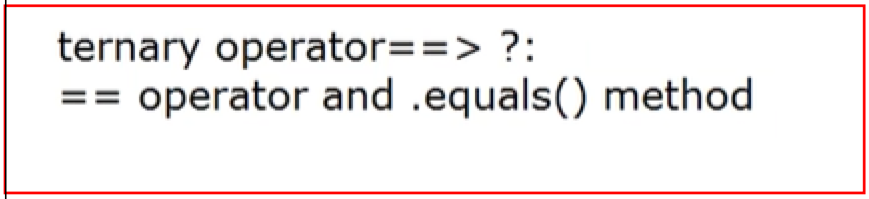
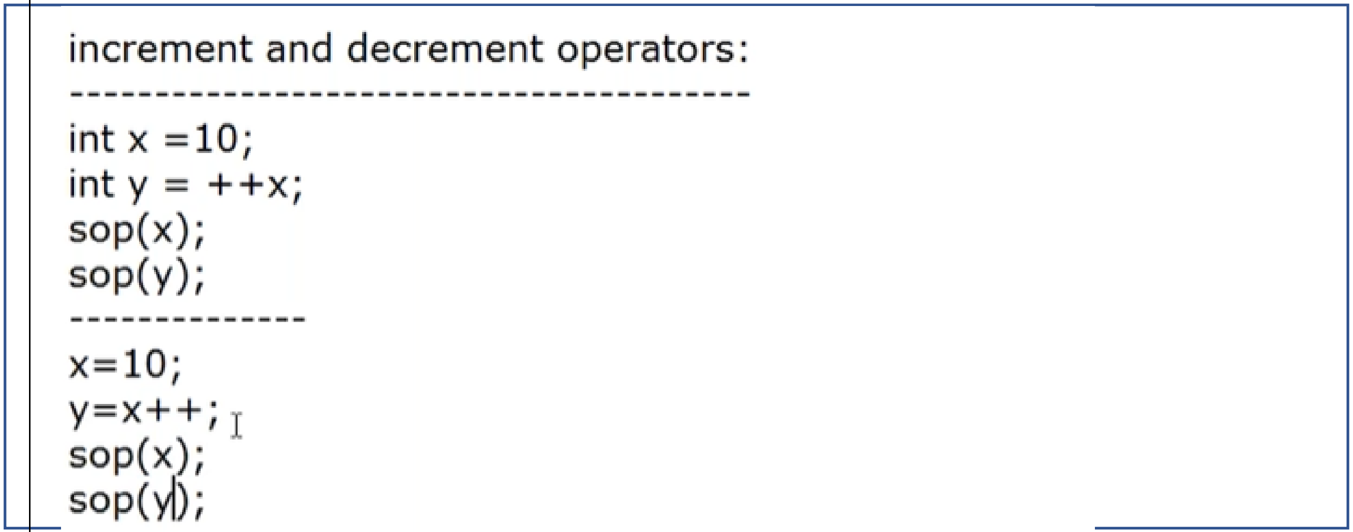
Operator and Assignments

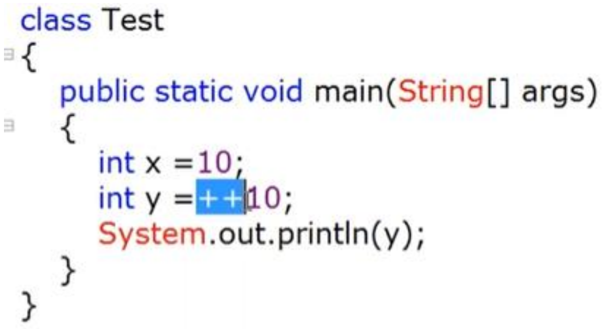


ternary operator==> ?:
== operator and .equals() method

VVIP

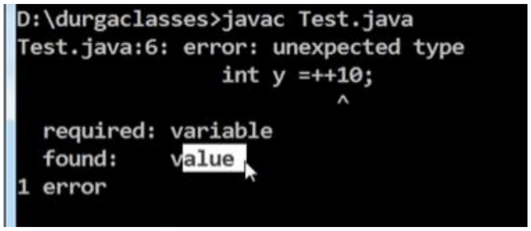


```
increment and decrement operators:
--------------------------------------
int x =10;
int y = ++x;
sop(x);
sop(y);
-------------
x=10;
y=x++;
sop(x);
sop(y);
```

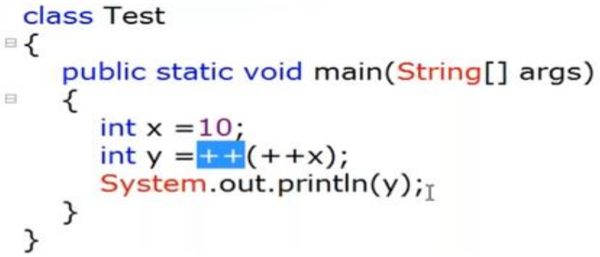


```
class Test
{
    public static void main(String[] args)
    {
        int x =10;
        int y =++10;
        System.out.println(y);
    }
}
```

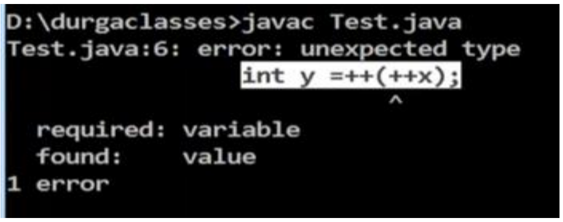
This will give compile time error



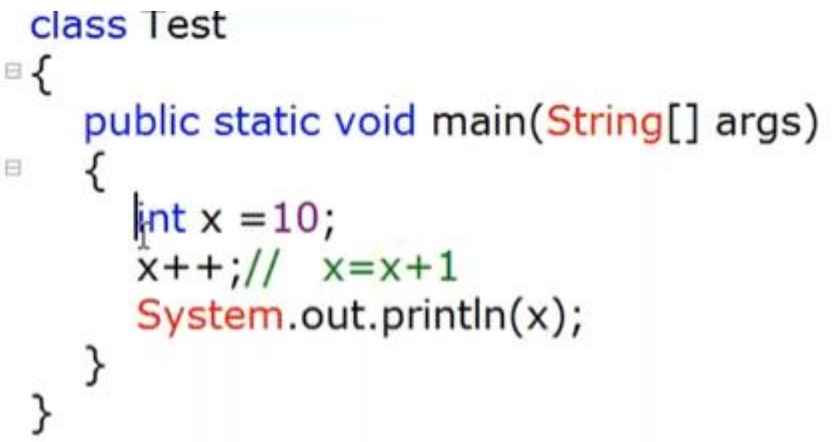
```
D:\durgaclasses>javac Test.java
Test.java:6: error: unexpected type
                int y =++10;
                         ^
  required: variable
  found:    value
1 error
```



```
class Test
{
    public static void main(String[] args)
    {
        int x =10;
        int y =++(++x);
        System.out.println(y);
    }
}
```

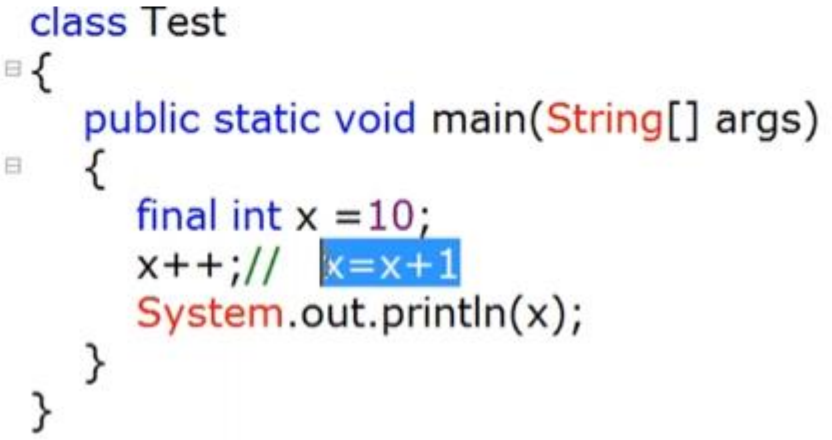


```
D:\durgaclasses>javac Test.java
Test.java:6: error: unexpected type
                int y =++(++x);
                         ^
  required: variable
  found:    value
1 error
```



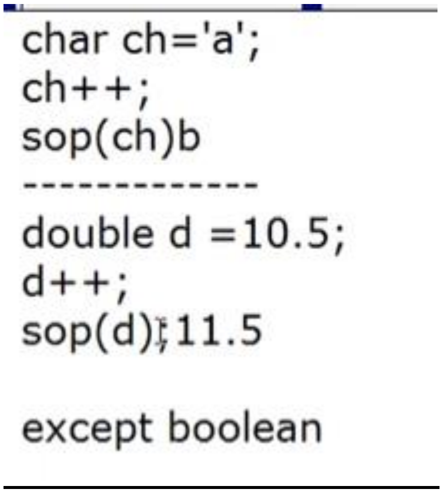
```
class Test
{
    public static void main(String[] args)
    {
        int x =10;
        x++;//   x=x+1
        System.out.println(x);
    }
}
```

Runs smoothly

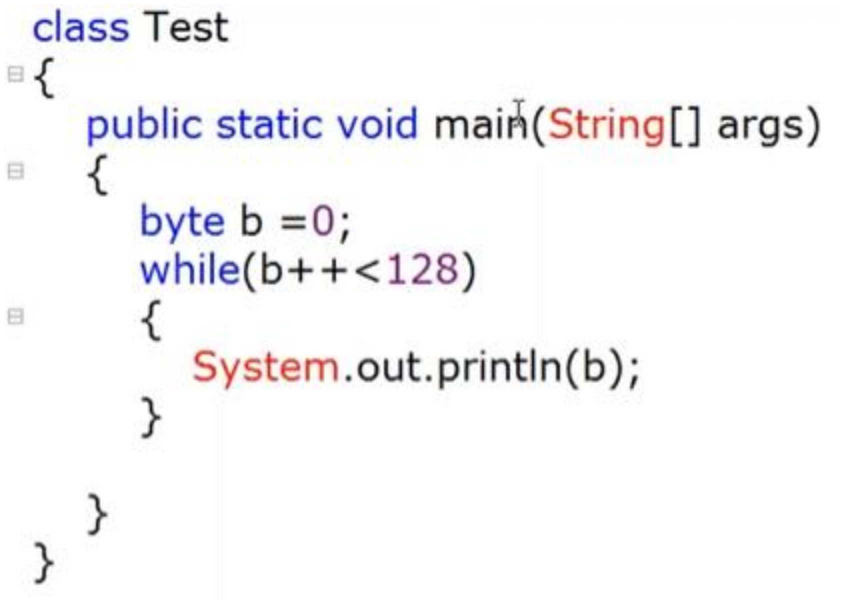


```
class Test
{
    public static void main(String[] args)
    {
        final int x =10;
        x++;//   x=x+1
        System.out.println(x);
    }
}
```

Compile time error because of reassignment

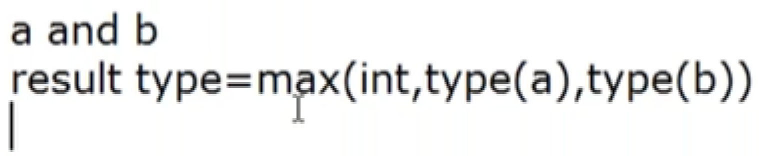


```
char ch='a';
ch++;
sop(ch)b
-------------
double d =10.5;
d++;
sop(d);11.5

except boolean
```



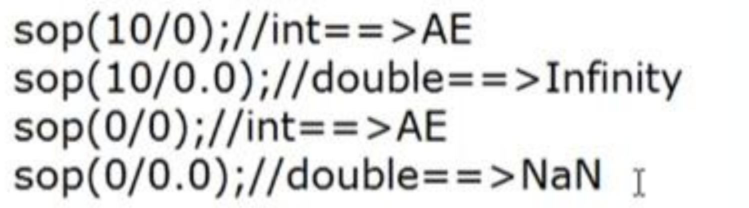
```
class Test
{
    public static void main(String[] args)
    {
        byte b =0;
        while(b++<128)
        {
            System.out.println(b);
        }

    }
}
```

Infinite loop

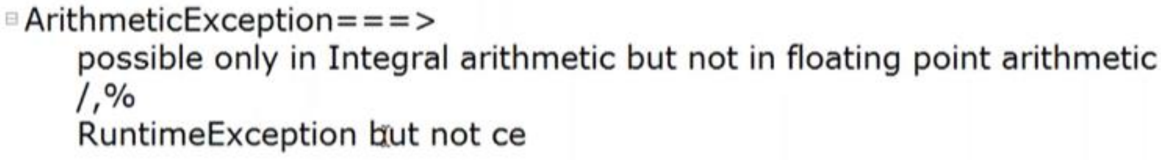


```
a and b
result type=max(int,type(a),type(b))
```

formula for conversion

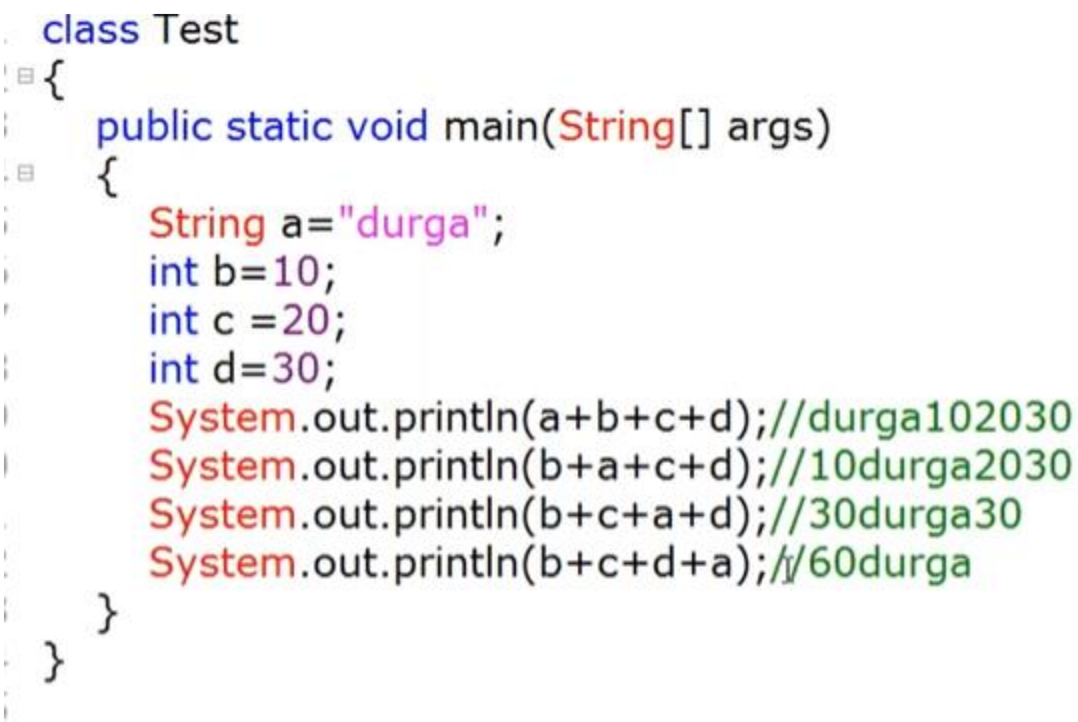


```
sop(10/0);//int==>AE
sop(10/0.0);//double==>Infinity
sop(0/0);//int==>AE
sop(0/0.0);//double==>NaN
```



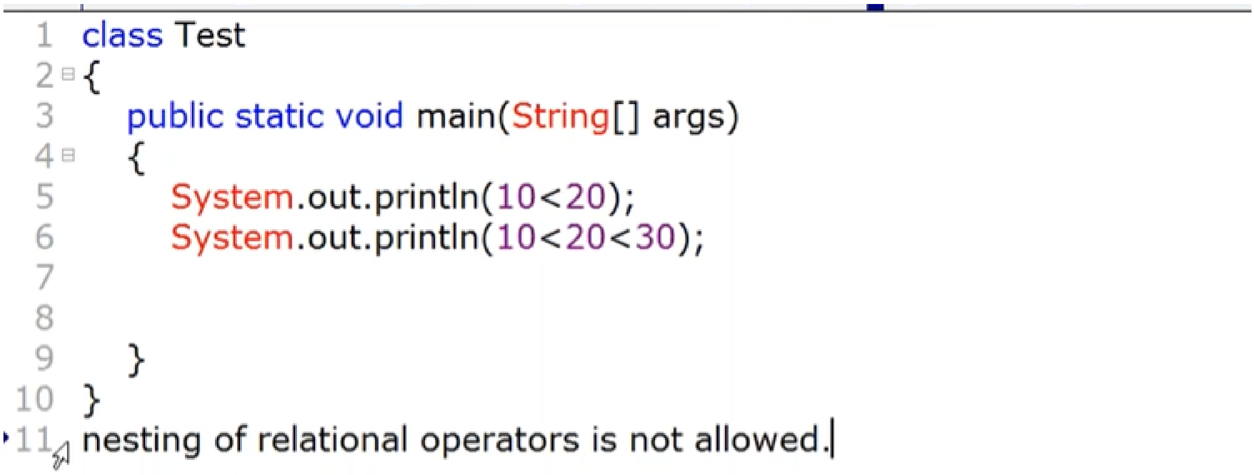
```
ArithmeticException===>
    possible only in Integral arithmetic but not in floating point arithmetic
    /,%
    RuntimeException but not ce
```

String concatenation operator



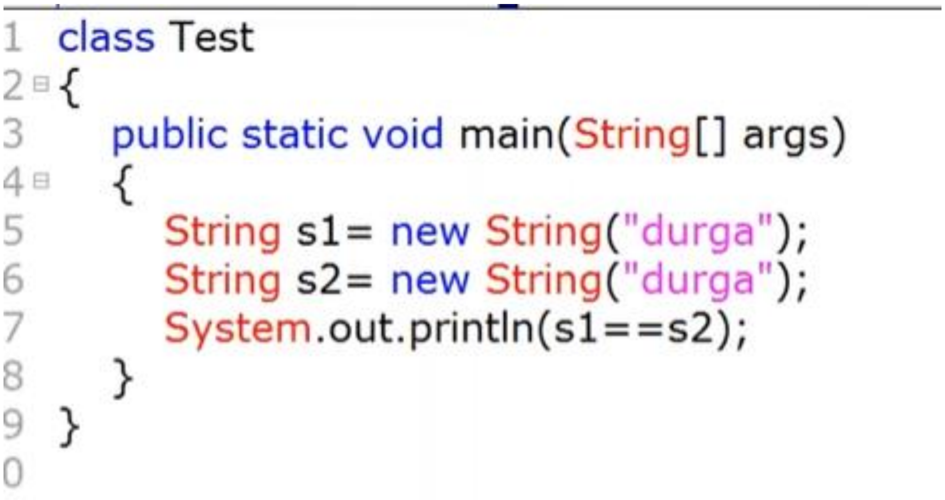
```
class Test
{
    public static void main(String[] args)
    {
        String a="durga";
        int b=10;
        int c =20;
        int d=30;
        System.out.println(a+b+c+d);//durga102030
        System.out.println(b+a+c+d);//10durga2030
        System.out.println(b+c+a+d);//30durga30
        System.out.println(b+c+d+a);//60durga
    }
}
```

Relational Operator

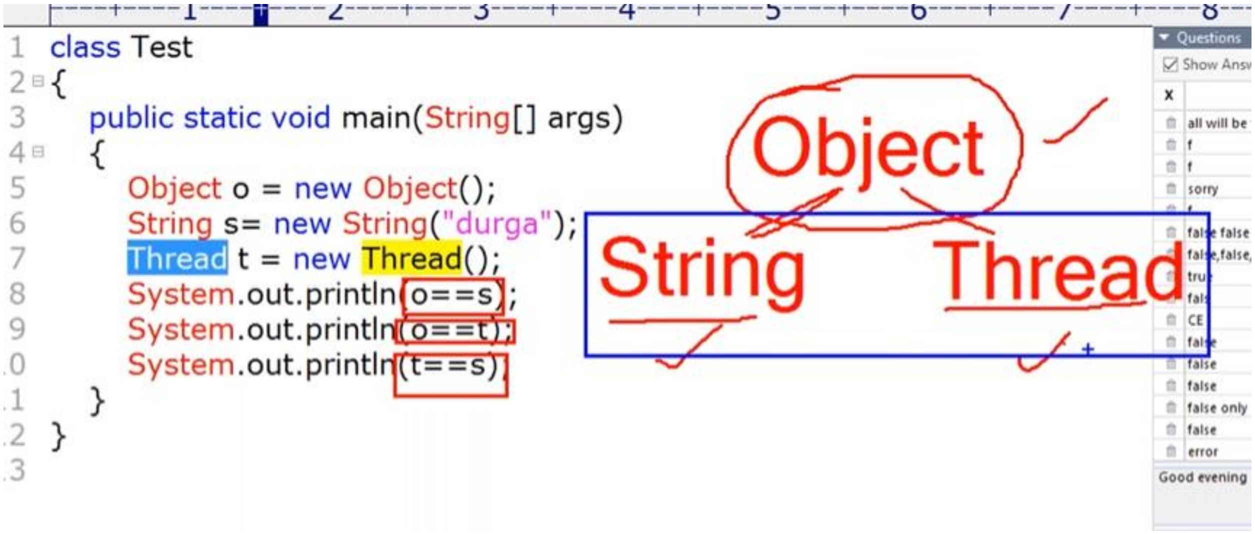


```
class Test
{
    public static void main(String[] args)
    {
        System.out.println(10<20);
        System.out.println(10<20<30);


    }
}
nesting of relational operators is not allowed.
```

Relational operator



```
class Test
{
    public static void main(String[] args)
    {
        String s1= new String("durga");
        String s2= new String("durga");
        System.out.println(s1==s2);
    }
}
```



```
class Test
{
    public static void main(String[] args)
    {
        Object o = new Object();
        String s= new String("durga");
        Thread t = new Thread();
        System.out.println(o==s);
        System.out.println(o==t);
        System.out.println(t==s);
    }
}
```

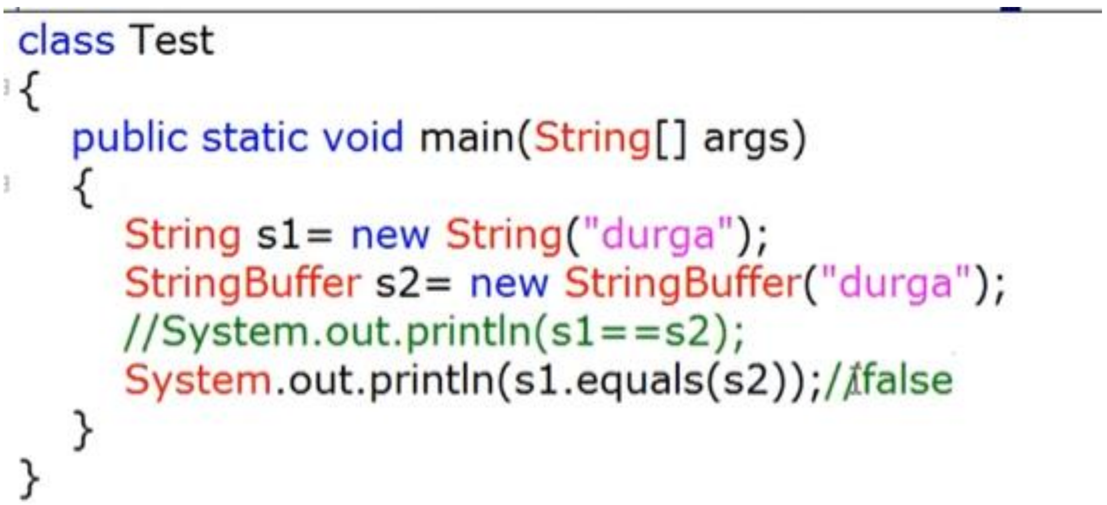
all will be
f
f
sorry
f
false false
false,false,
true
fals
CE
false
false
false
false only
false
error

Good evening

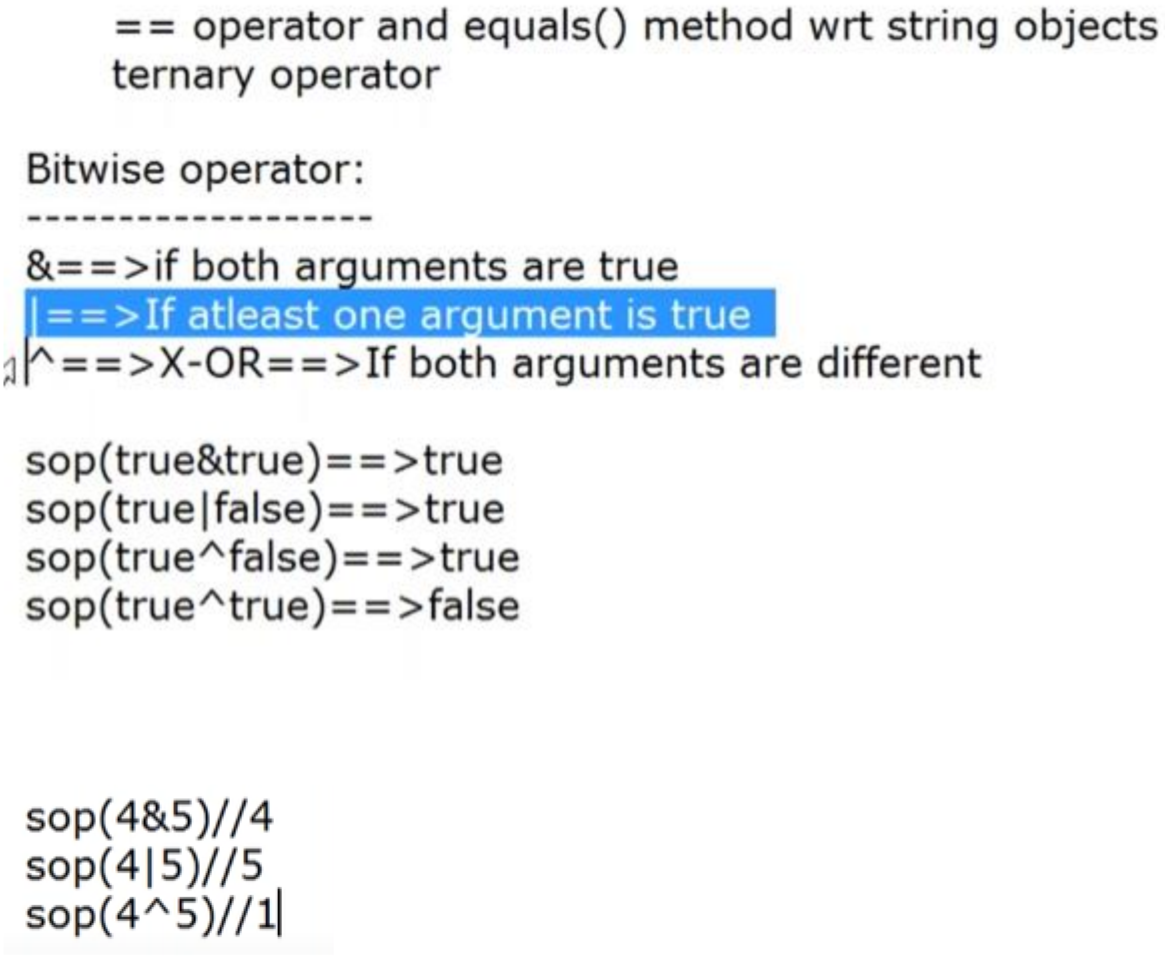
Difference between == operator and equals() method:

---

== reference comparison
equals() method content comparison



```
class Test
{
   public static void main(String[] args)
   {
      String s1= new String("durga");
      StringBuffer s2= new StringBuffer("durga");
      //System.out.println(s1==s2);
      System.out.println(s1.equals(s2));//false
   }
}
```



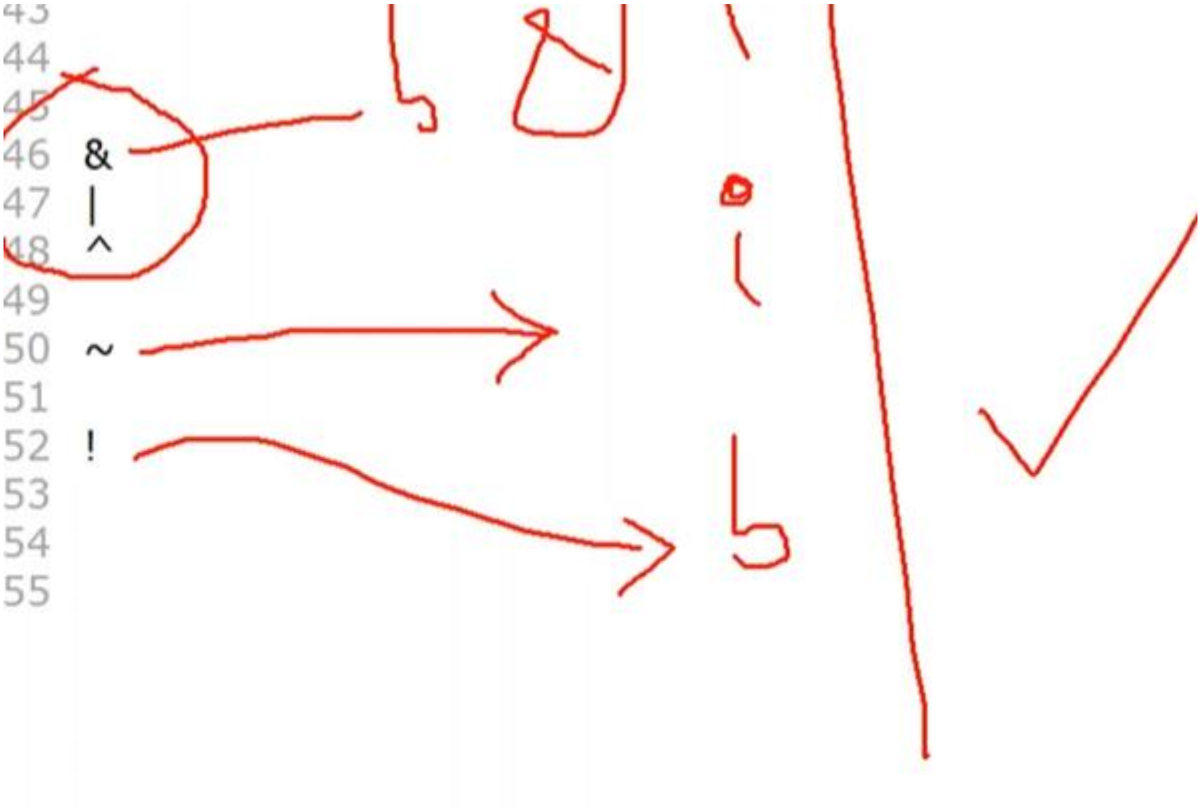
== operator and equals() method wrt string objects
ternary operator

Bitwise operator:

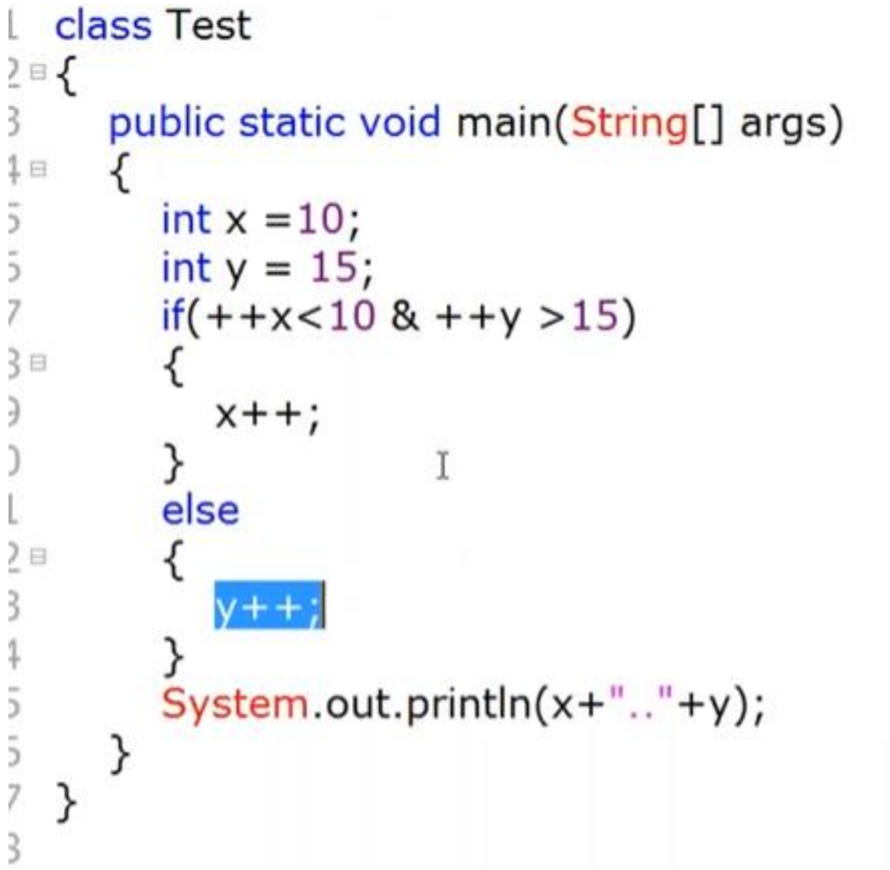
&==>if both arguments are true
|==>If atleast one argument is true
^==>X-OR==>If both arguments are different

```
sop(true&true)==>true
sop(true|false)==>true
sop(true^false)==>true
sop(true^true)==>false
```

```
sop(4&5)//4
sop(4|5)//5
sop(4^5)//1
```

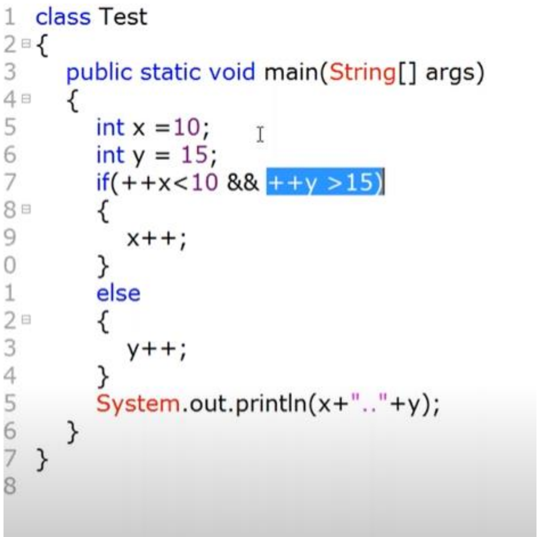


```
&
|
^

~

!
```



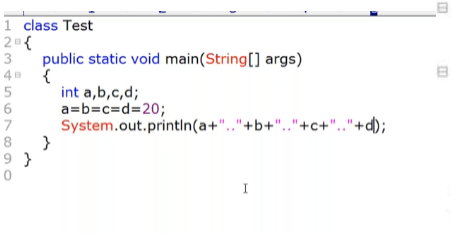
```
class Test
{
   public static void main(String[] args)
   {
      int x =10;
      int y = 15;
      if(++x<10 & ++y >15)
      {
         x++;
      }
      else
      {
         y++;
      }
      System.out.println(x+".."+y);
   }
}
```

11: 17

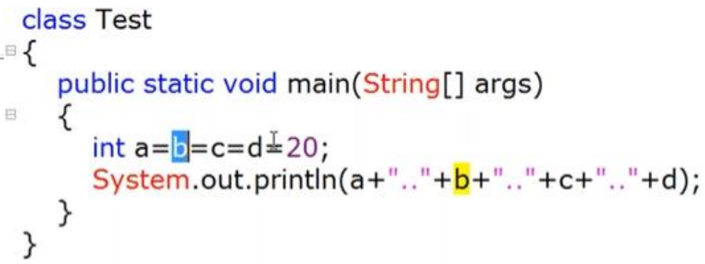


```
class Test
{
   public static void main(String[] args)
   {
      int x =10;
      int y = 15;
      if(++x<10 && ++y >15)
      {
         x++;
      }
      else
      {
         y++;
      }
      System.out.println(x+".."+y);
   }
}
```

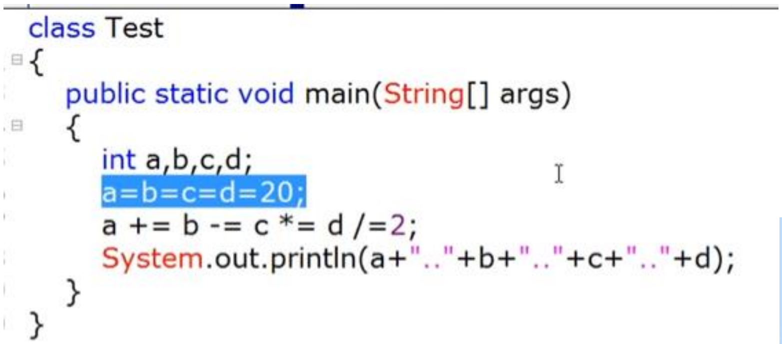
11:16



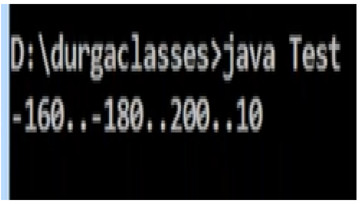
```java
class Test
{
    public static void main(String[] args)
    {
        int a,b,c,d;
        a=b=c=d=20;
        System.out.println(a+".."+b+".."+c+".."+d);
    }
}
```



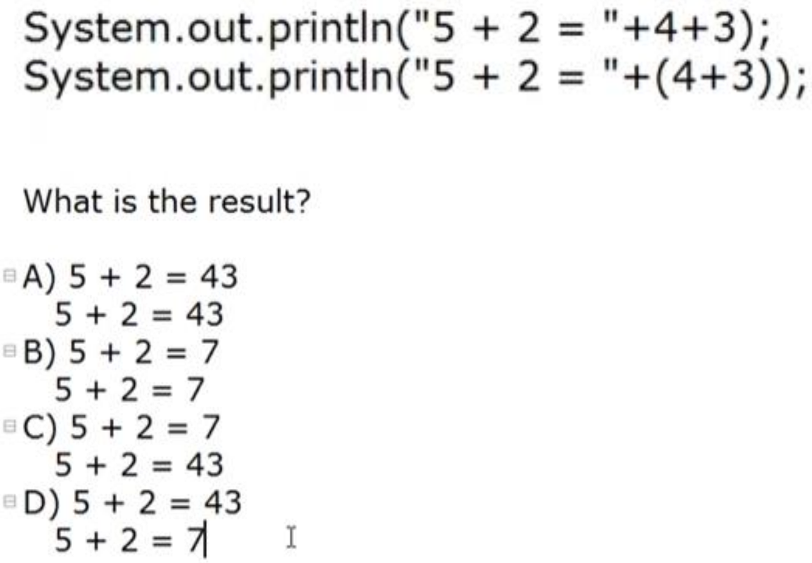
```java
class Test
{
    public static void main(String[] args)
    {
        int a=b=c=d=20;
        System.out.println(a+".."+b+".."+c+".."+d);
    }
}
```



```java
class Test
{
    public static void main(String[] args)
    {
        int a,b,c,d;
        a=b=c=d=20;
        a += b -= c *= d /=2;
        System.out.println(a+".."+b+".."+c+".."+d);
    }
}
```



```
D:\durgaclasses>java Test
-160..-180..200..10
```



```java
System.out.println("5 + 2 = "+4+3);
System.out.println("5 + 2 = "+(4+3));
```

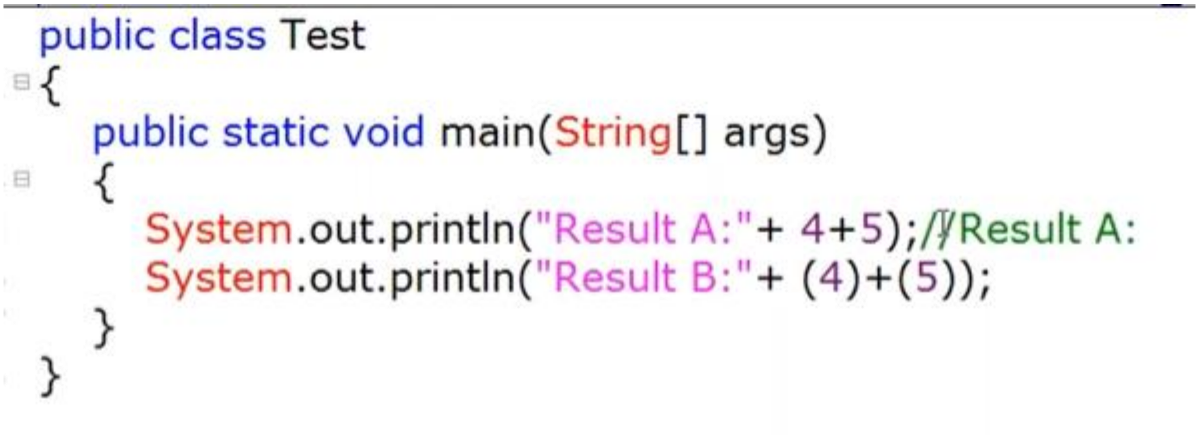
What is the result?

A) 5 + 2 = 43
   5 + 2 = 43

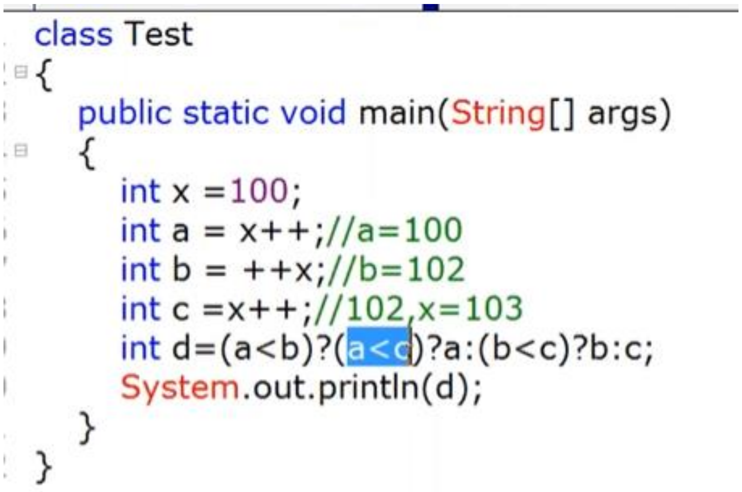
B) 5 + 2 = 7
   5 + 2 = 7

C) 5 + 2 = 7
   5 + 2 = 43

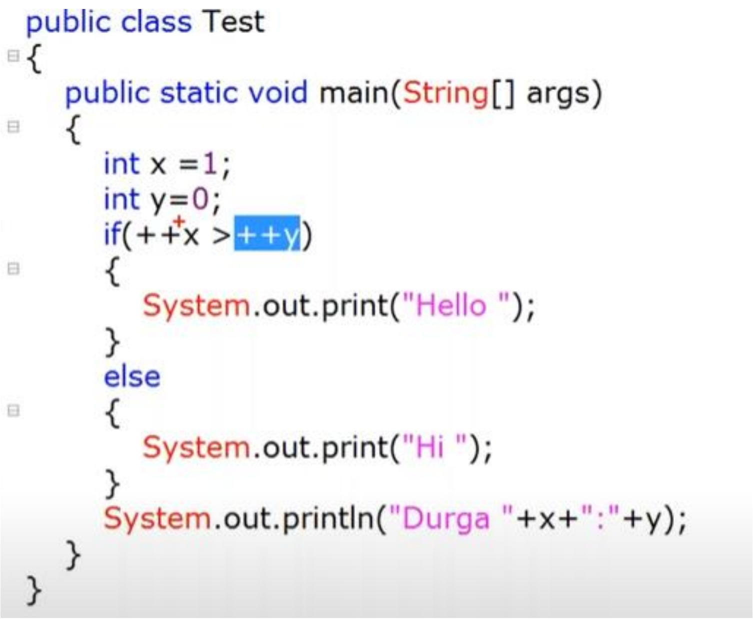
D) 5 + 2 = 43
   5 + 2 = 7



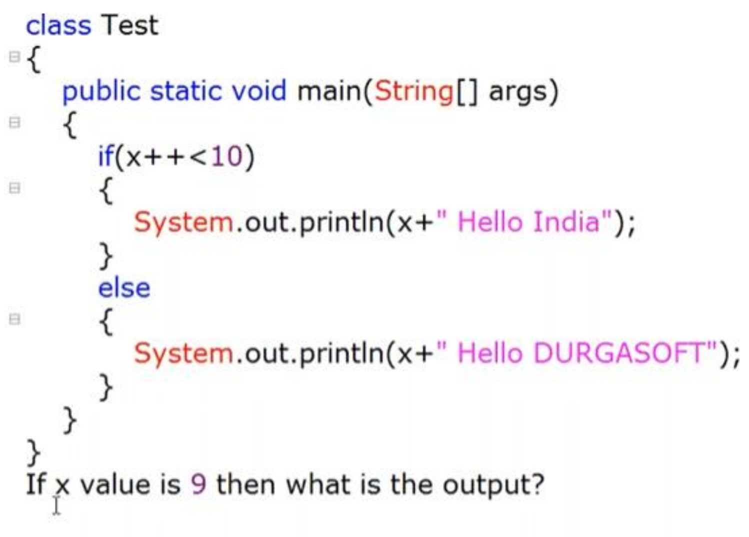
```
public class Test
{
    public static void main(String[] args)
    {
        System.out.println("Result A:"+ 4+5);//Result A:
        System.out.println("Result B:"+ (4)+(5));
    }
}
```



```
class Test
{
    public static void main(String[] args)
    {
        int x =100;
        int a = x++;//a=100
        int b = ++x;//b=102
        int c =x++;//102,x=103
        int d=(a<b)?(a<c)?a:(b<c)?b:c;
        System.out.println(d);
    }
}
```

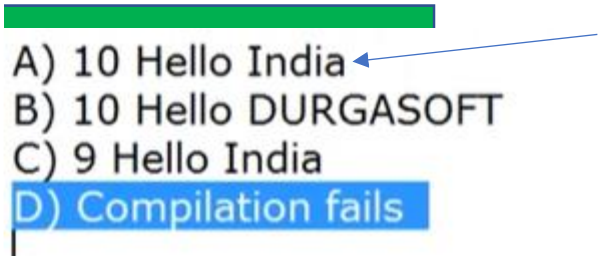


```
public class Test
{
    public static void main(String[] args)
    {
        int x =1;
        int y=0;
        if(++x >++y)
        {
            System.out.print("Hello ");
        }
        else
        {
            System.out.print("Hi ");
        }
        System.out.println("Durga "+x+":"+y);
    }
}
```



```
class Test
{
   public static void main(String[] args)
   {
      if(x++<10)
      {
         System.out.println(x+" Hello India");
      }
      else
      {
         System.out.println(x+" Hello DURGASOFT");
      }
   }
}
```

If x value is 9 then what is the output?

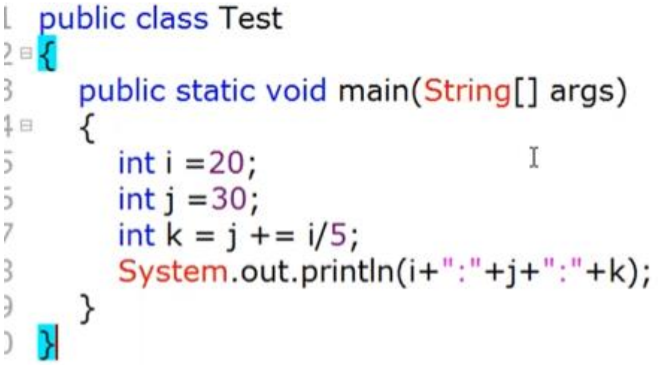


A) 10 Hello India

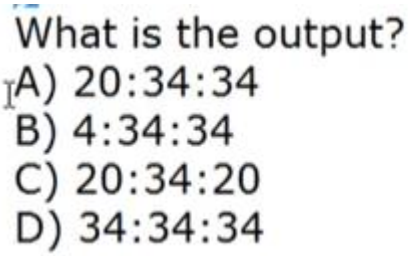
B) 10 Hello DURGASOFT

C) 9 Hello India

D) Compilation fails



```
public class Test
{
   public static void main(String[] args)
   {
      int i =20;
      int j =30;
      int k = j += i/5;
      System.out.println(i+":"+j+":"+k);
   }
}
```



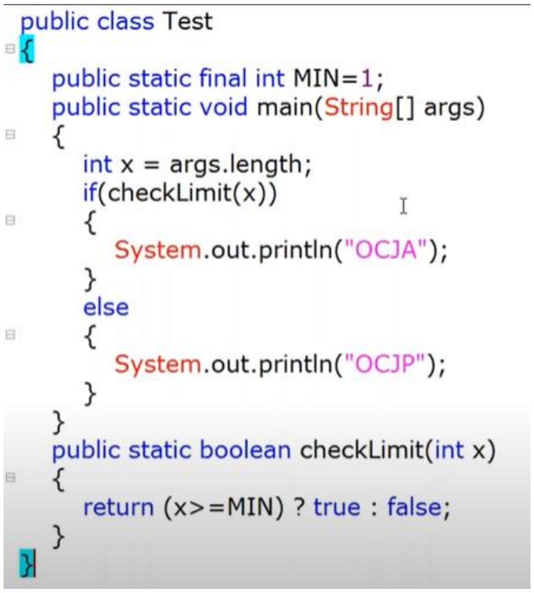
What is the output?

A) 20:34:34

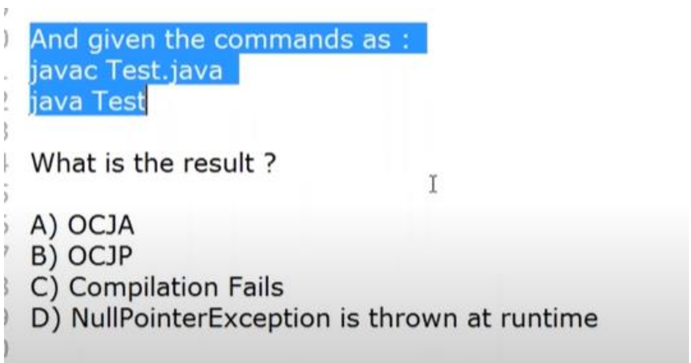
B) 4:34:34

C) 20:34:20

D) 34:34:34



```
public class Test
{
   public static final int MIN=1;
   public static void main(String[] args)
   {
      int x = args.length;
      if(checkLimit(x))
      {
         System.out.println("OCJA");
      }
      else
      {
         System.out.println("OCJP");
      }
   }
   public static boolean checkLimit(int x)
   {
      return (x>=MIN) ? true : false;
   }
}
```



And given the commands as :

```
javac Test.java
java Test
```

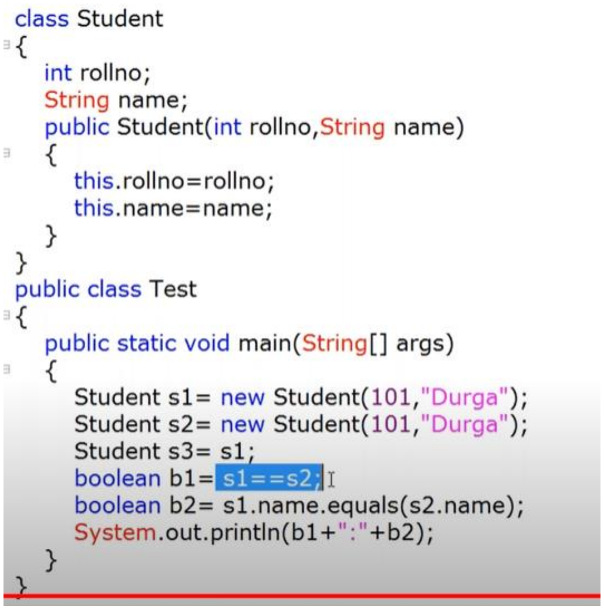
What is the result ?

A) OCJA

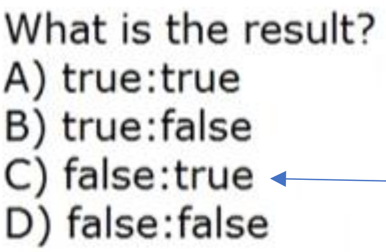
B) OCJP

C) Compilation Fails

D) NullPointerException is thrown at runtime



```
class Student
{
    int rollno;
    String name;
    public Student(int rollno,String name)
    {
        this.rollno=rollno;
        this.name=name;
    }
}
public class Test
{
    public static void main(String[] args)
    {
        Student s1= new Student(101,"Durga");
        Student s2= new Student(101,"Durga");
        Student s3= s1;
        boolean b1= s1==s2;
        boolean b2= s1.name.equals(s2.name);
        System.out.println(b1+":"+b2);
    }
}
```



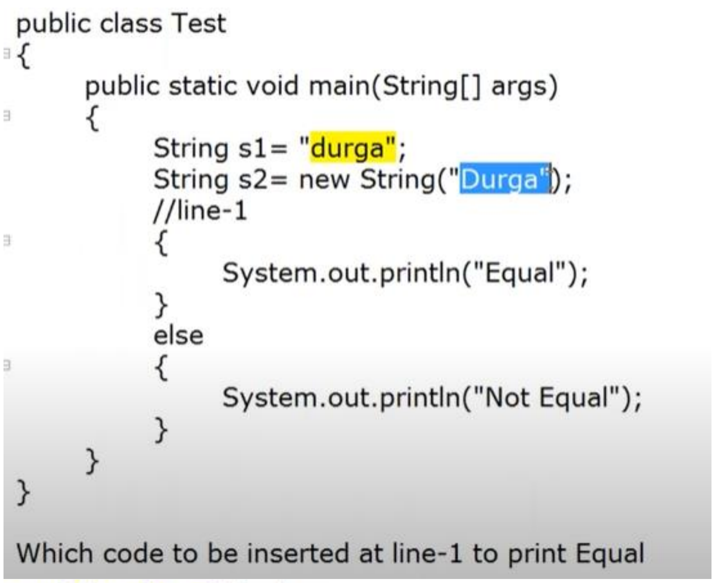
What is the result?

A) true:true

B) true:false

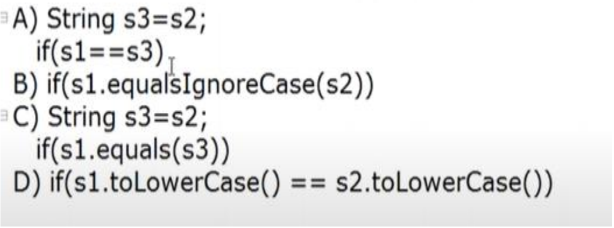
C) false:true

D) false:false

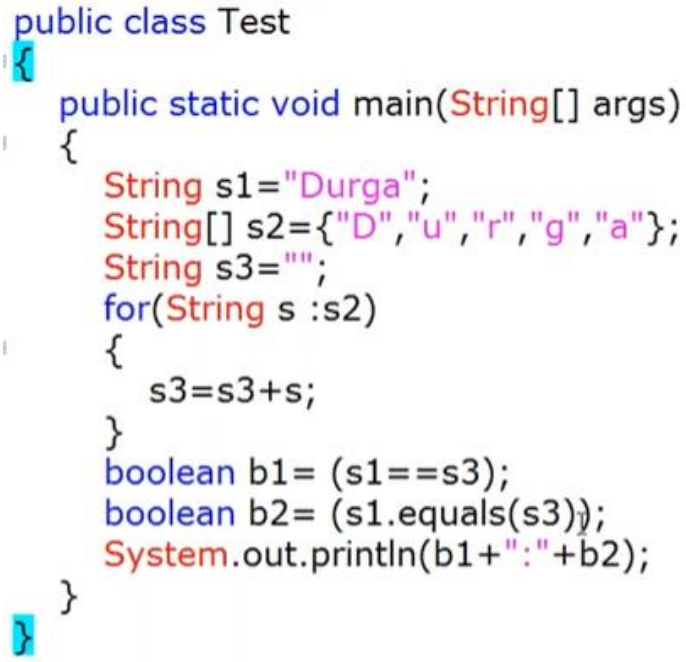


```
public class Test
{
        public static void main(String[] args)
        {
                String s1= "durga";
                String s2= new String("Durga");
                //line-1
                {
                        System.out.println("Equal");
                }
                else
                {
                        System.out.println("Not Equal");
                }
        }
}
```

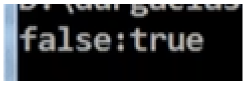
Which code to be inserted at line-1 to print Equal



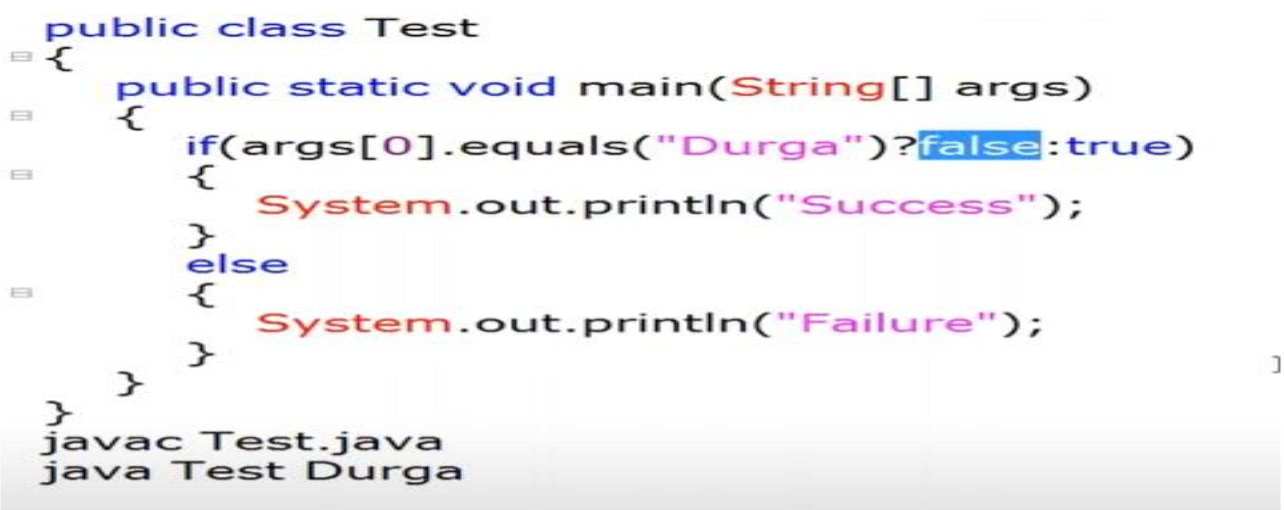
```
A) String s3=s2;
   if(s1==s3)
B) if(s1.equalsIgnoreCase(s2))
C) String s3=s2;
   if(s1.equals(s3))
D) if(s1.toLowerCase() == s2.toLowerCase())
```



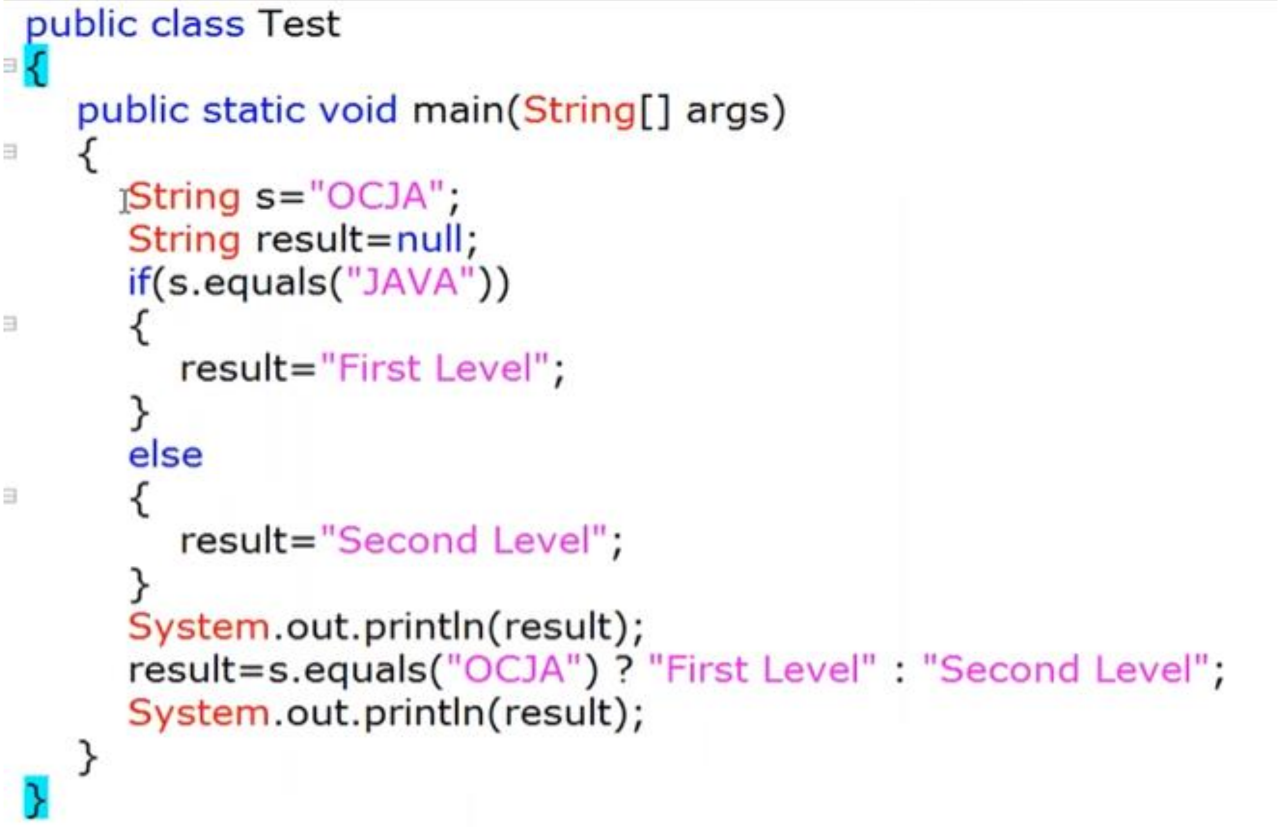
```
public class Test
{
    public static void main(String[] args)
    {
        String s1="Durga";
        String[] s2={"D","u","r","g","a"};
        String s3="";
        for(String s :s2)
        {
            s3=s3+s;
        }
        boolean b1= (s1==s3);
        boolean b2= (s1.equals(s3));
        System.out.println(b1+":"+b2);
    }
}
```



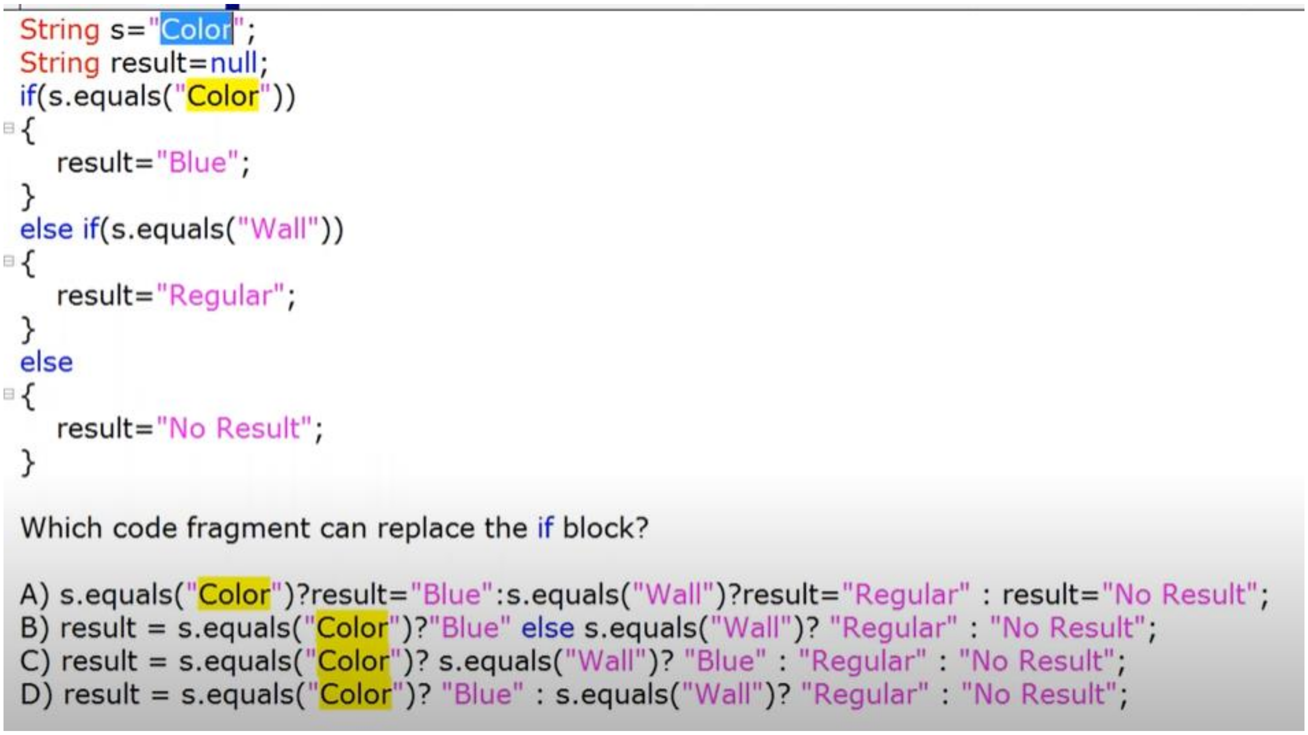
```
false:true
```



```
public class Test
{
   public static void main(String[] args)
   {
      if(args[0].equals("Durga")?false:true)
      {
         System.out.println("Success");
      }
      else
      {
         System.out.println("Failure");
      }
   }
}
javac Test.java
java Test Durga
```



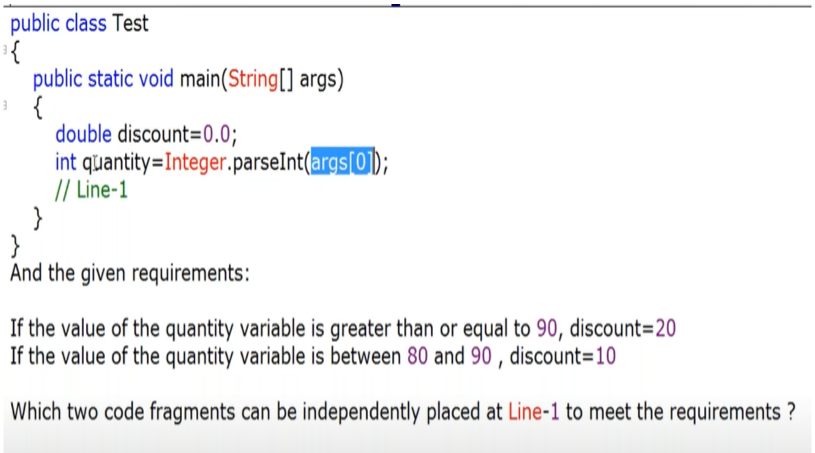
```
public class Test
{
   public static void main(String[] args)
   {
      String s="OCJA";
      String result=null;
      if(s.equals("JAVA"))
      {
         result="First Level";
      }
      else
      {
         result="Second Level";
      }
      System.out.println(result);
      result=s.equals("OCJA") ? "First Level" : "Second Level";
      System.out.println(result);
   }
}
```



```
String s="Color";
String result=null;
if(s.equals("Color"))
{
   result="Blue";
}
else if(s.equals("Wall"))
{
   result="Regular";
}
else
{
   result="No Result";
}
```

Which code fragment can replace the if block?

A) s.equals("Color")?result="Blue":s.equals("Wall")?result="Regular" : result="No Result";
B) result = s.equals("Color")?"Blue" else s.equals("Wall")? "Regular" : "No Result";
C) result = s.equals("Color")? s.equals("Wall")? "Blue" : "Regular" : "No Result";
D) result = s.equals("Color")? "Blue" : s.equals("Wall")? "Regular" : "No Result";

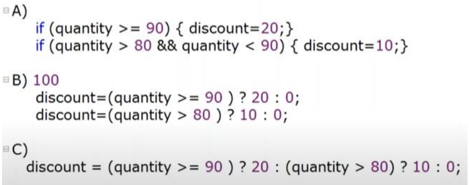


```
public class Test
{
  public static void main(String[] args)
  {
    double discount=0.0;
    int quantity=Integer.parseInt(args[0]);
    // Line-1
  }
}
```

And the given requirements:

If the value of the quantity variable is greater than or equal to 90, discount=20
If the value of the quantity variable is between 80 and 90 , discount=10

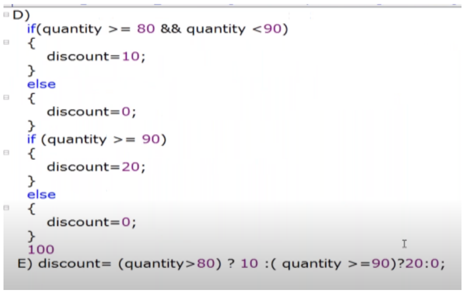
Which two code fragments can be independently placed at Line-1 to meet the requirements ?



A)
```
    if (quantity >= 90) { discount=20;}
    if (quantity > 80 && quantity < 90) { discount=10;}
```

B) 100
```
    discount=(quantity >= 90 ) ? 20 : 0;
    discount=(quantity > 80 ) ? 10 : 0;
```

C)
```
  discount = (quantity >= 90 ) ? 20 : (quantity > 80) ? 10 : 0;
```



D)
```
  if(quantity >= 80 && quantity <90)
  {
    discount=10;
  }
  else
  {
    discount=0;
  }
  if (quantity >= 90)
  {
    discount=20;
  }
  else
  {
    discount=0;
  }
  100
```
E) discount= (quantity>80) ? 10 :(quantity >=90)?20:0;