

Static

Even though you could write static public or public static, the program will work just fine. Still, it's more common to use public static. In the context of Java, the static keyword means that the main method is a class method. What this also means is that you can access this method without having an instance of the class (Remember that the public keywords makes the method accessible by all classes, so this makes it possible.)

Void : is used to define the Return Type of the Method. It defines what the method can return. Void means the Method will not return any value.

main: is the name of the Method. This Method name is searched by JVM as a starting point for an application with a particular signature only.

String args[] : is the parameter to the main Method.

Sample viva questions:

- 1) What is a method?
- 2) What is the prototype of a method?
- 3) Prototype of a class
- 4) What is the default input type of Java?
- 5) What does void mean?
- 6) Differentiate between Top-Down approach and Bottom-up approach.
- 7) What is access modifier?
- 8) Is public static void main(int k) valid?give reason.

5) Write a JAVA Program to search for an element in a given list of elements (Binary search)

Aim: To search for an element in a given list of elements

Description:

Binary search provides a best optimum solution to search for an element. Binary search logic

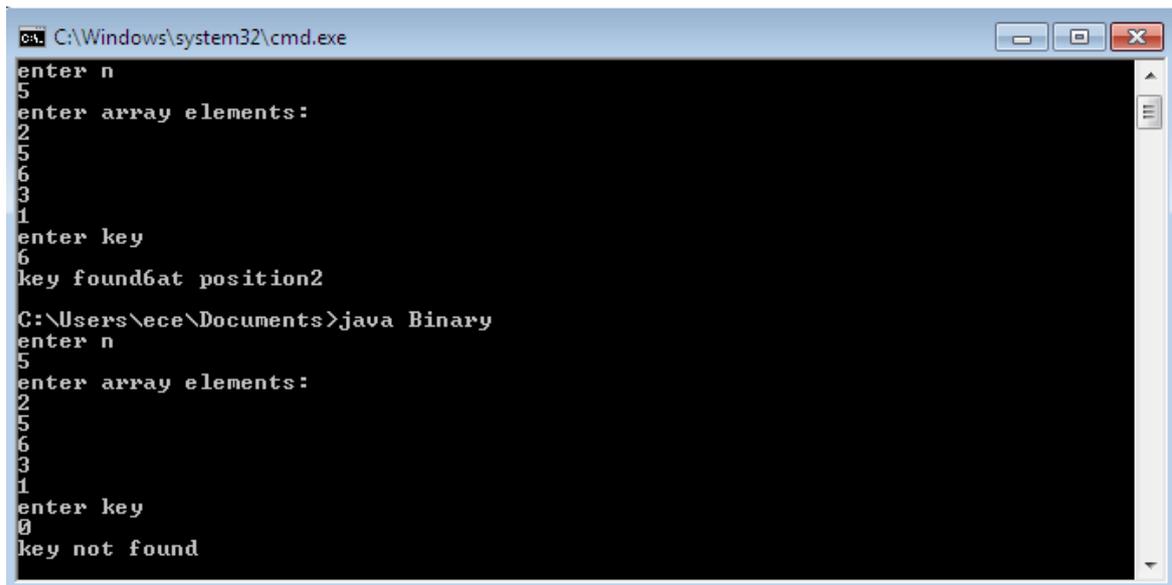
Let's consider our list has below sorted values and we want to search for an element 45.
{23, 45, 67, 87, 98}

- Get the center element from the array and in this case it is 67.
- Compare center element with the key element i.e.; in this case compare 67 with 45
- Since the center element is greater than the key element, we're sure that the key element is in the first half of the list of elements because the array has already been sorted.
- If the center element is less than the key element, the element is in the second half of the list of elements.
- Consider the list of values as either first half or second half and repeat the above steps until element is found or all the array of elements are checked.

Program:

```
import java.io.*;
class Binary
{
    public static void main(String args[]) throws IOException
    {
        int i,j,n,key,l,h,m;
        DataInputStream dis=new DataInputStream(System.in);
        System.out.println("enter n");
        n=Integer.parseInt(dis.readLine());
        int a[]=new int[n];
        System.out.println("enter array elements:");
        for(i=0;i<n;i++)
        {
            a[i]=Integer.parseInt(dis.readLine());
        }
        System.out.println("enter key");
        key=Integer.parseInt(dis.readLine());
        l=0;
        h=n-1;
        m=(l+h)/2;
        while(l<=h)
        {
            if(a[m]<key)
            {
                l=m+1;
            }
            else if(a[m]==key)
            {
                System.out.println("key found"+key+"at position"+m);
                break;
            }
        }
    }
}
```

```
        else
        {
            h=m-1;
        }
        m=(l+h)/2;
    }
    if(l>h)
        System.out.println("key not found");
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe
enter n
5
enter array elements:
2
5
6
3
1
enter key
6
key found at position 2
C:\Users\ece\Documents>java Binary
enter n
5
enter array elements:
2
5
6
3
1
enter key
0
key not found
```

Sample viva questions:

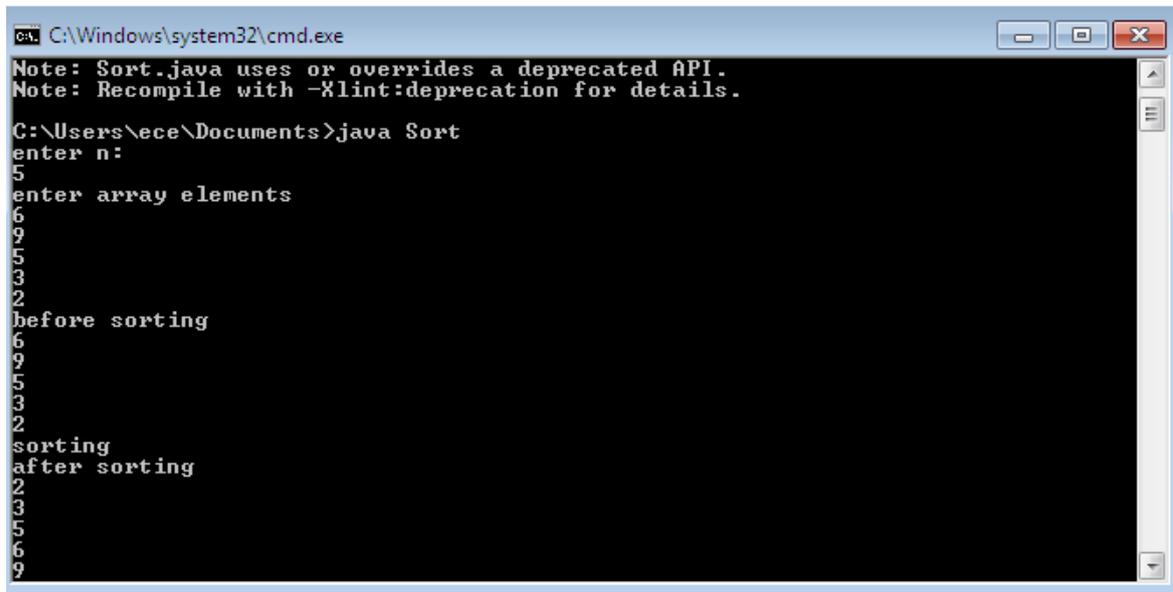
- 1) What is the difference between entry and exit controlled loops?
- 2) What is the difference between for and while loop?
- 3) What are instance variables?
- 4) What is the advantage of Binary search over linear search?
- 5) What is the time complexity of Binary search?
- 6) How to define a constant variable in Java?
- 7) What is the argument of main() method?
- 8) What are the classes used to convert a string into primitive datatypes?
- 9) What is a wrapper class?
- 10) How to convert a primitive datatype into a string?

6) Write a JAVA program to sort for an element in a given list of elements using bubble sort**Aim: To sort the given list using bubble sort****Description:**

Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order.

Program:

```
import java.io.*;
class Sort
{
    public static void main(String args[]) throws IOException
    {
        int i,j,n,temp;
        DataInputStream dis=new DataInputStream(System.in);
        System.out.println("enter n:");
        n=Integer.parseInt(dis.readLine());
        int a[]=new int[n];
        System.out.println("enter array elements");
        for(i=0;i<n;i++)
        {
            a[i]=Integer.parseInt(dis.readLine());
        }
        System.out.println("before sorting");
        for(i=0;i<n;i++)
        {
            System.out.println(a[i]);
        }
        System.out.println("sorting");
        for(i=0;i<n;i++)
        {
            for(j=i+1;j<n;j++)
            {
                if(a[i]>a[j])
                {
                    temp=a[i];
                    a[i]=a[j];
                    a[j]=temp;
                }
            }
        }
        System.out.println("after sorting");
        for(i=0;i<n;i++)
        {
            System.out.println(a[i]);
        }
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe
Note: Sort.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\ece\Documents>java Sort
enter n:
5
enter array elements
6
9
5
3
2
before sorting
6
9
5
3
2
sorting
after sorting
2
3
5
6
9
```

Sample viva questions:

- 1) What is the need of sorting?
- 2) What are the different sorting algorithms available?
- 3) What is a time complexity for various sorting algorithms?
- 4) Which algorithm will be best for sorting in terms of time complexity?
- 5) What is the time complexity of proposed algorithm?
- 6) Among bubble and quick sort, which is the best one? why?

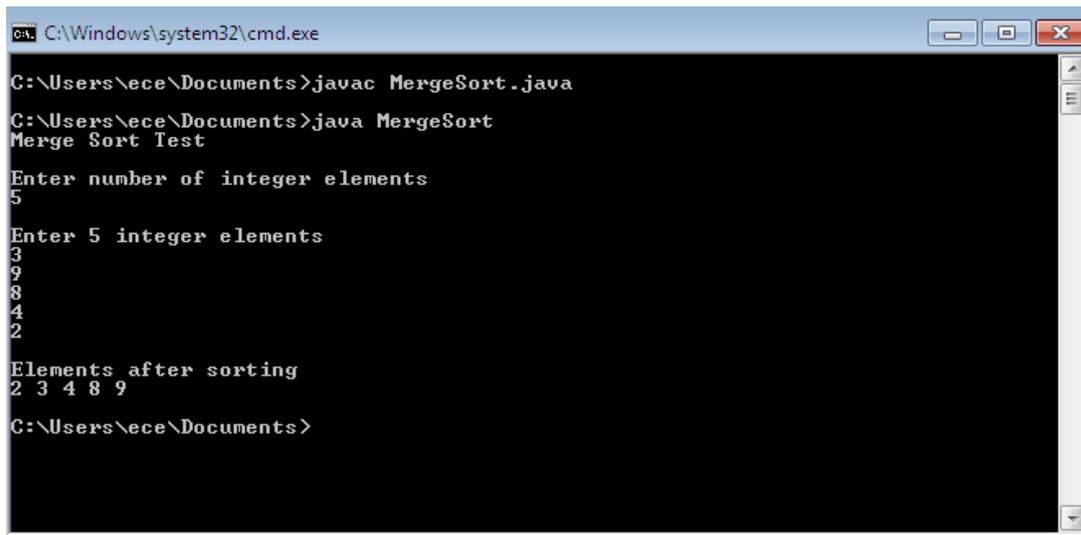
7) Write a JAVA program to sort for an element in a given list of elements using merge sort.**Aim: To sort the given list using merge sort****Description:**

Merge Sort is a Divide and Conquer algorithm. It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves. The merge() function is used for merging two halves. The merge(arr, l, m, r) is key process that assumes that arr[l..m] and arr[m+1..r] are sorted and merges the two sorted sub-arrays into one.

Program:

```
import java.util.Scanner;
public class MergeSort
{
    public static void sort(int[] a, int low, int high)
    {
        int N = high - low;
        if (N <= 1)
            return;
        int mid = low + N/2;
        sort(a, low, mid);
        sort(a, mid, high);
        int[] temp = new int[N];
        int i = low, j = mid;
        for (int k = 0; k < N; k++)
        {
            if (i == mid)
                temp[k] = a[j++];
            else if (j == high)
                temp[k] = a[i++];
            else if (a[j]<a[i])
                temp[k] = a[j++];
            else
                temp[k] = a[i++];
        }
        for (int k = 0; k < N; k++)
            a[low + k] = temp[k];
    }
    public static void main(String[] args)
    {
        Scanner scan = new Scanner( System.in );
        System.out.println("Merge Sort Test\n");
        int n, i;
        System.out.println("Enter number of integer elements");
        n = scan.nextInt();
        int arr[] = new int[ n ];
        System.out.println("\nEnter "+ n +" integer elements");
        for (i = 0; i < n; i++)
            arr[i] = scan.nextInt();
        sort(arr, 0, n);
    }
}
```

```
System.out.println("\nElements after sorting ");
for (i = 0; i < n; i++)
    System.out.print(arr[i]+" ");
System.out.println();
}
}
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\nece\Documents>javac MergeSort.java
C:\Users\nece\Documents>java MergeSort
Merge Sort Test
Enter number of integer elements
5
Enter 5 integer elements
3
9
8
4
2
Elements after sorting
2 3 4 8 9
C:\Users\nece\Documents>
```

Sample viva questions:

- 1) What is recursion?
- 2) What are the types of recursion?
- 3) What is the time complexity of merge sort?
- 4) What is best case, worst case and average case?
- 5) What strategy does merge sort follow? What type of sorting it is?
- 6) What are the types of sorting?

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

Aim: To delete and remove some of the characters from a string using StringBuffer class

Description:

Java StringBuffer class is used to create mutable (modifiable) string. The StringBuffer class in java is same as String class except it is mutable i.e. it can be changed.

Important Constructors of StringBuffer class

Constructor	Description
StringBuffer()	creates an empty string buffer with the initial capacity of 16.
StringBuffer(String str)	creates a string buffer with the specified string.
StringBuffer(int capacity)	creates an empty string buffer with the specified capacity as length.

Important methods of StringBuffer class

public synchronized StringBuffer delete(int startIndex,int endIndex)	is used to delete the string from specified startIndex and endIndex.
public char charAt(int index)	is used to return the character at the specified position.

Program:

```
Import java.io.*;
```

```
public class JavaStringBufferDeleteExample {
```

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

```
StringBuffer sb1 = new StringBuffer("Hello World");
```

```
sb1.delete(0,6);
```

```
System.out.println(sb1);
```

```
StringBuffer sb2 = new StringBuffer("Some Content");
```

```
System.out.println(sb2);
```

```
sb2.delete(0, sb2.length());
```

```
System.out.println(sb2);
```

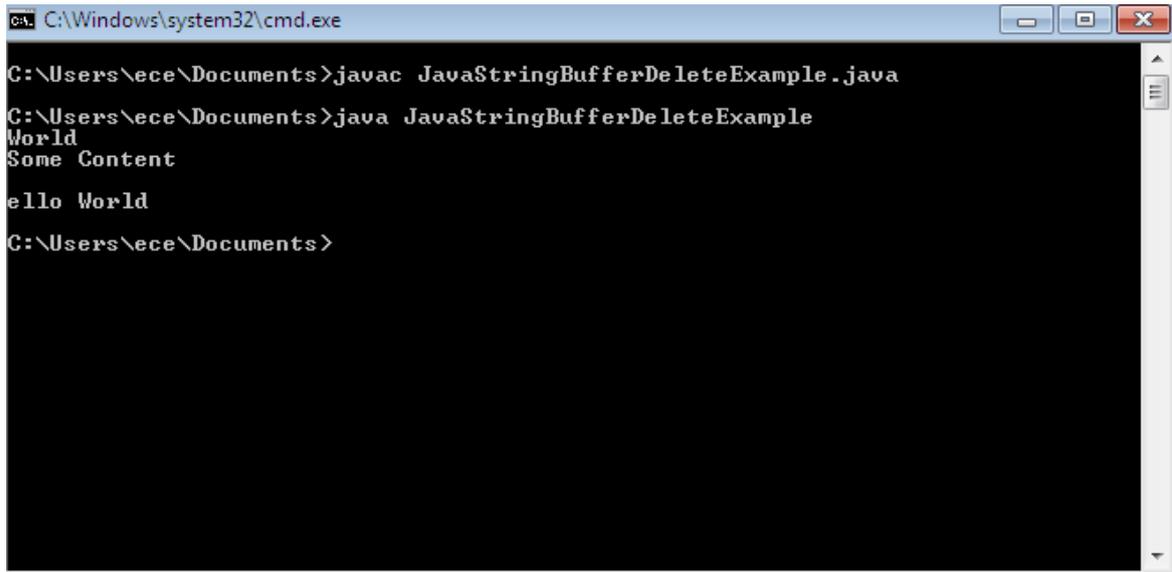
```
StringBuffer sb3 = new StringBuffer("Hello World");
```

```
sb3.deleteCharAt(0);
```

```
System.out.println(sb3);
```

```
}
```

```
}
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>javac JavaStringBufferDeleteExample.java
C:\Users\ece\Documents>java JavaStringBufferDeleteExample
World
Some Content
ello World
C:\Users\ece\Documents>
```

Sample viva questions:

- 1) what type of object does StringBuffer is?
- 2) What si the difference between mutable and immutable class?
- 3) Which method of class StringBuffer is used to find the length of current character sequence?
- 4) Which method of class StringBuffer is used to concatenate the string representation to the end of invoking string?
- 5) What do you mean by System.out in java?
- 6) What is the advantage of StringBuffer over String class ?
- 7) List out the methods of StringBuffer apart from String.
- 8) What is the default capacity of StringBuffer?
- 9) What is the maximum capacity of String and StringBuffer?
- 10) What is StringBuiler?

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

Aim: A Java Program to demonstrate class,method.

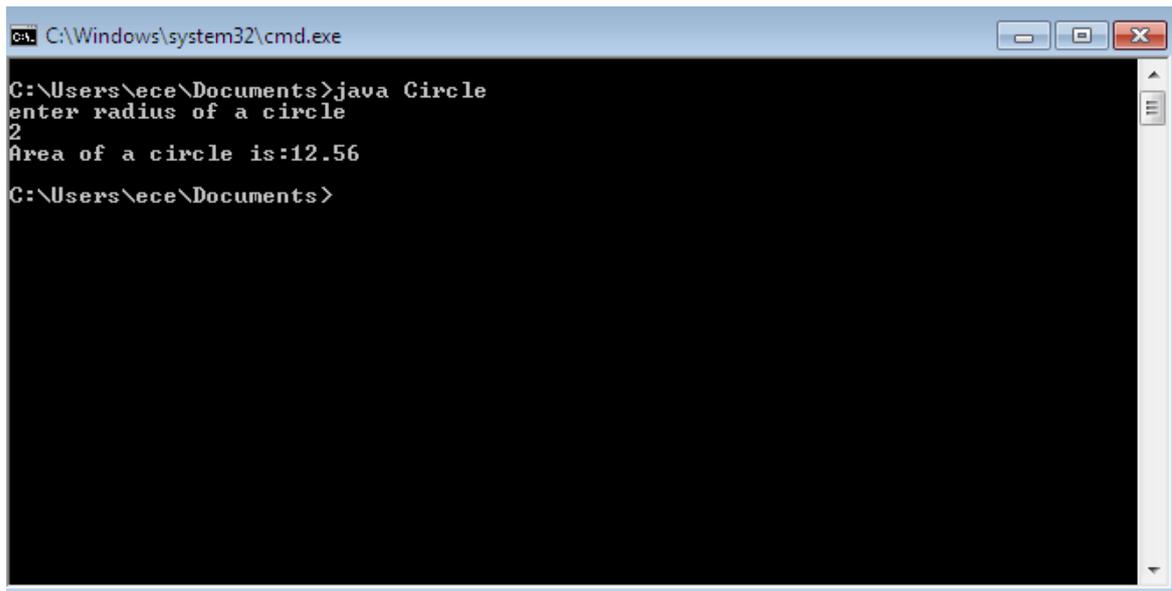
Description:

A class is a collection of member variables and member methods.To invoke the class members we need an object.

Program:

```
import java.io.*;
class Area
{
double a;
public double area(double r)
{
a=3.14*r*r;
return a;
}

}
class Circle
{
public static void main(String args[]) throws IOException
{
Area ob=new Area();
DataInputStream dis=new DataInputStream(System.in);
System.out.println("enter radius of a circle");
double r=Integer.parseInt(dis.readLine());
double ar=ob.area(r);
System.out.println("Area of a circle is:"+ar);
}
}
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>java Circle
enter radius of a circle
2
Area of a circle is:12.56
C:\Users\ece\Documents>
```

Sample viva questions:

- 1) Draw the class diagram for a class?
- 2) What are the features of OOP?
- 3) What is the structure of Java file?
- 4) 1)Do we need to create object for the main class?
- 5) Can a main() method be declared final?
- 6) Can a source file contain more than one class declaration?

10) Write a JAVA program to implement constructor.**Aim: A Java program on Constructors****Description:**

Java constructors are special methods that are called when an object is instantiated. In other words, when you use the new keyword. The constructor initializes the newly created object. ... A Java class constructor initializes instances (objects) of that class.

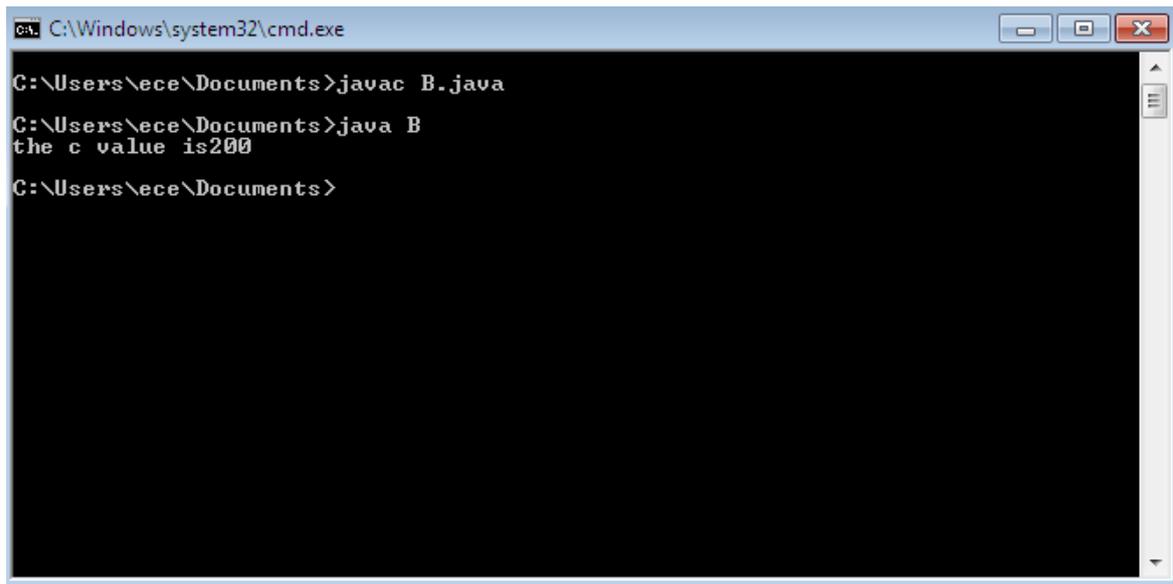
Rules for constructors

Name of the constructor must be same as that of a class name. If you give another name it will give compile time error. If you give another name, it is neither a method because of no return type, nor constructor because name is different from class name.

- Constructors must not have a return type.
- Every class should have at least one constructor.
- Constructor can be declared as private.
- One class can have more than one constructors. It is called Constructor Overloading.
- Duplicate Constructors not allowed.
- Multiple arguments of the constructors can't have same name.
- Only public, protected and private keywords are allowed before a constructor name.
- First statement in a constructor must be either super() or this().
- Recursive constructor calling is not allowed.

Program:

```
import java.io.*;
class A
{
A(int a,int b)
{
int c;
c=a*b;
System.out.println("the c value is"+c);
}
}
class B
{
public static void main(String args[]) throws IOException
{
A ob=new A(10,20);
}
}
```

Output:

```
ca. C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>javac B.java
C:\Users\ece\Documents>java B
the c value is200
C:\Users\ece\Documents>
```

Sample viva questions:

- 1) What is the use of a constructor?
- 2) What is a constructor?
- 3) When a constructor is called?
- 4) Do we have Copy Constructor in Java?
- 5) What happens if you keep a return type for a constructor?
- 6) What is default constructor?
- 7) Do we have destructors in Java?
- 8) What are the steps to create an object?
- 9) What is a reference type? Give an example.

11) Write a JAVA program to implement constructor overloading.**Aim: A Java Program on Constructor overloading.****Description**

Just like method overloading, constructors also can be overloaded. Same constructor declared with different parameters in the same class is known as constructor overloading. Compiler differentiates which constructor is to be called depending upon the number of parameters and their sequence of data types.

Rules for constructor overloading:

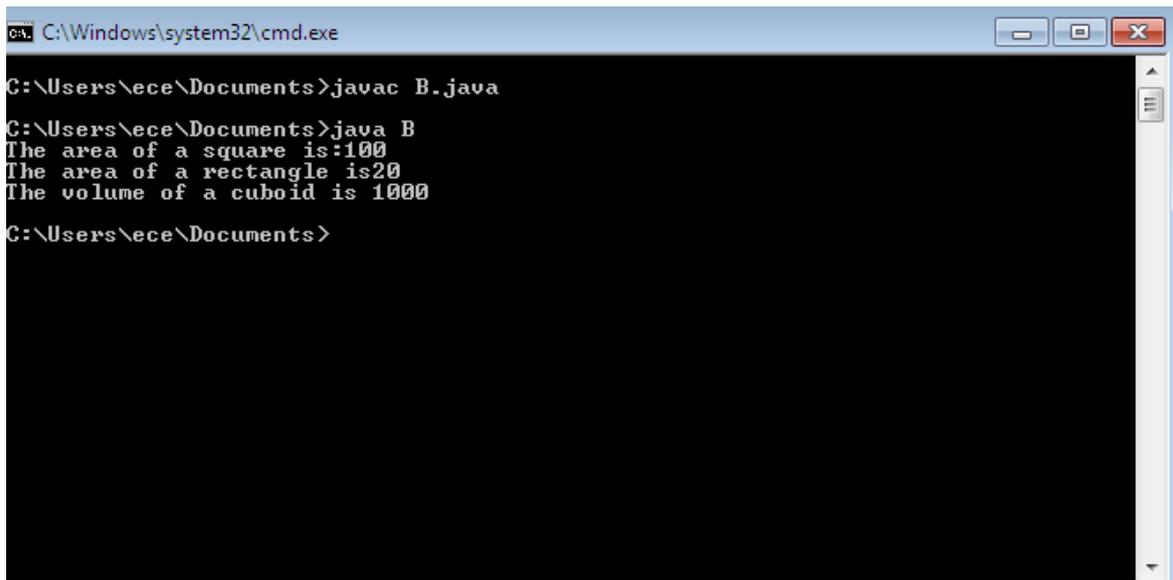
Name of the constructor must be same as that of a class name. If you give another name it will give compile time error. If you give another name, it is neither a method because of no return type, nor constructor because name is different from class name.

- Constructors must not have a return type.
- Every class should have at least one constructor.
- Constructor can be declared as private.
- One class can have more than one constructors. It is called Constructor Overloading.
- Duplicate Constructors not allowed.
- Multiple arguments of the constructors can't have same name.
- Only public, protected and private keywords are allowed before a constructor name.
- First statement in a constructor must be either super() or this().
- Recursive constructor calling is not allowed.
- No Cyclic calling of constructors.

Program:

```
import java.io.*;
class A
{
A(int a)
{
System.out.println("The area of a square is:"+a*a);
}
A(int a,int b)
{
super();
System.out.println("The area of a rectangle is "+a*b);
}
A(int a,int b,int c)
{
super();
System.out.println("The volume of a cuboid is "+a*b*c);
}
```

```
}  
class B  
{  
public static void main(String args[]) throws IOException  
{  
A ob=new A(10);  
A ob1=new A(10,2);  
A ob2=new A(10,2,50);  
}  
}
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>javac B.java
C:\Users\ece\Documents>java B
The area of a square is:100
The area of a rectangle is20
The volume of a cuboid is 1000
C:\Users\ece\Documents>
```

Sample viva questions:

1. What is the signature of a constructor?
2. When constructor overloading can be done?
3. Can you create an object without using new operator in Java?
4. Can we overload constructors?
5. How can we create objects if we make the constructor private ?
6. Can we use both "this" and "super" in a constructor ?
7. Is a constructor can be abstract?why?
8. Is a constructor can be static?

12) Write a JAVA program implement method overloading.**Aim: A Java Program on overloading of methods****Description:**

Method Overloading is a feature that allows a class to have more than one method having the same name, if their argument lists are different. It is similar to constructor overloading in Java, that allows a class to have more than one constructor having different argument lists.

Method Overloading in Java

If a class have multiple methods by same name but different parameters, it is known as Method Overloading. If we have to perform only one operation, having same name of the methods increases the readability of the program.

Different ways to overload the method

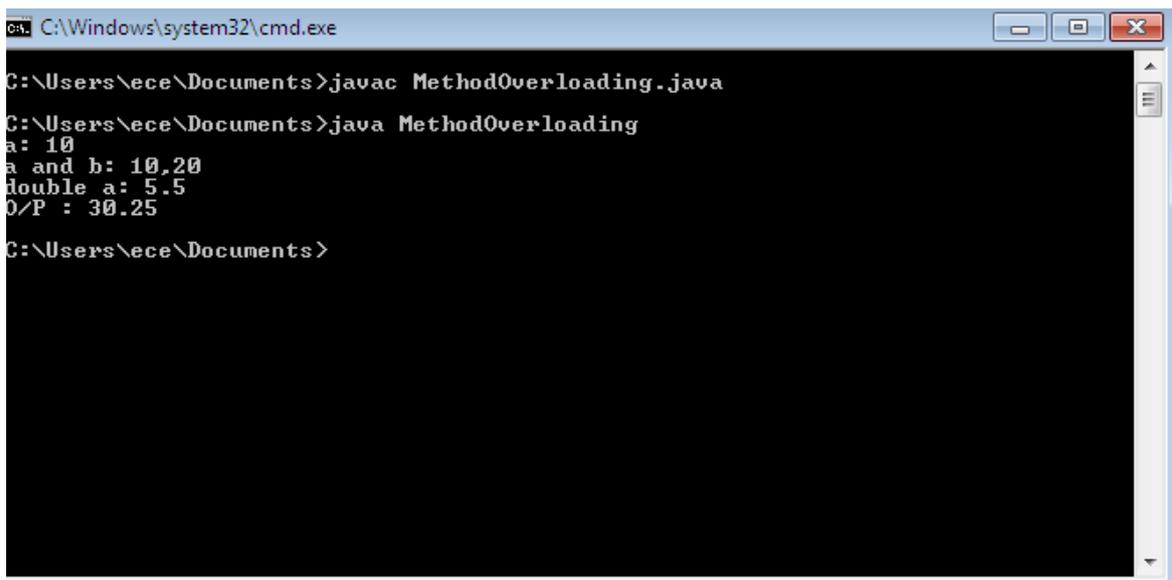
There are two ways to overload the method in java

1. By changing number of arguments
2. By changing the data type

Program:

```
class Overload
{
    void demo (int a)
    {
        System.out.println ("a: " + a);
    }
    void demo (int a, int b)
    {
        System.out.println ("a and b: " + a + "," + b);
    }
    double demo(double a) {
        System.out.println("double a: " + a);
    }
}
```

```
        return a*a;
    }
}
class MethodOverloading
{
    public static void main (String args [])
    {
        Overload Obj = new Overload();
        double result;
        Obj .demo(10);
        Obj .demo(10, 20);
        result = Obj .demo(5.5);
        System.out.println("O/P : " + result);
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>javac MethodOverloading.java
C:\Users\ece\Documents>java MethodOverloading
a: 10
a and b: 10.20
double a: 5.5
O/P : 30.25
C:\Users\ece\Documents>
```

Sample viva questions:

- 1) What is static binding or compile time binding?
- 2) What is method signature? What are the things it consist of?
- 3) Can we declare one overloaded method as static and another one as non-static?
- 4) Is it possible to have two methods in a class with same method signature but different return types?
- 5) Overloading is the best example of dynamic binding. True or false?
- 6) Can we overload main method?

13) Write a JAVA program to implement Single Inheritance**Aim: Illustrating simple inheritance****Description:**

Inheritance in Java

Inheritance in java is a mechanism in which one object acquires all the properties and behaviors of parent object.

The idea behind inheritance in java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of parent class, and you can add new methods and fields also.

Program:

```
import java.io.*;
public class Inherit_Single {

    protected String str;

    Inherit_Single() {

        str = "Java ";
    }
}

class SubClass extends Inherit_Single {

    SubClass() {

        str = str.concat("World !!!");
    }

    void display()
    {

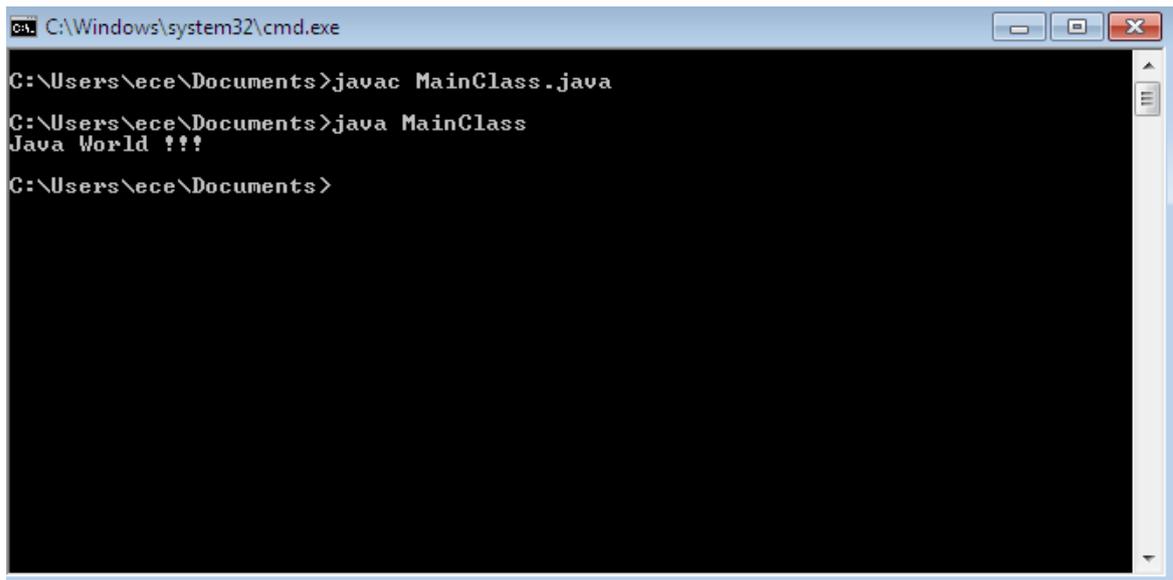
        System.out.println(str);
    }
}

class MainClass {

    public static void main (String args[]){

        SubClass obj = new SubClass();

        obj.display();
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe

C:\Users\ece\Documents>javac MainClass.java
C:\Users\ece\Documents>java MainClass
Java World !!!
C:\Users\ece\Documents>
```

Sample viva questions:

- 1) What is inheritance? Which principle of OOPs supports inheritance?
- 2) What is the advantage of inheritance?
- 3) What are the types of inheritance?
- 4) What is simple inheritance?
- 5) can a super class have more than one sub class?
- 6) what is a super class?
- 7) What is a sub class?
- 8) Which keyword is used to inherit a class from super class?

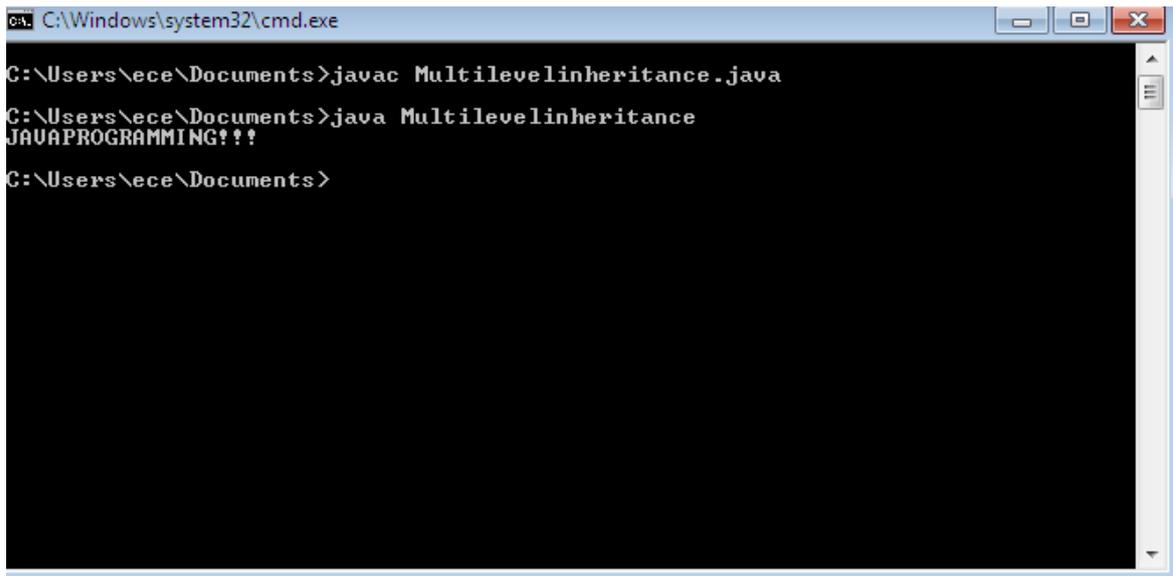
14) Write a JAVA program that illustrates Multi level inheritance**Aim: Illustrating multi level inheritance****Description:**

Multi level inheritance refers to the process of inheriting the properties of derived class into another class.

Program:

```
import java.io.*;
class Inherit

    protected String str;
    Inherit()
    {
        str="JA";
    }
}
class Subclass1 extends Inherit
{
    Subclass1()
    {
        str=str.concat("VA");
    }
}
class Subclass2 extends Subclass1
{
    Subclass2()
    {
        str=str.concat("PROGRAMMING");
    }
}
class Subclass3 extends Subclass2
{
    Subclass3()
    {
        str=str.concat("!!!");
    }
    void display()
    {
        System.out.println(str);
    }
}
class Multilevelinheritance
{
    public static void main(String args[]) throws IOException
    {
        Subclass3 ob=new Subclass3();
        ob.display();
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>javac Multilevelinheritance.java
C:\Users\ece\Documents>java Multilevelinheritance
JAVAPROGRAMMING!!!
C:\Users\ece\Documents>
```

Sample viva questions:

- 1) Differences between multiple inheritance and multilevel inheritance.
- 2) what is multilevel inheritance?
- 3) Does java support multiple inheritance. why?
- 4) What is the syntax of inheritance?
- 5) What is the differences between encapsulation and abstraction?
- 6) Is the statement with Super keyword must be as a first statement either in method or constructor ?Justify.
- 7) Can I use super.super to refer the method or variable in grand parent class?

15) Write a java program for abstract class to find areas of different shapes**Aim:A Java program for abstract class to find areas of different shapes.****Description:**

Abstraction is a process of hiding the implementation details and showing only functionality to the user.

Another way, it shows only important things to the user and hides the internal details for example sending sms, you just type the text and send the message. You don't know the internal processing about the message delivery.

Abstraction lets you focus on what the object does instead of how it does it.

Ways to achieve Abstraction

There are two ways to achieve abstraction in java

1. Abstract class (0 to 100%)
2. Interface (100%)

Abstract class

A class that is declared as abstract is known as abstract class. It needs to be extended and its method implemented. It cannot be instantiated.

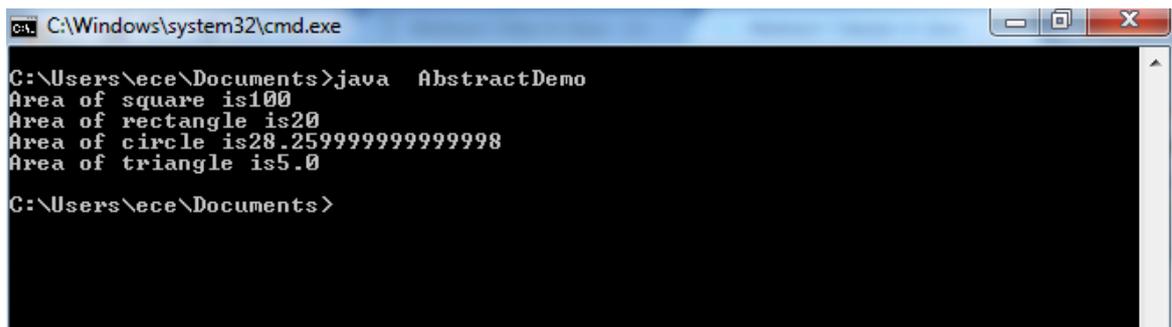
Example

```
abstract class A{ }
```

Program:

```
import java.io.*;
abstract class Area
{
    abstract void square(int s);
    abstract void rect(int l,int b);
    abstract void circle(int r);
    abstract void triangle(int b,int h);
}
class A extends Area
{
    void square(int s)
    {
        System.out.println("Area of square is"+s*s);
    }
    void rect(int l,int b)
    {
        System.out.println("Area of rectangle is"+l*b);
    }
    void circle(int r)
    {
        System.out.println("Area of circle is"+(3.14*r*r));
    }
    void triangle(int b,int h)
    {
        System.out.println("Area of triangle is"+(0.5*b*h));
    }
}
```

```
}  
class AbstractDemo  
{  
    public static void main(String args[]) throws IOException  
    {  
        A a1=new A();  
        a1.square(10);  
        a1.rect(10,2);  
        a1.circle(3);  
        a1.triangle(2,5);  
    }  
}
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>java AbstractDemo
Area of square is100
Area of rectangle is20
Area of circle is28.259999999999998
Area of triangle is5.0
C:\Users\ece\Documents>
```

Sample viva questions:

- 1) What is abstract class?
- 2) can we create object for the abstract class?
- 3) How a superior of a team controls his team members using abstract method?
- 4) what is the need of inheritance in abstraction?
- 5) Differences between inheritance and abstraction.

16) Write a JAVA program give example for “super” keyword.**Aim: Demonstrating usage of ‘super’ keyword****Description:**

The super is a reference variable that is used to refer immediate parent class object. Whenever you create the instance of subclass, an instance of parent class is created implicitly i.e. referred by super reference variable.

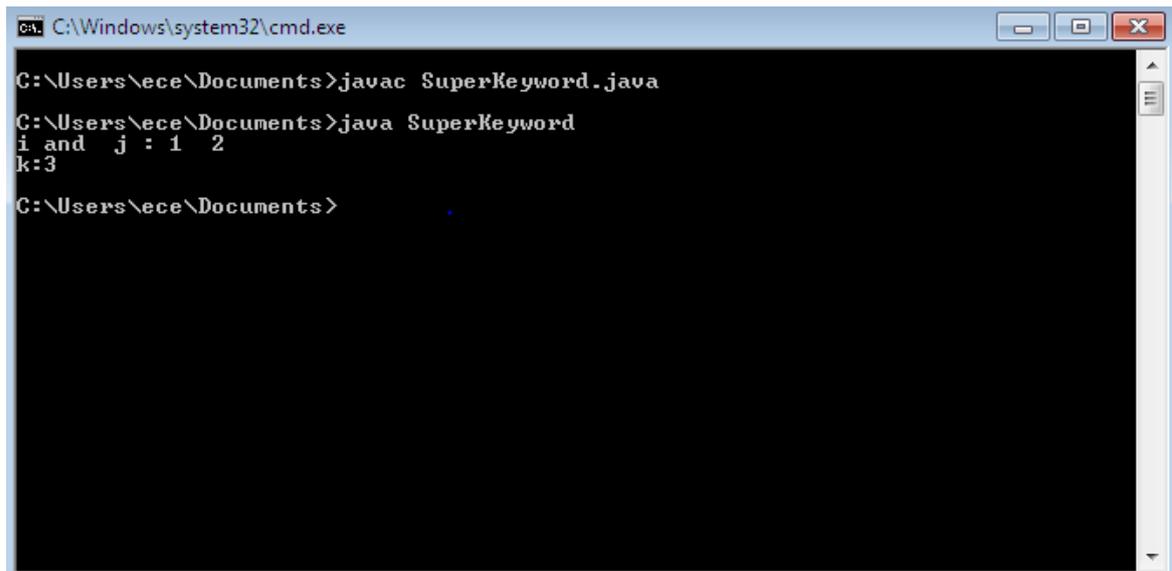
Usage of super Keyword

1. super is used to refer immediate parent class instance variable.
2. super() is used to invoke immediate parent class constructor.
3. super is used to invoke immediate parent class method.

Program:

```
class A
{
    int i, j;
    A()
    {
        i = 0;
        j = 0;
    }
    A(int a, int b)
    {
        i = a;
        j = b ;
    }
    void show()
    {
        System.out.println ("i and j : "+ i + " " + j);
    }
}
class B extends A
{
    int k;
    B()
    {
        super();
        k = 0;
    }
    B(int a, int b, int c)
    {
        super(a, b);
        k =c;
    }
    void show()
    {
        super.show();
        System.out.println("k:" + k);
    }
}
```

```
    }  
class SuperKeyword  
{  
    public static void main ( String args[])  
    {  
        B subob = new B( 1, 2, 3 );  
        subob.show();  
    }  
}
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>javac SuperKeyword.java
C:\Users\ece\Documents>java SuperKeyword
i and j : 1 2
k:3
C:\Users\ece\Documents>
```

Sample viva questions:

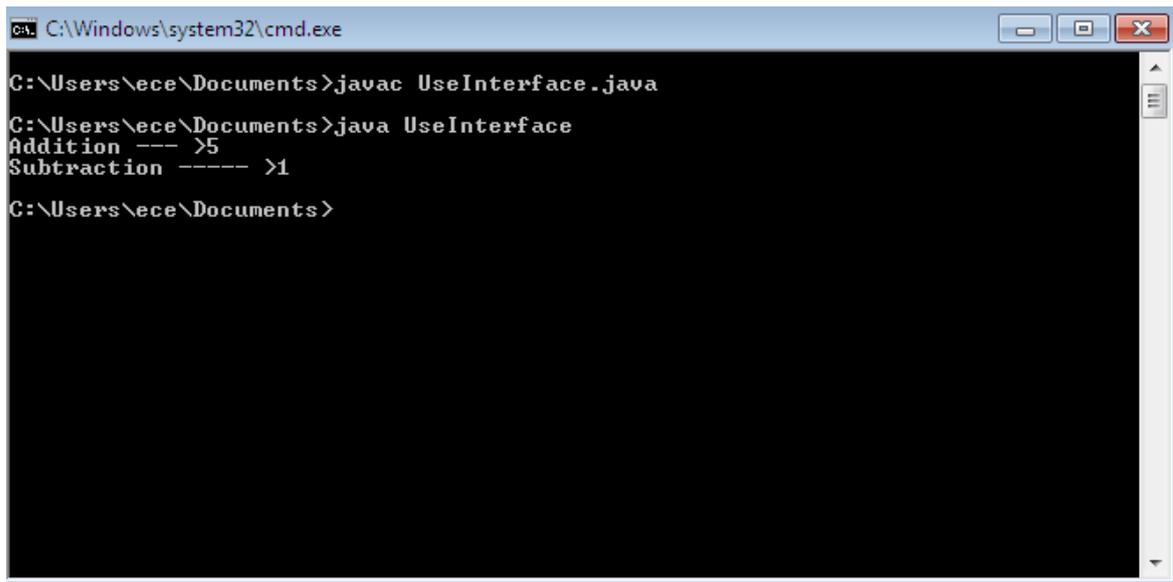
- 1) What is super keyword?
- 2) How to invoke super class's parameterized constructor from sub class ?
- 3) When super keyword is used?
- 4) How you can call the super class constructor?

17) Write a JAVA program to implement Interface. What kind of Inheritance can be achieved?**Aim: implementing interface and using this achieving multiple inheritance.****Description:**

An interface is a reference type in Java. It is similar to class. It is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface. Along with abstract methods, an interface may also contain constants, default methods, static methods, and nested types.

Program:

```
class Number
{
    protected int x;
    protected int y;
}
interface Arithmetic
{
    int add(int a, int b);
    int sub(int a, int b);
}
class UseInterface extends Number implements Arithmetic
{
    public int add(int a, int b)
    {
        return(a + b);
    }
    public int sub(int a, int b)
    {
        return (a - b);
    }
    public static void main(String args[])
    {
        UseInterface ui = new UseInterface();
        System.out.println("Addition --- >" + ui.add(2,3));
        System.out.println("Subtraction ----- >" + ui.sub(2,1));
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>javac UseInterface.java
C:\Users\ece\Documents>java UseInterface
Addition --- >5
Subtraction ----- >1
C:\Users\ece\Documents>
```

Sample viva questions:

- 1) what is interface?
- 2) What is the keyword used to inherit an interface?
- 3) What is the default access of variable and method in an interface
- 4) What is final in java language?
- 5) Using interface concept what type of inheritance is achieved
- 6) Does interface contain concrete methods?

18) Write a Java program to implement exception handling**Aim: Java program on Exception Handling.****Description:**

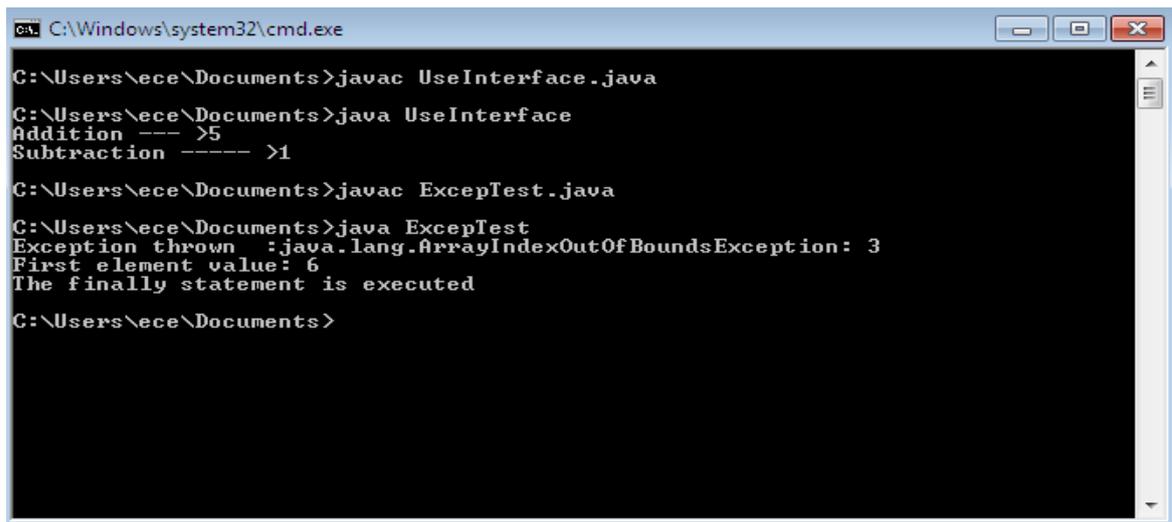
Exception is an abnormal condition. In java, exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.

Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IO, SQL, Remote etc. The core advantage of exception handling is to maintain the normal flow of the application. Exception normally disrupts the normal flow of the application that is why we use exception handling.

Program:

```
Import java.io.*;
public class ExcepTest{

    public static void main(String args[]){
        int a[] = new int[2];
        try{
            System.out.println("Access element three : " + a[3]);
        }catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Exception thrown : " + e);
        }
        finally{
            a[0] = 6;
            System.out.println("First element value: " +a[0]);
            System.out.println("The finally statement is executed");
        }
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe

C:\Users\ece\Documents>javac UseInterface.java
C:\Users\ece\Documents>java UseInterface
Addition --- >5
Subtraction ----- >1

C:\Users\ece\Documents>javac ExcepTest.java
C:\Users\ece\Documents>java ExcepTest
Exception thrown :java.lang.ArrayIndexOutOfBoundsException: 3
First element value: 6
The finally statement is executed

C:\Users\ece\Documents>
```

19) Write a JAVA program Illustrating Multiple catch clauses**Aim: Illustrating multiple catch clauses****Description:**

To perform different tasks at the occurrence of different Exceptions, use java multi catch block. A single statement may throw more than one type of exception. In such cases, Java allows you to put more than one catch blocks. One catch block handles one type of exception. When an exception is thrown by the try block, all the catch blocks are examined in the order they appear and one catch block which matches with exception thrown will be executed. After, executing catch block, program control comes out of try-catch unit.

Program:

```
public class ExceptionExample {  
  
    public static void main(String argv[]) {  
  
        int num1 = 10;  
        int num2 = 0;  
        int result = 0;  
        int arr[] = new int[5];  
  
        try {  
            arr[0] = 0;  
            arr[1] = 1;  
            arr[2] = 2;  
            arr[3] = 3;  
            arr[4] = 4;  
            //arr[5] = 5;  
  
            result = num1 / num2;  
            System.out.println("Result of Division : " + result);  
  
        } catch (ArithmeticException e) {  
            System.out.println("Err: Divided by Zero");  
  
        } catch (ArrayIndexOutOfBoundsException e) {  
            System.out.println("Err: Array Out of Bound");  
        }  
    }  
}
```

Output:

```
C:\Users\ece\Documents>javac ExceptionExample.java
C:\Users\ece\Documents>java ExceptionExample
Err: Array Out of Bound
C:\Users\ece\Documents>javac ExceptionExample.java
C:\Users\ece\Documents>java ExceptionExample
Err: Divided by Zero
C:\Users\ece\Documents>
```

Sample viva questions:

- 1) What is an exception?
- 2) What is the purpose of the throw and throws keywords?
- 3) How can you handle an exception?
- 4) How can you catch multiple exceptions?
- 5) What is the difference between a checked and an unchecked exception?
- 6) What is the difference between an exception and error?
- 7) What exception will be thrown executing the following code block?
- 8) What is exception chaining?
- 9) What is a stacktrace and how does it relate to an exception?
- 10) Why would you want to subclass an exception?
- 11) What are some advantages of exceptions?
- 12) Differentiate throw and throws
- 13) What is finally Give an example

20) Write a JAVA program that implements Runtime polymorphism.**Aim: Demonstrating method overriding.****Description:**

Method Overriding in Java

If subclass (child class) has the same method as declared in the parent class, it is known as method overriding.

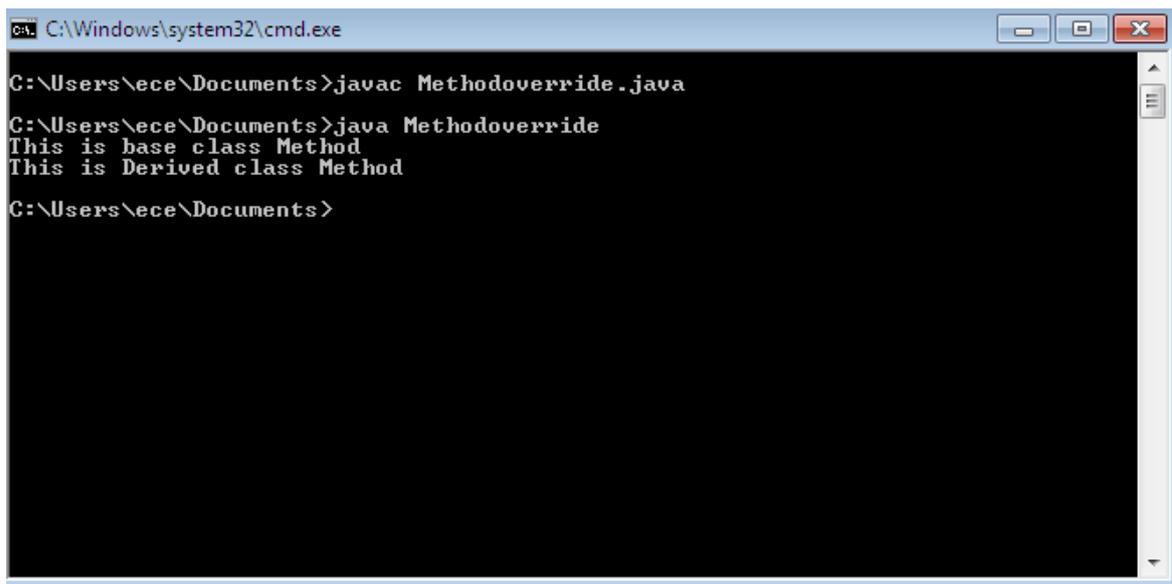
In other words, If subclass provides the specific implementation of the method that has been provided by one of its parent class, it is known as Method Overriding.

Rules for Method Overriding

1. method must have same name as in the parent class
2. method must have same parameter as in the parent class.
3. must be IS-A relationship (inheritance).

Program:

```
import java.io.*;
class Override
{
    public void methodoverride()
    {
        System.out.println("This is base class Method");
    }
}
class Overridederived extends Override
{
    public void methodoverride()
    {
        System.out.println("This is Derived class Method");
    }
}
public class Methodoverride
{
    public static void main(String args[]) throws IOException
    {
        Override ob=new Override();
        Overridederived ob1=new Overridederived();
        ob.methodoverride();
        ob1.methodoverride();
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>javac Methodoverride.java
C:\Users\ece\Documents>java Methodoverride
This is base class Method
This is Derived class Method
C:\Users\ece\Documents>
```

Sample viva questions:

- 1) what is Runtime polymorphism?
- 2) What is compile time polymorphism? Give an Example.
- 3) Differences between overloading and overriding in Java.
- 4) How polymorphism supported in Java?
- 5) What is function overloading and function overriding in Java?

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

Aim: case study on RunTime Polymorphism

Explanation

Dynamic Method dispatch or Runtime Polymorphism. Polymorphism is the capability of an action or method to do different things based on the object that it is acting upon. In other words, polymorphism allows you define one interface and have multiple implementation. This is one of the basic principles of object oriented programming.

The method overriding is an example of runtime polymorphism. You can have a method in subclass overrides the method in its super classes with the same name and signature. Java virtual machine determines the proper method to call at the runtime, not at the compile time.

Method overriding is one of the ways in which Java supports Runtime Polymorphism. Dynamic method dispatch is the mechanism by which a call to an overridden method is resolved at run time, rather than compile time.

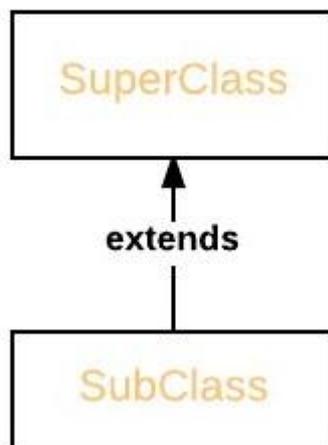
When an overridden method is called through a superclass reference, Java determines which version(superclass/subclasses) of that method is to be executed based upon the type of the object being referred to at the time the call occurs. Thus, this determination is made at run time.

At run-time, it depends on the type of the object being referred to (not the type of the reference variable) that determines which version of an overridden method will be executed

A superclass reference variable can refer to a subclass object. This is also known as upcasting. Java uses this fact to resolve calls to overridden methods at run time.

Upcasting

SuperClass obj = new SubClass



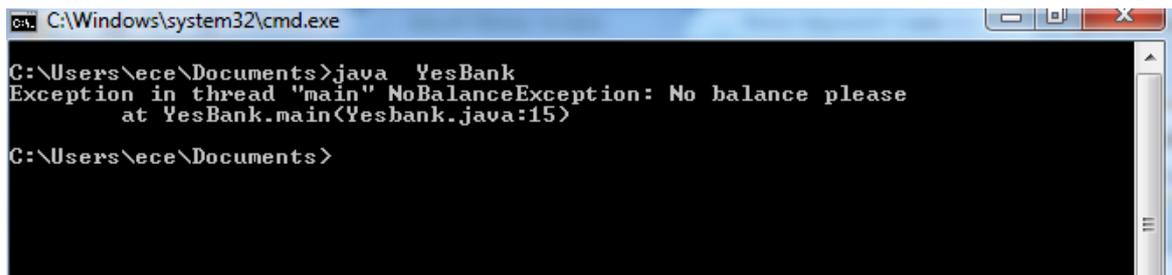
22) Write a JAVA program for creation of Illustrating throw.**Aim: Describing throw keyword for Exception mechanism****Description:**

Exception is an abnormal condition. In java, exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.

Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IO, SQL, Remote etc. The core advantage of exception handling is to maintain the normal flow of the application. Exception normally disrupts the normal flow of the application that is why we use exception handling.

Program:

```
class NoBalanceException extends Exception
{
    public NoBalanceException(String problem)
    {
        super(problem);
    }
}
class YesBank
{
    public static void main( String args[ ] ) throws NoBalanceException
    {
        int balance = 100, withdraw = 1000;
        if (balance < withdraw)
        {
            NoBalanceException e = new NoBalanceException("No balance please");
            throw e;
        }
        else
        {
            System.out.println("Draw & enjoy, Best wishes of the day");
        }
    }
}
```

Output:

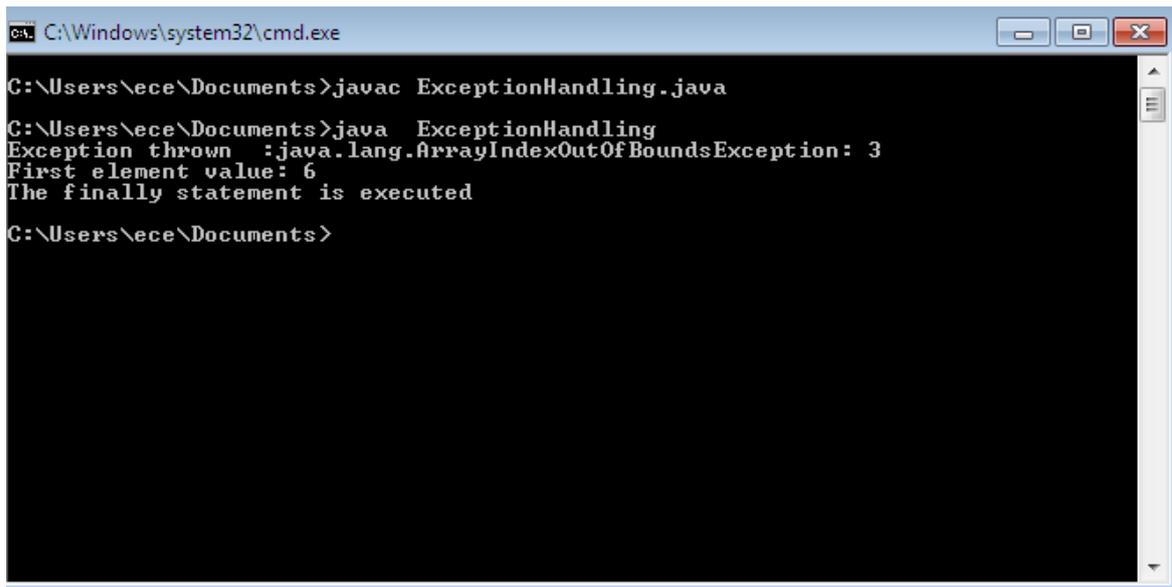
```
C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>java YesBank
Exception in thread "main" NoBalanceException: No balance please
    at YesBank.main(Yesbank.java:15)
C:\Users\ece\Documents>
```

Sample viva questions:

- 1)what is throw keyword?
- 2)why throw keyword is usefull in Exception Handling?
- 3)what are the types of Exceptions?
- 4)What is an user-defined Exception?
- 5)What happens if Exception was not handled?

23) Write a JAVA program for creation of Illustrating finally.**Aim: Using finally block****Program:**

```
public class ExceptionHandling
{
    public static void main(String args[])
    {
        int a[] = new int[2];
        try
        {
            System.out.println("Access element three :" + a[3]);
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Exception thrown :" + e);
        }
        finally
        {
            a[0] = 6;
            System.out.println("First element value: " +a[0]);
            System.out.println("The finally statement is executed");
        }
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe

C:\Users\ece\Documents>javac ExceptionHandling.java

C:\Users\ece\Documents>java ExceptionHandling
Exception thrown :java.lang.ArrayIndexOutOfBoundsException: 3
First element value: 6
The finally statement is executed

C:\Users\ece\Documents>
```

Sample Viva Questions:

- 1) What is finally keyword?
- 2) What is difference in final, finalize and finally keyword in Java?
- 3) In Java what finally block do?
- 4) What is super Exception

24) Write a JAVA program for creation of Java Built-in Exceptions**Aim: To illustrate built-in exceptions in exception handling****Program:**

```
class ArithmeticException_Demo {
public static void main(String args[])
{
    try {
        int a = 30, b = 0;
        int c = a / b; // cannot divide by zero
        System.out.println("Result = " + c);
    }
    catch (ArithmeticException e) {
        System.out.println("Can't divide a number by 0");
    }
}
}
```

Output:

```
C:\Users\ece\Documents>javac ArithmeticException_Demo.java
C:\Users\ece\Documents>java ArithmeticException_Demo
Can't divide a number by 0
C:\Users\ece\Documents>
```

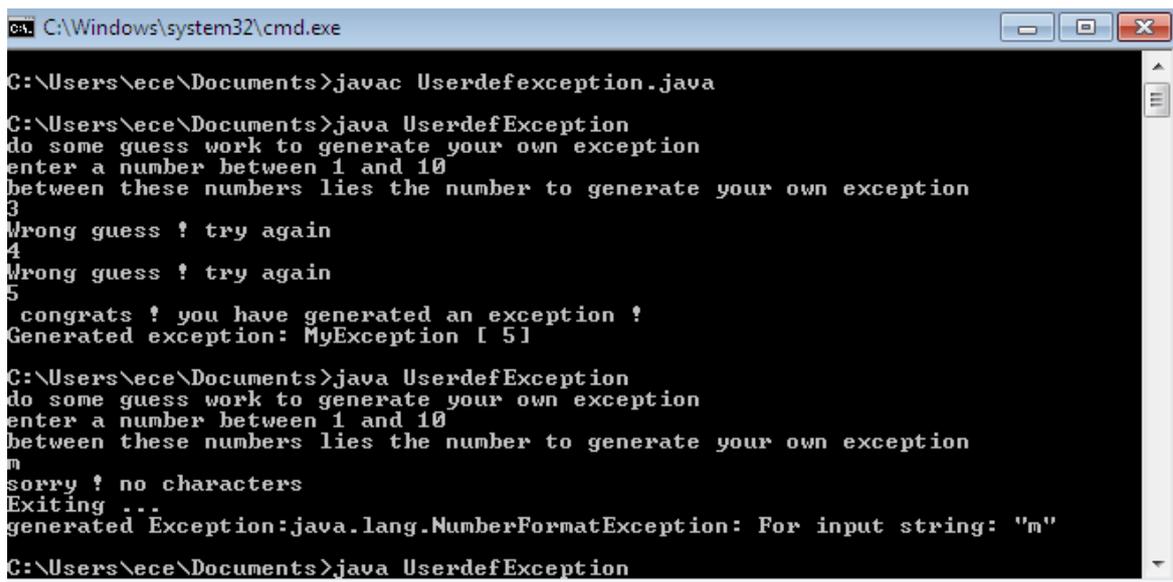
Sample viva questions:

- 1)what is Built-in Exception?
- 2)Give some examples for built-in Exceptions?
- 3)Differences between compile time error and runtime error.

25) Write a JAVA program for creation of User defined exception.**Aim: Creating user defined exception.****Program:**

```
import java.io.*;
class MyException extends Exception
{
    private int a;
    MyException(int b)
    {
        a = b;
    }
    public String toString()
    {
        return "MyException [ " + a + " ]";
    }
}
class Userdefexception
{
    public int x;
    final int k = 5;
    void getInt()
    {
        try {
            BufferedReader d = new BufferedReader(new InputStreamReader(System.in));
            System.out.println("do some guess work to generate your own exception ");
            System.out.println("enter a number between 1 and 10");
            System.out.println("between these numbers lies the number to generate your own
exception ");
            String line;
            while((line = d.readLine()) != null)
            {
                x = Integer.parseInt(line);
                if ( x == 5)
                {
                    System.out.println(" congrats ! you have generated an exception !");
                    throw new MyException(x);
                }
            }
            else
            System.out.println("Wrong guess ! try again");
            continue;
        }
        catch(MyException m) {
            System.out.println("Generated exception: " +m);
        }
        catch(NumberFormatException n)
        {
            System.out.println("sorry ! no characters");
        }
    }
}
System.out.println("Exiting ...");
```

```
System.out.println("generated Exception:"+n);
}
catch(IOException e) { }
}
public static void main(String a[]) {
UserdefException u = new UserdefException();
u.getInt();
}
}
```

Output:

```
C:\Windows\system32\cmd.exe

C:\Users\ece\Documents>javac Userdefexception.java

C:\Users\ece\Documents>java UserdefException
do some guess work to generate your own exception
enter a number between 1 and 10
between these numbers lies the number to generate your own exception
3
Wrong guess ! try again
4
Wrong guess ! try again
5
congrats ! you have generated an exception !
Generated exception: MyException [ 5]

C:\Users\ece\Documents>java UserdefException
do some guess work to generate your own exception
enter a number between 1 and 10
between these numbers lies the number to generate your own exception
m
sorry ! no characters
Exiting ...
generated Exception:java.lang.NumberFormatException: For input string: "m"

C:\Users\ece\Documents>java UserdefException
```

Sample viva questions:

- 1)what is user defined exception?
- 2)How to handle user defined Exception?
- 3)How to create user defined Exception in Java?
- 4:What do you mean by checked Exception?

26) 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)

Program:

```
class GoodMorning extends Thread
{
public void run()
{
try
{
for(int i=1;i<=10;i++)
{
sleep(1000);
System.out.println("good morning");
}
}
catch(Exception e)
{
System.out.println(e);
}
}
class Hello extends Thread
{
public void run()
{
try
{
for(int j=1;j<=10;j++)
{
sleep(2000);
System.out.println("hello");
}
}
catch(Exception e)
{
System.out.println(e);
}
}
class Welcome extends Thread
{
public void run()
```

```
{
try

{
for(int k=1;k<=10;k++)
{
sleep(3000);
System.out.println("welcome");
}
}
catch(Exception e)
{
System.out.println(e);
}
}
}
class ThreadDemo1
{
public static void main(String args[])
{
GoodMorning a1=new GoodMorning();
Hello b1=new Hello();
Welcome c1=new Welcome();
a1.start();
b1.start();
c1.start();
}
}

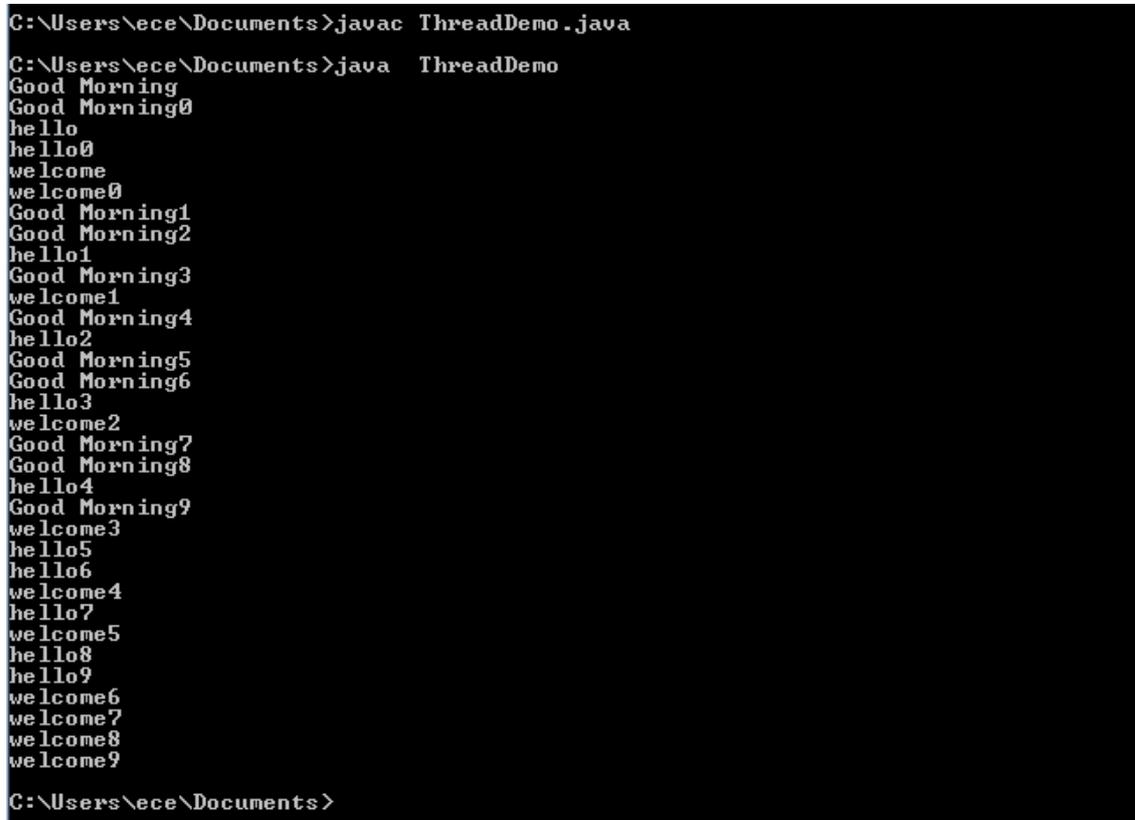

---


class GoodMorning implements Runnable
{
public void run()
{
try
{
for(int i=1;i<=10;i++)
{
Thread.sleep(1000);
System.out.println("good morning");
}
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

```
class Hello implements Runnable
{
public void run()
{
try
{
for(int j=1;j<=10;j++)
{
Thread.sleep(2000);
System.out.println("hello");
}
}
catch(Exception e)
{
System.out.println(e);
}
}
}
class Welcome implements Runnable
{
public void run()

{
try
{
for(int k=1;k<=10;k++)
{
Thread.sleep(3000);
System.out.println("welcome");
}
}
catch(Exception e)
{
System.out.println(e);
}
}
}
class ThreadDemo
{
public static void main(String args[])
{
GoodMorning a1=new GoodMorning();
Hello b1=new Hello();
Welcome c1=new Welcome();
Thread t1=new Thread(a1);
Thread t2=new Thread(b1);
```

```
Thread t3=new Thread(c1);
t1.start();
t2.start();
t3.start();
}
}
```

Output:

```
C:\Users\ece\Documents>javac ThreadDemo.java
C:\Users\ece\Documents>java ThreadDemo
Good Morning
Good Morning0
hello
hello0
welcome
welcome0
Good Morning1
Good Morning2
hello1
Good Morning3
welcome1
Good Morning4
hello2
Good Morning5
Good Morning6
hello3
welcome2
Good Morning7
Good Morning8
hello4
Good Morning9
welcome3
hello5
hello6
welcome4
hello7
welcome5
hello8
hello9
welcome6
welcome7
welcome8
welcome9
C:\Users\ece\Documents>
```

Sample viva questions:

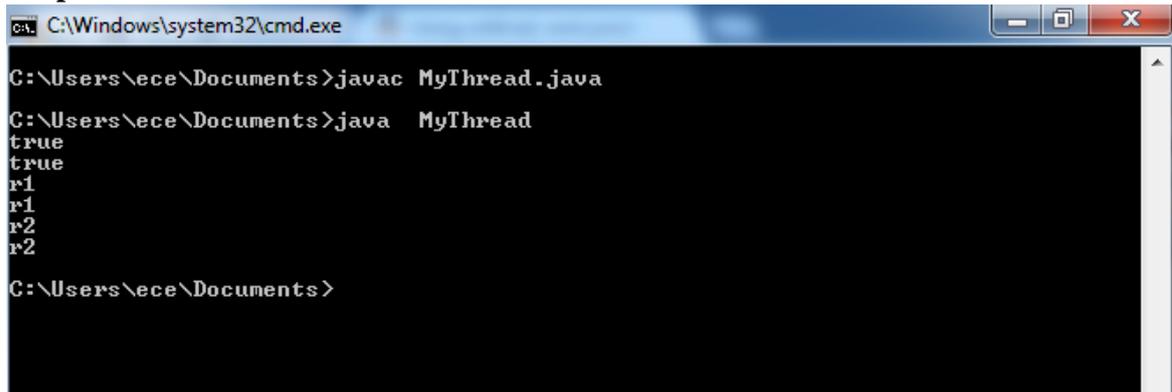
- 1)what is a thread?
- 2)what are the stages in lifecycle of a thread?
- 3)how many ways to create a thread. what are they?
- 4)What is the difference between inheritance and interface?
- 5)What do you mean by multi-Threading?

27) Write a program illustrating isAlive and join ().**Aim: Creating a thread class by extending Thread class and use isAlive and join methods****Program:**

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

try {
        t1.join(); // Waiting for c1 to finish
    }
    catch (InterruptedException ie)
    {
    }

        t2.start();
        System.out.println(t1.isAlive());
        System.out.println(t2.isAlive());
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>javac MyThread.java
C:\Users\ece\Documents>java MyThread
true
r1
r1
r2
r2
C:\Users\ece\Documents>
```

Sample viva questions:

- 1) what is suspending a thread?
- 2) How a thread can come out of suspended state?
- 3) How many types of threads are present in java?
- 4) What are the different Thread models?
- 5) what do you mean by Inter-Thread communication?

28) Write a Program illustrating Daemon Threads.**Aim: Illustrating Daemon Threads.****Description:**

A daemon thread is a thread that does not prevent the JVM from exiting when the program finishes but the thread is still running. An example for a daemon thread is the garbage collection. You can use the `setDaemon(boolean)` method to change the Thread daemon properties before the thread starts.

Program:

```
public class DaemonThreadExample1 extends Thread{

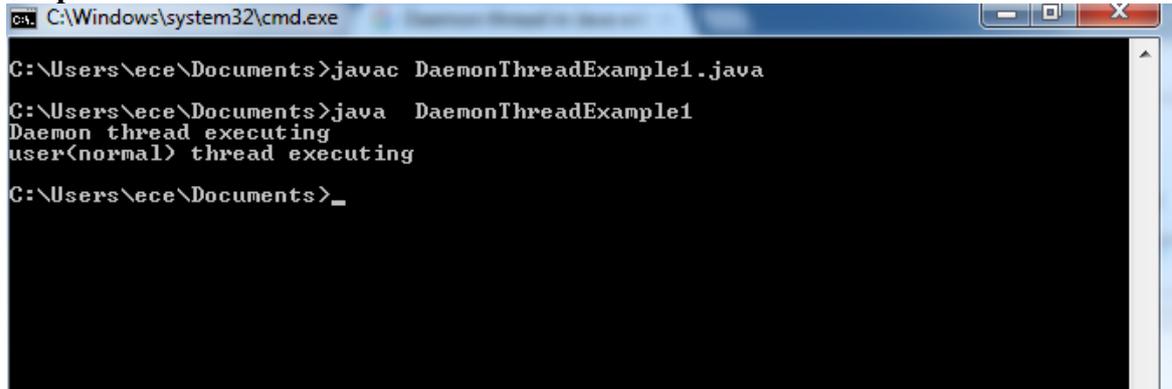
    public void run(){

        // Checking whether the thread is Daemon or not
        if(Thread.currentThread().isDaemon()){
            System.out.println("Daemon thread executing");
        }
        else{
            System.out.println("user(normal) thread executing");
        }
    }

    public static void main(String[] args){
        /* Creating two threads: by default they are
        * user threads (non-daemon threads)
        */
        DaemonThreadExample1 t1=new DaemonThreadExample1();
        DaemonThreadExample1 t2=new DaemonThreadExample1();

        //Making user thread t1 to Daemon
        t1.setDaemon(true);

        //starting both the threads
        t1.start();
        t2.start();
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\ece\Documents>javac DaemonThreadExample1.java
C:\Users\ece\Documents>java DaemonThreadExample1
Daemon thread executing
user<normal> thread executing
C:\Users\ece\Documents>_
```

Sample viva questions:

- 1)What is a Deamon Thread?
- 2)What is the differences between process and thread in java?
- 3)what does wait() mean?
- 4)What is the difference between notify() and notifyAll() in java?
- 5)What is the need of synchronised keyword?

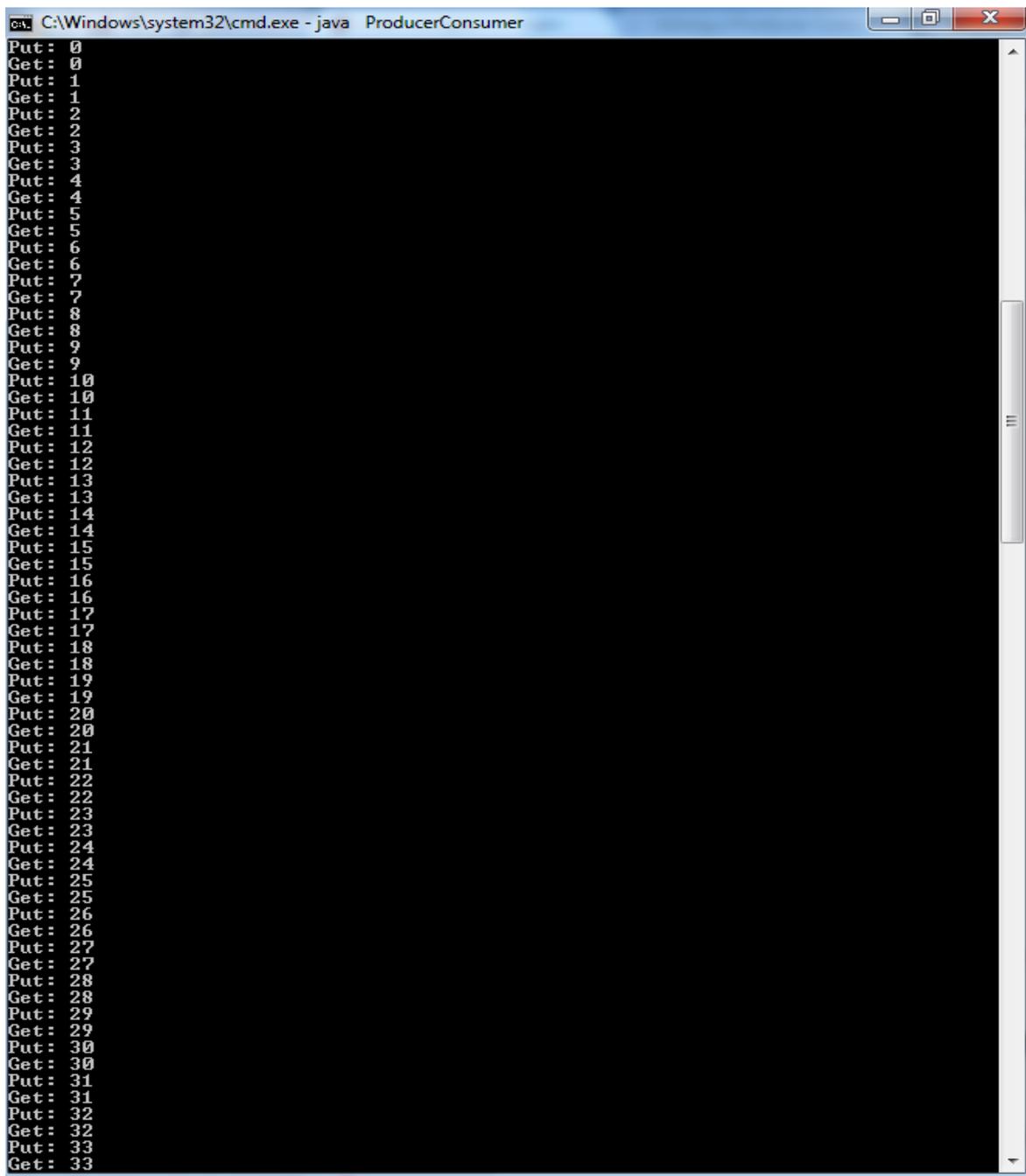
29) Write a JAVA program Producer Consumer Problem**Aim: Java program on producer consumer problem.****Description:**

The producer–consumer problem (also known as the bounded-buffer problem) is a classic example of a multi-process synchronization problem. The problem describes two processes, the producer and the consumer, who share a common, fixed-size buffer used as a queue.

Program:

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

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

Output:

```
C:\Windows\system32\cmd.exe - java ProducerConsumer
Put : 0
Get : 0
Put : 1
Get : 1
Put : 2
Get : 2
Put : 3
Get : 3
Put : 4
Get : 4
Put : 5
Get : 5
Put : 6
Get : 6
Put : 7
Get : 7
Put : 8
Get : 8
Put : 9
Get : 9
Put : 10
Get : 10
Put : 11
Get : 11
Put : 12
Get : 12
Put : 13
Get : 13
Put : 14
Get : 14
Put : 15
Get : 15
Put : 16
Get : 16
Put : 17
Get : 17
Put : 18
Get : 18
Put : 19
Get : 19
Put : 20
Get : 20
Put : 21
Get : 21
Put : 22
Get : 22
Put : 23
Get : 23
Put : 24
Get : 24
Put : 25
Get : 25
Put : 26
Get : 26
Put : 27
Get : 27
Put : 28
Get : 28
Put : 29
Get : 29
Put : 30
Get : 30
Put : 31
Get : 31
Put : 32
Get : 32
Put : 33
Get : 33
```

Sample viva questions:

- 1)what is producer consumer problem?
- 2)What do you mean by synchronised variable?
- 3)What is a synchronised block?
- 4)what is a synchronised method?

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

Aim: case study on thread Synchronisation after giving solution to the producer consumer problem.

In computing, the producer–consumer problem (also known as the bounded-buffer problem) is a classic example of a multi-process synchronization problem. The problem describes two processes, the producer and the consumer, which share a common, fixed-size buffer used as a queue.

The producer’s job is to generate data, put it into the buffer, and start again.

At the same time, the consumer is consuming the data (i.e. removing it from the buffer), one piece at a time.

Problem

To make sure that the producer won’t try to add data into the buffer if it’s full and that the consumer won’t try to remove data from an empty buffer.

Solution

The producer is to either go to sleep or discard data if the buffer is full. The next time the consumer removes an item from the buffer, it notifies the producer, who starts to fill the buffer again. In the same way, the consumer can go to sleep if it finds the buffer to be empty. The next time the producer puts data into the buffer, it wakes up the sleeping consumer. An inadequate solution could result in a deadlock where both processes are waiting to be awakened.

sleep() at the end of both methods just make the output of program run in step wise manner and not display everything all at once so that you can see what actually is happening in the program.

31) Write a JAVA program illustrate class path.**Aim: Illustrating class path.****Description:**

In setting up JDK and Java applications, you will encounter these environment variables: PATH , CLASSPATH , JAVA_HOME and JRE_HOME The OS searches the PATH entries for executable programs, such as Java Compiler (javac) and Java Runtime (java).

program:

package pack:

public class A

```
{
    public int c;
    public int add(int a,int b)
    {
        c=a+b;
        return c;
    }
    public int subtract(int a,int b)
    {
        c=a-b;
        return c;
    }
}
```

Sample viva questions:

- 1)What is JDK?
- 2)What is JVM?
- 3)What is JRE?
- 4)How to set classpath using command prompt?

32) Write a case study on including in class path in your os environment of your package.**Aim: Illustrating the setup of classpath in operating system.****Program:**

CLASSPATH is actually an environment variable in Java, and tells Java applications and the Java Virtual Machine (JVM) where to find the libraries of classes. These include any that you have developed on your own.

An environment variable is a global system variable, accessible by the computer's operating system (e.g., Windows). Other variables include COMPUTERTNAME, USERNAME (computer's name and user name).

In Java, CLASSPATH holds the list of Java class file directories, and the JAR file, which is Java's delivered class library file.

If you are trying to run a stand-alone Java program, you may find it necessary to change the CLASSPATH variable. When the program runs, Java's run-time system, called the interpreter, is working through your code. If it comes across a class name, it will look at each directory listed in the CLASSPATH variable. If it does not find the class name, it will error out.

We can set the value of CLASSPATH in DOS. The following example changes the variable to a local folder that we've created called CustomClasses; it's located in a folder on the C: drive called Java:

```
C:\>set CLASSPATH = C:\Java\CustomClasses
```

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

Aim: use and import package

Program:

```
import pack.*;
```

```
class B
```

```
{
```

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

```
    {
```

```
        A a1=new A();
```

```
        int l=a1.add(2,3);
```

```
        System.out.println("The addition of two no's is : "+l);
```

```
        l=a1.subtact(10,4);
```

```
        System.out.println("The subtraction of two no's is : "+l);
```

```
    }
```

```
}
```

Output:

The addition of two no's is : 5

The subtraction of two no's is : 6

Sample viva questions:

- 1)What is a package?
- 2)what is Abstraction of Data?
- 3)how to use a package?
- 4)can we create a package and use it in our programs?

34) Write a JAVA program to paint like paint brush in applet.**Aim: Creating an applet to paint like brush****Description:**

A very small application, especially a utility program performing one or a few simple functions.

Program:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<html>
<body>
<applet code="MouseDrag.class" width="300" height="300">
</applet>
</body>
</html> */
public class MouseDrag extends Applet implements MouseMotionListener{

public void init(){
addMouseMotionListener(this);
setBackground(Color.red);
}

public void mouseDragged(MouseEvent me){
Graphics g=getGraphics();
g.setColor(Color.white);
g.fillOval(me.getX(),me.getY(),5,5);
}
public void mouseMoved(MouseEvent me){ }

}
```

Sample viva questions:

- 1)What is an applet?
- 2)what are the stages in lifecycle of an applet?
- 3)How will you initialise an applet?
- 4)What is the difference between paint() and repaint()?

35) Write a JAVA program to display analog clock using Applet.**Aim: A JAVA program to display analog clock using Applet.****Program:**

```
import java.applet.*;
import java.awt.*;
import java.util.*;
import java.text.*;
/*<html>
<body>
<applet code="MyClock.class" width="300" height="300">
</applet>
</body>
</html> */
public class MyClock extends Applet implements Runnable {

    int width, height;
    Thread t = null;
    boolean threadSuspended;
    int hours=0, minutes=0, seconds=0;
    String timeString = "";

    public void init() {
        width = getSize().width;
        height = getSize().height;
        setBackground( Color.black );
    }

    public void start() {
        if ( t == null ) {
            t = new Thread( this );
            t.setPriority( Thread.MIN_PRIORITY );
            threadSuspended = false;
            t.start();
        }
        else {
            if ( threadSuspended ) {
                threadSuspended = false;
                synchronized( this ) {
                    notify();
                }
            }
        }
    }
}
```

```

public void stop() {
    threadSuspended = true;
}

public void run() {
    try {
        while (true) {

            Calendar cal = Calendar.getInstance();
            hours = cal.get( Calendar.HOUR_OF_DAY );
            if ( hours > 12 ) hours -= 12;
            minutes = cal.get( Calendar.MINUTE );
            seconds = cal.get( Calendar.SECOND );

            SimpleDateFormat formatter
                = new SimpleDateFormat( "hh:mm:ss", Locale.getDefault() );
            Date date = cal.getTime();
            timeString = formatter.format( date );

            // Now the thread checks to see if it should suspend itself
            if ( threadSuspended ) {
                synchronized( this ) {
                    while ( threadSuspended ) {
                        wait();
                    }
                }
            }
            repaint();
            t.sleep( 1000 ); // interval specified in milliseconds
        }
    }
    catch (Exception e) { }
}

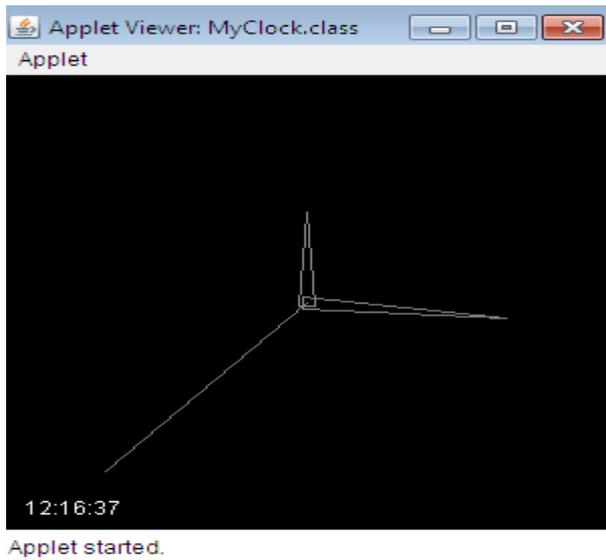
void drawHand( double angle, int radius, Graphics g ) {
    angle -= 0.5 * Math.PI;
    int x = (int)( radius*Math.cos(angle) );
    int y = (int)( radius*Math.sin(angle) );
    g.drawLine( width/2, height/2, width/2 + x, height/2 + y );
}

void drawWedge( double angle, int radius, Graphics g ) {
    angle -= 0.5 * Math.PI;
    int x = (int)( radius*Math.cos(angle) );
    int y = (int)( radius*Math.sin(angle) );
    angle += 2*Math.PI/3;
}

```

```
int x2 = (int)( 5*Math.cos(angle) );
int y2 = (int)( 5*Math.sin(angle) );
angle += 2*Math.PI/3;
int x3 = (int)( 5*Math.cos(angle) );
int y3 = (int)( 5*Math.sin(angle) );
g.drawLine( width/2+x2, height/2+y2, width/2 + x, height/2 + y );
g.drawLine( width/2+x3, height/2+y3, width/2 + x, height/2 + y );
g.drawLine( width/2+x2, height/2+y2, width/2 + x3, height/2 + y3 );
}

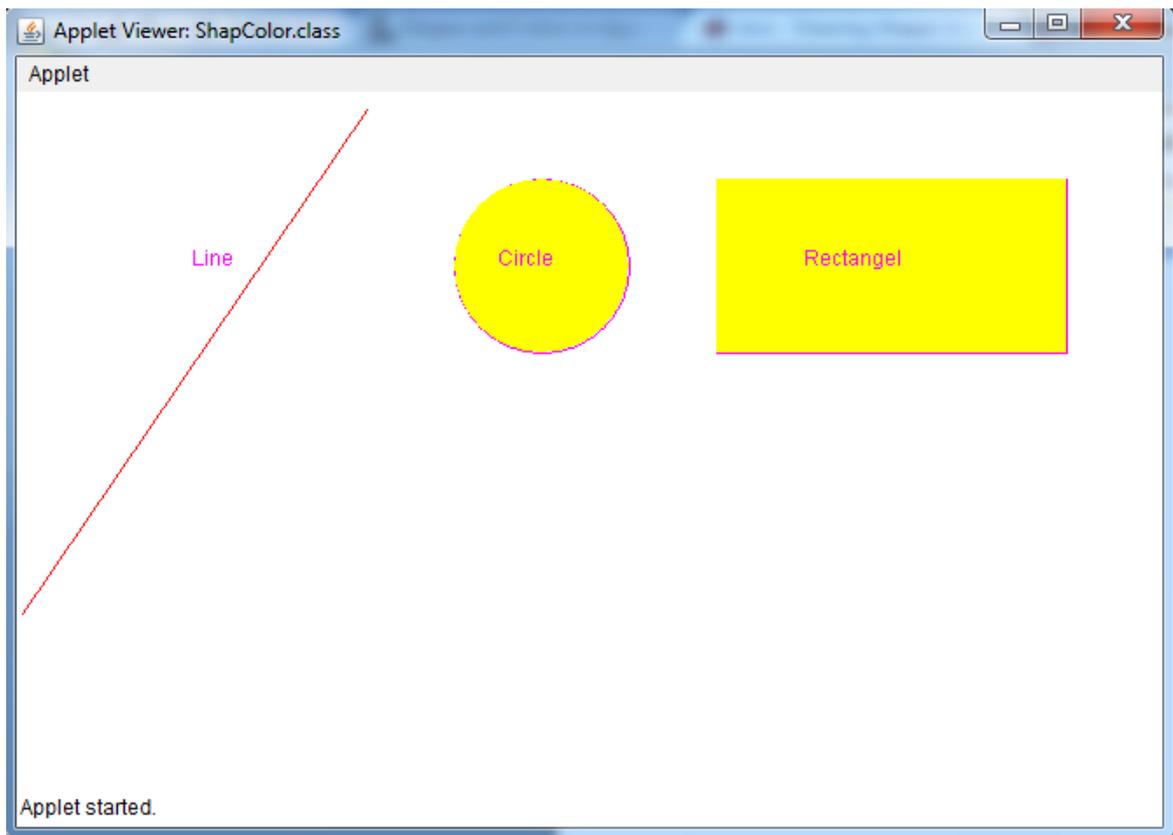
public void paint( Graphics g ) {
    g.setColor( Color.gray );
    drawWedge( 2*Math.PI * hours / 12, width/5, g );
    drawWedge( 2*Math.PI * minutes / 60, width/3, g );
    drawHand( 2*Math.PI * seconds / 60, width/2, g );
    g.setColor( Color.white );
    g.drawString( timeString, 10, height-10 );
}
}
```

Output:**Sample Viva Questions:**

- 1)Mention the methods in Graphics class?
- 2)What is the significance of repaint() ?

36) Write a JAVA program to create different shapes and fill colors using Applet.**Aim: A JAVA program to create different shapes and fill colors using Applet.****Program:**

```
import java.applet.*;
import java.awt.*;
/*
<applet code="ShapColor.class" width="500" height="400">
</applet>
*/
public class ShapColor extends Applet{
    int x=300,y=100,r=50;
    public void paint(Graphics g){
        g.setColor(Color.red); //Drawing line color is red
        g.drawLine(3,300,200,10);
        g.setColor(Color.magenta);
        g.drawString("Line",100,100);
        g.drawOval(x-r,y-r,100,100);
        g.setColor(Color.yellow); //Fill the yellow color in circle
        g.fillOval( x-r,y-r, 100, 100 );
        g.setColor(Color.magenta);
        g.drawString("Circle",275,100);
        g.drawRect(400,50,200,100);
        g.setColor(Color.yellow); //Fill the yellow color in rectangel
        g.fillRect( 400, 50, 200, 100 );
        g.setColor(Color.magenta);
        g.drawString("Rectangel",450,100);
    }
}
```

Output:**Sample Viva Questions:**

- 1) How do Applets differ from Applications?
- 2) Can we pass parameters to an applet from HTML page to an applet? How?

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

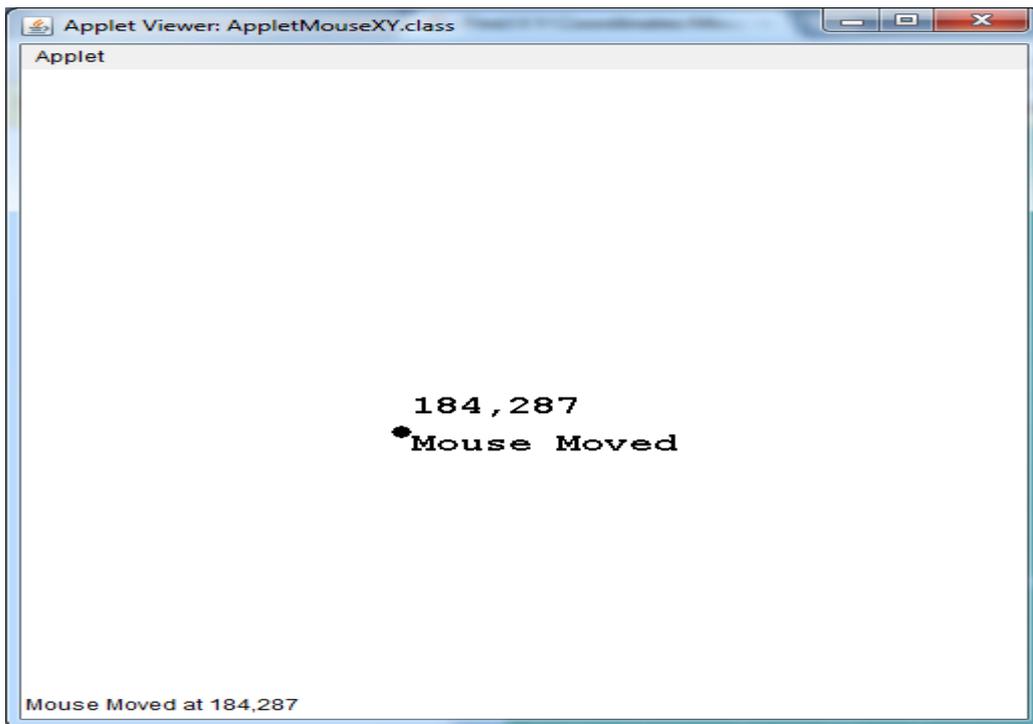
Aim: A JAVA program that display the x and y position of the cursor movement using Mouse.

Program:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;
/*
<applet code="AppletMouseXY.class" width="500" height="500">
</applet>
*/
public class AppletMouseXY extends Applet implements MouseListener,
MouseMotionListener
{
    int x, y;
    String str="";
    public void init()
    {
        addMouseListener(this);
        addMouseMotionListener(this);
    }
        // override ML 5 abstract methods
    public void mousePressed(MouseEvent e)
    {
        x = e.getX();
        y = e.getY();
        str = "Mouse Pressed";
        repaint();
    }
    public void mouseReleased(MouseEvent e)
    {
        x = e.getX();
        y = e.getY();
        str = "Mouse Released";
        repaint();
    }
    public void mouseClicked(MouseEvent e)
    {
        x = e.getX();
        y = e.getY();
        str = "Mouse Clicked";
        repaint();
    }
}
```

```
public void mouseEntered(MouseEvent e)
{
    x = e.getX();
    y = e.getY();
    str = "Mouse Entered";
    repaint();
}
public void mouseExited(MouseEvent e)
{
    x = e.getX();
    y = e.getY();
    str = "Mouse Exited";
    repaint();
}
// override two abstract methods of MouseMotionListener
public void mouseMoved(MouseEvent e)
{
    x = e.getX();
    y = e.getY();
    str = "Mouse Moved";
    repaint();
}
public void mouseDragged(MouseEvent e)
{
    x = e.getX();
    y = e.getY();
    str = "Mouse dragged";
    repaint();
}
// called by repaint() method
public void paint(Graphics g)
{
    g.setFont(new Font("Monospaced", Font.BOLD, 20));
    g.fillOval(x, y, 10, 10);
    g.drawString(x + "," + y, x+10, y -10);
    g.drawString(str, x+10, y+20);
    showStatus(str + " at " + x + "," + y);
}
}
```

Output



Sample Viva Questions:

- 1) Can applets on different pages communicate with each other?
- 2) Which classes and interfaces does Applet class consist?

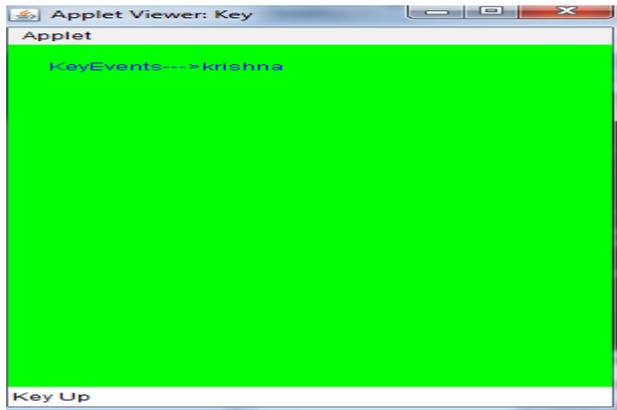
38) Write a JAVA program that identifies key-up key-down event user entering text in a Applet.

Aim: A JAVA program that identifies key-up key-down event user entering text in a Applet.

Program:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
<applet code="Key" width=300 height=400>
</applet>
*/
public class Key extends Applet
implements KeyListener
{
    int X=20,Y=30;
    String msg="KeyEvents--->";
    public void init()
    {
        addKeyListener(this);
        requestFocus();
        setBackground(Color.green);
        setForeground(Color.blue);
    }
    public void keyPressed(KeyEvent k)
    {
        showStatus("KeyDown");
        int key=k.getKeyCode();
        switch(key)
        {
            case KeyEvent.VK_UP:
                showStatus("Move to Up");
                break;
            case KeyEvent.VK_DOWN:
                showStatus("Move to Down");
                break;
            case KeyEvent.VK_LEFT:
                showStatus("Move to Left");
                break;
            case KeyEvent.VK_RIGHT:
                showStatus("Move to Right");
                break;
        }
        repaint();
    }
    public void keyReleased(KeyEvent k)
    {
        showStatus("Key Up");
    }
}
```

```
    }  
    public void keyTyped(KeyEvent k)  
    {  
        msg+=k.getKeyChar();  
        repaint();  
    }  
    public void paint(Graphics g)  
    {  
        g.drawString(msg,X,Y);  
    }  
}
```

Output:**Sample Viva Questions:**

- 1) What tags are mandatory when creating HTML to display an applet?
- 2) What are the Applets information methods?

39) Write a JAVA program to build a Calculator in Swings**Aim:A JAVA program to build a Calculator in Swings****Program:**

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

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

    Calc()
    {
        f=new JFrame("Calculator");
        t=new JTextField();
        b1=new JButton("1");
        b2=new JButton("2");
        b3=new JButton("3");
        b4=new JButton("4");
        b5=new JButton("5");
        b6=new JButton("6");
        b7=new JButton("7");
        b8=new JButton("8");
        b9=new JButton("9");
        b0=new JButton("0");
        bdiv=new JButton("/");
        bmul=new JButton("*");
        bsub=new JButton("-");
        badd=new JButton("+");
        bdec=new JButton(".");
        beq=new JButton("=");
        bdel=new JButton("Delete");
        bclr=new JButton("Clear");
        t.setBounds(30,40,280,30);
        b7.setBounds(40,100,50,40);
        b8.setBounds(110,100,50,40);
        b9.setBounds(180,100,50,40);
        bdiv.setBounds(250,100,50,40);
        b4.setBounds(40,170,50,40);
        b5.setBounds(110,170,50,40);
        b6.setBounds(180,170,50,40);
        bmul.setBounds(250,170,50,40);
        b1.setBounds(40,240,50,40);
        b2.setBounds(110,240,50,40);
        b3.setBounds(180,240,50,40);
```

```
bsub.setBounds(250,240,50,40);
bdec.setBounds(40,310,50,40);
b0.setBounds(110,310,50,40);
beq.setBounds(180,310,50,40);
badd.setBounds(250,310,50,40);
bdel.setBounds(60,380,100,40);
bclr.setBounds(180,380,100,40);
f.add(t);
f.add(b7);
f.add(b8);
f.add(b9);
f.add(bdiv);
f.add(b4);
f.add(b5);
f.add(b6);
f.add(bmul);
f.add(b1);
f.add(b2);
f.add(b3);
f.add(bsub);
f.add(bdec);
f.add(b0);
f.add(beq);
f.add(badd);
f.add(bdel);
f.add(bclr);
f.setLayout(null);
f.setVisible(true);
f.setSize(350,500);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setResizable(false);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
b6.addActionListener(this);
b7.addActionListener(this);
b8.addActionListener(this);
b9.addActionListener(this);
b0.addActionListener(this);
badd.addActionListener(this);
bdiv.addActionListener(this);
bmul.addActionListener(this);
bsub.addActionListener(this);
bdec.addActionListener(this);
beq.addActionListener(this);
bdel.addActionListener(this);
bclr.addActionListener(this);
}
```

```
public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==b1)
        t.setText(t.getText().concat("1"));
    if(e.getSource()==b2)
        t.setText(t.getText().concat("2"));
    if(e.getSource()==b3)
        t.setText(t.getText().concat("3"));
    if(e.getSource()==b4)
        t.setText(t.getText().concat("4"));
    if(e.getSource()==b5)
        t.setText(t.getText().concat("5"));
    if(e.getSource()==b6)
        t.setText(t.getText().concat("6"));
    if(e.getSource()==b7)
        t.setText(t.getText().concat("7"));
    if(e.getSource()==b8)
        t.setText(t.getText().concat("8"));
    if(e.getSource()==b9)
        t.setText(t.getText().concat("9"));
    if(e.getSource()==b0)
        t.setText(t.getText().concat("0"));
    if(e.getSource()==bdec)
        t.setText(t.getText().concat("."));
    if(e.getSource()==badd)
    {
        a=Double.parseDouble(t.getText());
        operator=1;
        t.setText("");
    }
    if(e.getSource()==bsub)
    {
        a=Double.parseDouble(t.getText());
        operator=2;
        t.setText("");
    }
    if(e.getSource()==bmul)
    {
        a=Double.parseDouble(t.getText());
        operator=3;
        t.setText("");
    }
    if(e.getSource()==bdiv)
    {
        a=Double.parseDouble(t.getText());
        operator=4;
        t.setText("");
    }
    if(e.getSource()==beq)
    {
```

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

    case 2: result=a-b;
        break;

    case 3: result=a*b;
        break;

    case 4: result=a/b;
        break;

    default: result=0;
}

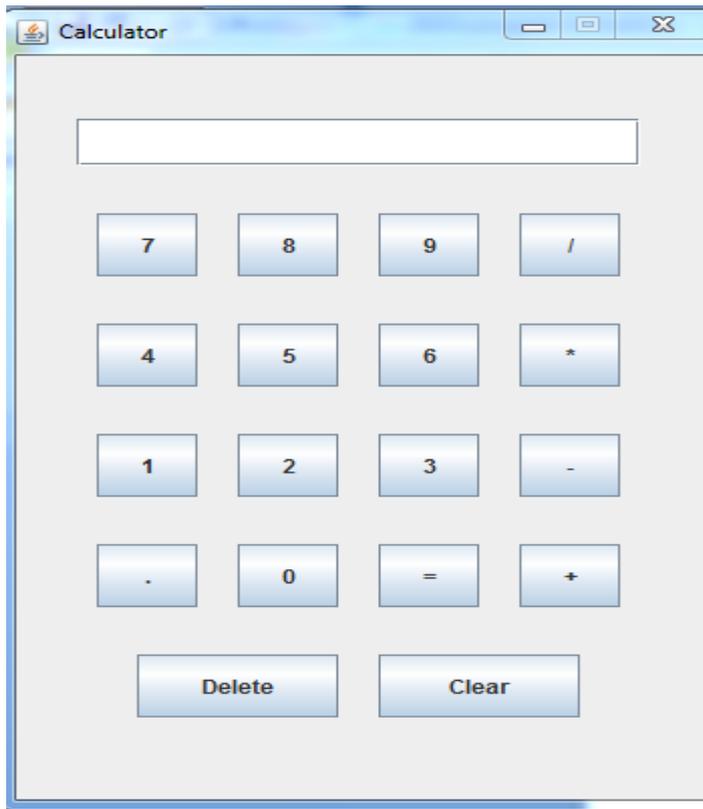
t.setText(""+result);
}

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

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

}

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

Output:**Sample viva Questions:**

- 1) How will you communicate between two Applets?
- 2) What is Difference between AWT and Swing?

40) Write a JAVA program to display the digital watch in swing tutorial.**Aim: A JAVA program to display the digital watch in swing tutorial.****Program:**

```

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

DigitalWatch(){
    f=new JFrame();

    t = new Thread(this);
    t.start();

    b=new JButton();
    b.setBounds(100,100,100,50);

    f.add(b);
    f.setSize(300,400);
    f.setLayout(null);
    f.setVisible(true);
}

public void run() {
    try {
        while (true) {

            Calendar cal = Calendar.getInstance();
            hours = cal.get( Calendar.HOUR_OF_DAY );
            if ( hours > 12 ) hours -= 12;
            minutes = cal.get( Calendar.MINUTE );
            seconds = cal.get( Calendar.SECOND );

            SimpleDateFormat formatter = new SimpleDateFormat("hh:mm:ss");
            Date date = cal.getTime();
            timeString = formatter.format( date );

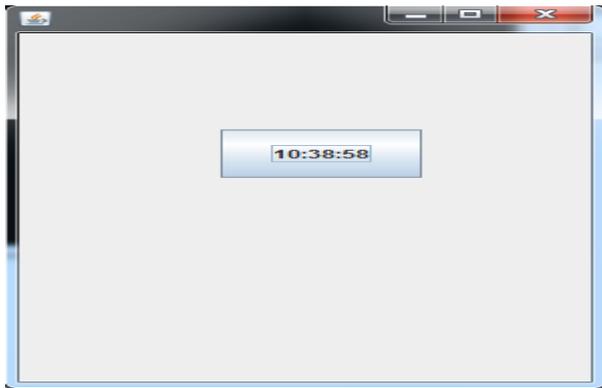
            printTime();

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

```

```
}  
  
public void printTime(){  
    b.setText(timeString);  
}  
  
public static void main(String[] args) {  
    new DigitalWatch();  
  
}  
}
```

Output:



Sample viva Questions:

- 1) Why do you Canvas?
- 2) What type of sound file formats can I use for the applets?

41) Write a JAVA program that to create a single ball bouncing inside a JPanel**Aim: A JAVA program that to create a single ball bouncing inside a JPanel****Program:**

```
import java.awt.*;
import javax.swing.*;

public class BouncingBall extends JPanel {

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

    // Ball Size
    float radius = 40;
    float diameter = radius * 2;

    // Center of Call
    float X = radius + 50;
    float Y = radius + 20;

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

    public BouncingBall() {

        Thread thread = new Thread() {
            public void run() {
                while (true) {

                    width = getWidth();
                    height = getHeight();

                    X = X + dx ;
                    Y = Y + dy;

                    if (X - radius < 0) {
                        dx = -dx;
                        X = radius;
                    } else if (X + radius > width) {
                        dx = -dx;
                        X = width - radius;
                    }

                    if (Y - radius < 0) {
```

```
        dy = -dy;
        Y = radius;
    } else if (Y + radius > height) {
        dy = -dy;
        Y = height - radius;
    }
    repaint();

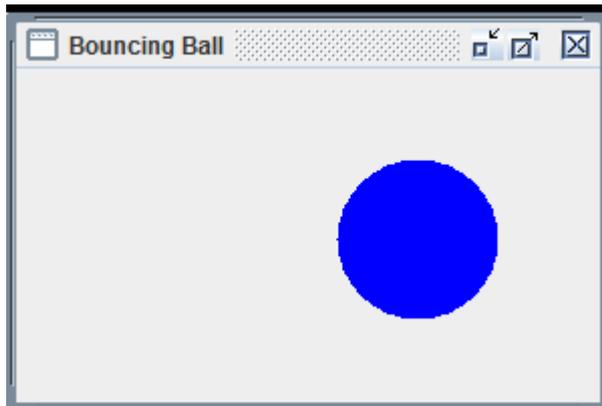
    try {
        Thread.sleep(50);
    } catch (InterruptedException ex) {
    }

    }
}
};
thread.start();
}

public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setColor(Color.BLUE);
    g.fillOval((int)(X-radius), (int)(Y-radius), (int)diameter, (int)diameter);
}

public static void main(String[] args) {
    JFrame.setDefaultLookAndFeelDecorated(true);
    JFrame frame = new JFrame("Bouncing Ball");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(300, 200);
    frame.setContentPane(new BouncingBall());
    frame.setVisible(true);
}
}
```

Output:



Sample viva Questions:

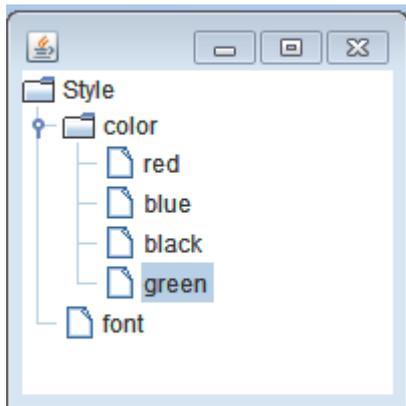
- 1)What are the attributes of Applet Tags?
- 2)Explain how to read information from the applet parameters?

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

Aim: A JAVA program JTree as displaying a real tree upside down.

Program:

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

Output:**Sample viva Questions:**

- 1) Explain how to play sound in applet?
- 2) What is the significance of drawstring method?