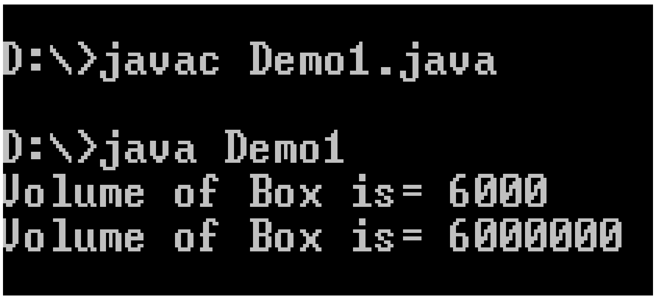
## 1. Write a Java Program of constructor and method.

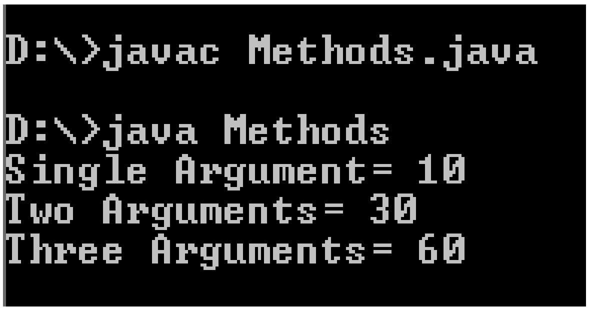
```
class Box
        {
                int length, breadth, height;                    // Class Variable

                Box (int l, int b, int h)                       //Constructor
                {
                        length = l;
                        breadth = b;
                        height = h;
                }
                void display()                                  //Method
                {
                        int volume=length*breadth*height;
                        System.out.println ("Volume of Box is= "+volume);
                }
        }

class Demo1
{
        public static void main(String M[])
        {
                Box b1=new Box (10, 20, 30);

                Box b2=new Box (100, 200, 300);

                b1.display();

                b2.display();

        }
}
```



```
D:\>javac Demo1.java

D:\>java Demo1
Volume of Box is= 6000
Volume of Box is= 6000000
```

## 2. Write a Java Program of **Methods Overloading**

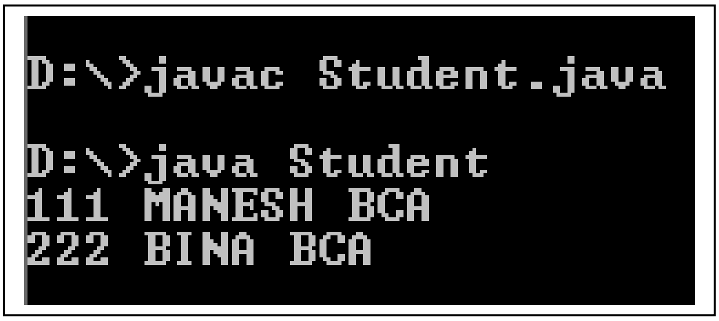
```
class Demo
        {
                int a,b;

                void display(int x)
                {
                        System.out.println ("Single Argument= "+X);
                }

                void display(int x, int y)
                {
                        a=x+y;
                        System.out.println ("Two Arguments= "+a);
                }

                void display(int x, int y, int z)
                {
                        b=x+y+z;
                        System.out.println ("Three Arguments= "+b);
                }
        }

class Methods
{
        public static void main(String M[])
        {
                Demo D1=new Demo();

                D1.display(10);

                D1.display(10,20);

                D1.display(10,20,30);

           }
      }
```



```
D:\>javac Methods.java

D:\>java Methods
Single Argument= 10
Two Arguments= 30
Three Arguments= 60
```

## 3. Write a Java Program of static variable.

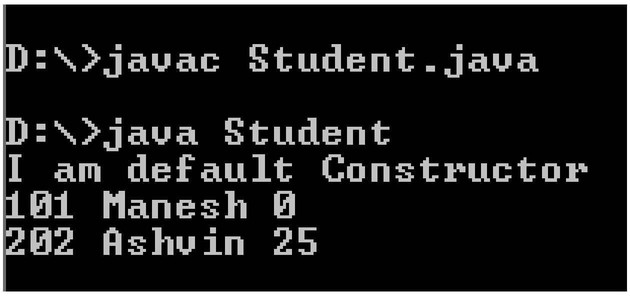
```
class Student
{
        int rollno;
        String name;
        static String college ="BCA";

       Student (int r, String n)
       {
              rollno = r;
              name = n;
       }

       void display ()
       {
              System.out.println(rollno+" "+name+" "+college);
       }


        public static void main(String args[])
       {
              Student s1 = new Student (111,"MANESH");
              Student s2 = new Student (222,"BINA");

              s1.display();
              s2.display ();
       }

}
```



```
D:\>javac Student.java

D:\>java Student
111 MANESH BCA
222 BINA BCA
```

## 4. Write a Java Program of Constructor Overloading.

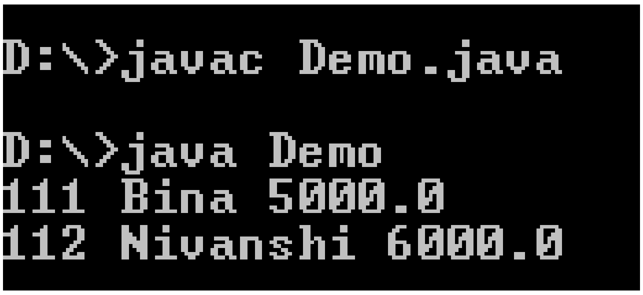
```
class Student
{
    int id, age;
    String name;
    Student()
    {
            System.out.println ("I am default Constructor");
    }
    Student (int i, String n)
    {
            id = i;
            name = n;
    }

    Student (int x, String y, int z)
    {
            id = x;
            name = y;
            age = z;
    }
    void display()
    {
            System.out.println (id+" "+name+" "+age);
    }
    public static void main (String M[])
     {
            Student s = new Student();
            Student s1 = new Student (101, "Manesh");
            Student s2 = new Student (202, "Ashvin", 25);
            s1.display ();
            s2.display ();
     }
  }
```



```
D:\>javac Student.java

D:\>java Student
I am default Constructor
101 Manesh 0
202 Ashvin 25
```

## 5. Write a Java Program of this keyword

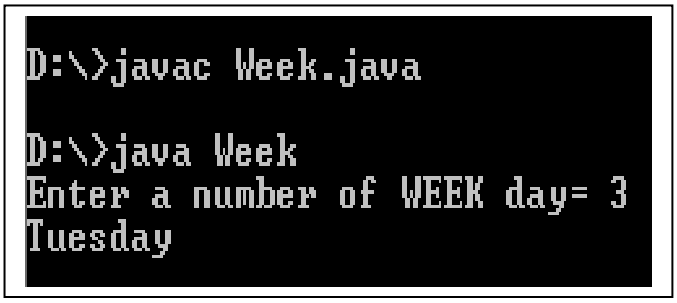
```
class Student
{
        int rollno;
        String name;
        float fee;

        Student (int rollno, String name, float fee)
        {
                this.rollno = rollno;
                this.name = name;
                this.fee = fee;
        }

        void display ()
        {
                System.out.println (rollno+" "+name+" "+fee);
        }
}


class Demo
    {
        public static void main (String M[])
        {
                Student s1=new Student (111, "Bina", 5000f);

                Student s2=new Student (112, "Nivanshi", 6000f);

                s1.display ();

                s2.display ();

     }
 }
```



```
D:\>javac Demo.java

D:\>java Demo
111 Bina 5000.0
112 Nivanshi 6000.0
```

## 6. Write a Java Program of switch case.

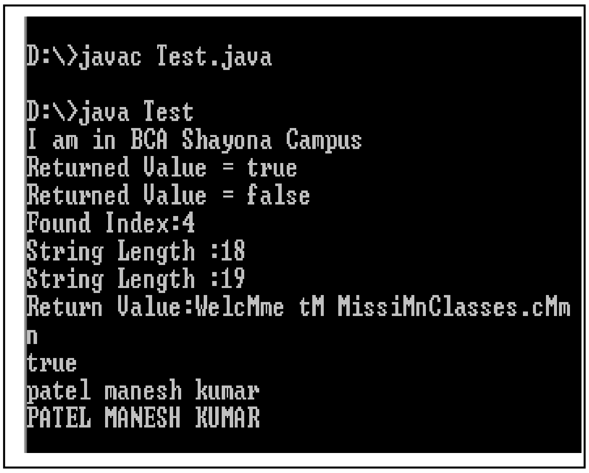
```
import java.util.*;

class Week
{
      public static void main(String Mehul[])
    {
              int no;

              Scanner S=new Scanner (System.in);
              System.out.print ("Enter a number of WEEK day= ");
               no=S.nextInt();

        switch(no)
         {
              case 1:
                      System.out.println ("sunday");
                      break;
              case 2:
                      System.out.println ("Monday");
                      break;
              case 3:
                      System.out.println ("Tuesday");
                      break;
              case 4:
                      System.out.println ("Wednesday");
                      break;
              default:
                      System.out.println ("Plz…enter number between 1 and 4");
                      break;
        }

    }
}
```



```
D:\>javac Week.java

D:\>java Week
Enter a number of WEEK day= 3
Tuesday
```

## 7. Write a Java Program of String class methods.

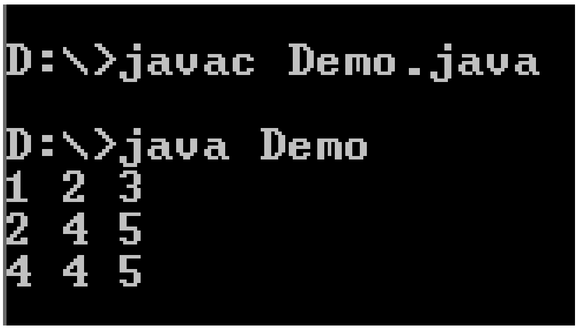
```
class Test {
  public static void main(String M[]) {

         String s = "I am in BCA";
         s = s.concat (" Shayona Campus");
         System.out.println(s);

        String Str = new String ("This is really not immutable");
        boolean retVal;

         retVal = Str.endsWith ( "immutable" );
         System.out.println ("Returned Value = " + retVal);

        retVal = Str.endsWith( "immu" );
        System.out.println ("Returned Value = " + retVal);

        String Str2 = new String ("Welcome to Shayona BCA College");
        System.out.print ("Found Index:");
        System.out.println (Str2.indexOf ( 'o'));

        String Str3 = new String (“Patel Manesh kumar");
         String Str4 = new String ("Manesh is a Teacher");

        System.out.print ("String Length :");
        System.out.println (Str3.length());

        System.out.print ("String Length :");
        System.out.println(Str4.length());

        String Str5 = new String ("Welcome to MissionClasses.com");

        System.out.print ("Return Value:");
        System.out.println (Str5.replace ('o', 'M'));

        String s6 = "Patel Manesh kumar";
        char result = s6.charAt(8);
        System.out.println (result);

        System.out.println (s6.startsWith ("Patel"));
        System.out.println (s6.toLowerCase ());
        System.out.println (s6.toUpperCase ());
    }
}
```



```
D:\>javac Test.java

D:\>java Test
I am in BCA Shayona Campus
Returned Value = true
Returned Value = false
Found Index:4
String Length :18
String Length :19
Return Value:WelcMme tM MissiMnClasses.cMm
n
true
patel manesh kumar
PATEL MANESH KUMAR
```

## 8. Write a Java Program of Two Dimensional Array.

```
class Demo
{
        public static void main(String S[])
        {

                int arr[][]={{1,2,3},{2,4,5},{4,4,5}};

                for (int i=0;i<3;i++)
                {
                        for (int j=0;j<3;j++)
                        {
                                System.out.print (arr[i][j]+" ");
                        }
                        System.out.println ();
                }
        }
  }
```

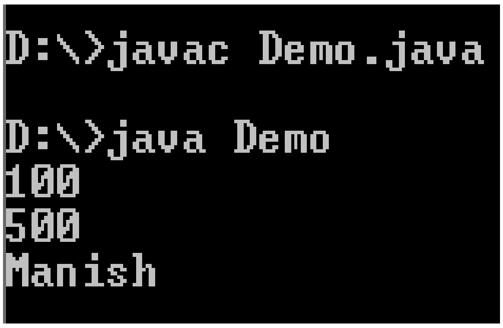


```
D:\>javac Demo.java

D:\>java Demo
1 2 3
2 4 5
4 4 5
```

## 9. Write a Java Program of *Multilevel Inheritance*

```
class A
{
        int no1;
        String name;

        void get()
        {
                no1 = 100;
                name = “Manish”;
        }
}

class B extends A
{
        int no2;
        void put()
        {
                no2 = 500;
        }
}

class C extends B
{
        void display()
        {
                System.out.println (no1);
                System.out.println (no2);
                System.out.println (name);
        }
}
```

```
class Demo
{
  public static void main (String M[])
        {
                C obj = new C();

                obj.get();

                obj.put();

                obj.display();

        }
}
```

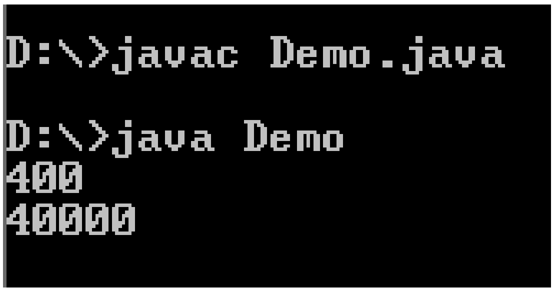


```
D:\>javac Demo.java

D:\>java Demo
100
500
Manish
```

## 10. Write a Java Program of **Hierarchical Inheritance.**

```
class A
{
        int no1,no2;

        void get()
        {
                no1 = 200;
                no2 = 200;
        }
}

class B extends A
{
        int sum;
        void add()
        {
                sum = no1 + no2;
                System.out.println (sum);
        }
}

class C extends A
{
        int res;

        void mul()

        {
                res = no1 * no2;
                System.out.println (res);
        }
}
```

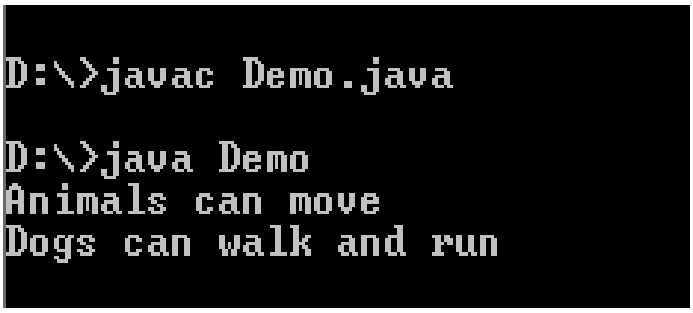
```
class Demo
{
  public static void main (String M[])
        {
                B obj1 = new B();

                C obj2 = new C();


                obj1.get();

                obj1.add();


                obj2.get();

                obj2.mul();

        }
}
```



```
D:\>javac Demo.java

D:\>java Demo
400
40000
```

## 11. Write a Java Program of **Method overriding.**

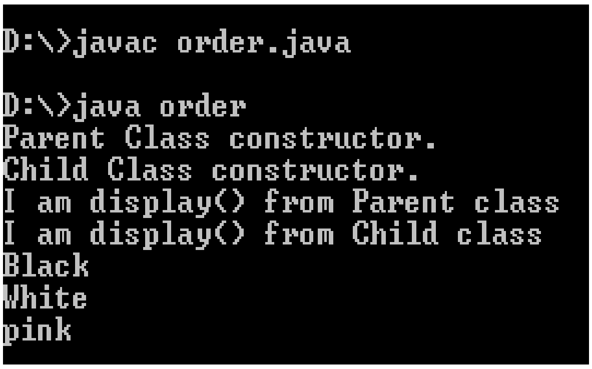
```
class Animal
{
        public void move()
        {
                System.out.println ("Animals can move");
         }
}

class Dog extends Animal
{
        public void move()
        {
                System.out.println ("Dogs can walk and run");
        }
}

class Demo
{
          public static void main (String M[])
         {
                Animal a = new Animal();
                Animal b = new Dog();

                a.move();

                b.move();

          }
}
```



```
D:\>javac Demo.java

D:\>java Demo
Animals can move
Dogs can walk and run
```

## 12. Write a Java Program of super concept.

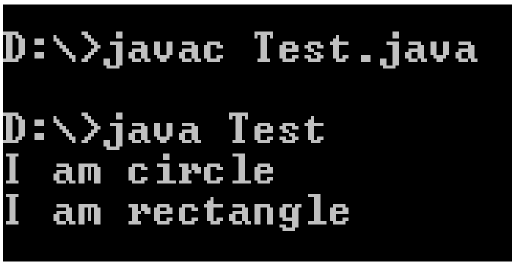
```
class Parent
{
          String color="White";

      Parent()
    {
       System.out.println ("Parent Class constructor.");
    }

     void display()
    {
       System.out.println ("I am display() from Parent class");
     }
}
class Child extends Parent
{
          String color="Black";
          Child()
          {
                    super();
                    System.out.println ("Child Class constructor calling.");
          }

          void display()
         {
          super.display();
          System.out.println ("I am display() from Child class ");

          System.out.println (color);
          System.out.println (super.color);
          super.color=”pink”;
          System.out.println (super.color);
          }
  }
class order
{
    public static void main (String M[])
    {
       Child C=new Child();
       C.display();
   }
}
```



```
D:\>javac order.java

D:\>java order
Parent Class constructor.
Child Class constructor.
I am display() from Parent class
I am display() from Child class
Black
White
pink
```

## 13. Write a Java Program of Abstract class.

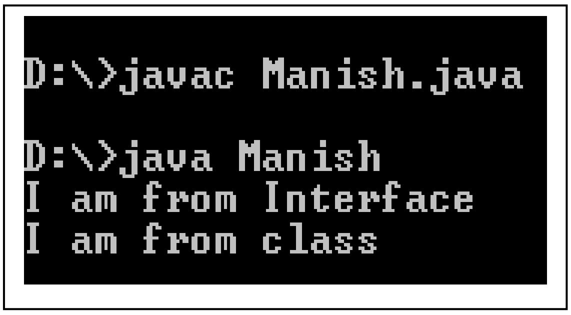
```
abstract class Shape
{
     abstract void show();
}

class Rectangle extends Shape
{
     void show()
     {
        System.out.println ("I am rectangle");
      }
}

class Circle extends Shape
  {
     void show()
      {
         System.out.println ("I am circle");
       }
}

class Test
{
public static void main (String M[])
    {
        Shape s=new Circle();                //Reference
        s.show();

        Shape s1=new Rectangle();           //Reference
        s1.show();
     }
}
```



```
D:\>javac Test.java

D:\>java Test
I am circle
I am rectangle
```

14. Write a Java Program of **Multiple Inheritance by Interface.**

```
interface ABC                    // interface
{
     void print();
}

class XYZ                        // class
{
          void show()
       {
            System.out.println ("I am from class");
       }

}

class Manish extends XYZ implements ABC
{
       public void print()
       {
            System.out.println ("I am from Interface");
       }

       public static void main(String args[])
       {
            Manish obj = new Manish();

            obj.print();               // method of interface

            obj.show();                // method of class

        }
}
```



```
D:\>javac Manish.java

D:\>java Manish
I am from Interface
I am from class
```