

Primitives call by value , instance – call by ref

The screenshot shows a Java code editor window with the following code:

```
9 int x=10;
10 m1(x);
11 SOP(x)
12
13 reference variable==>
14
15
16 Test t1 = new Test();
17 SOP(t1.x); 10
18 m1(t1);
19 SOP(t1.x);
20
21
22 m1(Test t)
23 {
24     t.x=777;
25 }
```

The code demonstrates passing primitives by value. Lines 16-19 show the value of `t1.x` being passed to the `m1` method, which prints the original value of 10. Lines 23-25 show changing the value of `t.x` to 777, which does not affect the original variable `x`.

Passing primitives.

The screenshot shows a Java code editor window with the following code:

```
1 public class Test
2 {
3     public void m1(int i, int j)
4     {
5         i=i+10;
6         j=j+10;
7         System.out.println("Inside Method:"+i+".."+j);//110..210
8     }
9     public static void main(String[] args)
10    {
11        int x=100;
12        int y =200;
13        Test t = new Test();
14        t.m1(x,y);
15        System.out.println("After Completing Method:"+x+".."+y);//100..200
16    }
17 }
18
```

This code demonstrates that primitive variables are passed by value. The `m1` method increments both `i` and `j` by 10, but these changes do not affect the original variables `x` and `y` in the `main` method. The output shows the original values of 100 and 200.

Local variable i and j is not impacted by instance methods and const.

3.. 5

```
1 public class Test
2 {
3     int i,j;
4     public Test(int i,int j)
5     {
6         initialize(i,j);
7     }
8     public void initialize(int i,int j)
9     {
10        this.i = i*i;
11        this.j=j*j;
12    }
13    public static void main(String[] args)
14    {
15        int i =3, j= 5;
16        Test t = new Test(i,j);
17        System.out.println(i+"..."+j);
18    }
19 }
```

The output window shows the result of the program execution:

```
i...j
```

```
1 public class Test
2 {
3     int i,j;
4     public Test(int i,int j)
5     {
6         initialize(i,j);
7     }
8     public void initialize(int i,int j)
9     {
10        this.i = i*i; 9
11        this.j=j*j; 25
12    }
13    public static void main(String[] args)
14    {
15        int i =3, j= 5;
16        Test t = new Test(i,j);
17        System.out.println(i+"..."+j);
18    }
19 }
```

The right side of the screen shows a 'Questions' panel with a list of user questions and answers:

X	Question	Asker
cons	dinesh durgasoft	
const	Pooja Chavan	
constructor	Gowindu Rayapur	
constructor	SANDEEP REDDY	
constructor	john acharya	
3 5 is ans	Shubham Gupta	
3..5	Shubham Gupta	
3.5 is ans	Shubham Gupta	
9 25	Rajat Gupta	
3..5	Pooja Chavan	
35	dinesh durgasoft	

```

1 public class Test
2 {
3     int i,j;
4     public Test(int i,int j)
5     {
6         initialize(i,j);
7     }
8     public void initialize(int i,int j)
9     {
10        this.i = i*i;
11        this.j=j*j;
12    }
13    public static void main(String[] args)
14    {
15        int i =3, j= 5;
16        Test t = new Test(i,j);
17        System.out.println(i+"..."+j);
18        System.out.println(t.i+"..."+t.j);
19    }
20 }

```

Test.java Untitled4 TypesOfVariables_44
For Help, press F1 In 18 col 40 20 6A PC ANSI

```

11
12
13
14

```

this D:\durgaclasses>java Test
3...5
9...25

```

1 class Demo
2 {
3     int x;
4     int y;
5 };
6 public class Test
7 {
8     public void m1(Demo d)
9     {
10        d.x=888;
11        d.y=999;
12    }
13    public static void main(String[] args)
14    {
15        Demo d1 = new Demo();
16        d1.x=10;
17        d1.y=20;
18        Test t = new Test();
19        t.m1(d1);
20        System.out.println(d1.x+"..."+d1.y);
21    }
22 }

```

Test.java Untitled4 TypesOfVariables_44
For Help, press F1 In 19 col 16 22 29 PC ANSI

Questions

X	Question	Asker
no sir	dinesh durgasoft	Poja Chavan
no		Gevindu Rayapur
clear sir		john acharya
nos ir		Shubham Gupta
888..999 is ans		SANDEEP REDDY
888..999		Rajat Gupta
888 999		RAMESWARA RED...
d1		Poja Chavan
d1		dinesh durgasoft
d1		john acharya

Is session started?

Send Privately Send to All

OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir Demo On 27 - 01 - 2018

```

1 class Product
2 {
3     double price;
4 }
5 public class Test
6 {
7     public void updatePrice(Product p, double price)
8     {
9         price = price * 2; 200
10        p.price = p.price + price;
11    }
12    public static void main(String[] args)
13    {
14        Product prt = new Product();
15        prt.price = 200;
16        double newPrice = 100;
17
18        Test t = new Test();
19        t.updatePrice(prt, newPrice);
20        System.out.println(prt.price + "...." + newPrice);
21    }
22 }

```

Handwritten annotations:

- Line 9: $price = price * 2; \boxed{200}$
- Line 10: $p.price = p.price + price;$
- Line 11: $(p.price + 200)$ (in a red oval)
- Line 16: $\boxed{newPrice = 100;}$
- Line 20: $400 \dots 100$ (in red)

Questions window:

X	Question	Asker
if t.m1(d)	dinesh durgasoft	
yes sir	dinesh durgasoft	
s	Poja Chavan	
s	Poja Chavan	
yes sir	Rajat Gupta	
yes sir	dinesh durgasoft	
yes sir	john acharya	
ye sr	Deepakaj Yadav	
400_100	Shubham Gupta	
400_100	Rajat Gupta	
400_100	Shubham Gupta	

Ques 508 What is the result?
 A) 200.0....100.0
 B) 400.0....400.0
 C) 400.0....100.0
 D) Compilation Fails

Int y in doStuff are local variables updating local variable y will not impact t1.y

OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir Demo On 27 - 01 - 2018

```

1 public class Test
2 {
3     int x; int y;
4     public void doStuff(int x, int y)
5     {
6         this.x=x;
7         y=this.y;
8     }
9     public void print()
10    {
11        System.out.print(x+":"+y+":");
12    }
13    public static void main(String[] args)
14    {
15        Test t1=new Test();
16        t1.x=100;
17        t1.y=200;
18
19        Test t2 = new Test();
20        t2.doStuff(t1.x,t1.y);
21        t1.print();
22        t2.print();
23    }

```

Handwritten annotations:

- Line 6: $t1.x = 100$
- Line 7: $y = 200$
- Line 17: $x = 100$
- Line 18: $y = 200$
- Line 20: $t2.doStuff(t1.x, t1.y);$
- Line 21: $t1.print();$
- Line 22: $t2.print();$
- Red wavy line: *Print Only*

Questions window:

X	Question	Asker
www www	john acharya	
wait sir	SANDEEP REDDY	
100_0	dinesh durgasoft	
y=200	dinesh durgasoft	
x=0	SANDEEP REDDY	
i think am wrong	dinesh durgasoft	
100_200_100_0	RAMESWARA RED...	
100_200_100_0	Deepakaj Yadav	
100_200_100_200	Govindu Rayapur	
100_200	john acharya	

```

9 public D:\durgaclasses>java Test
10 {
11     100:200:100:0:

```

OCJA 1.8 Java SE 8 Programmer - I (1Z0-808) By Durga Sir Demo On 27-01-2018

```
1 public class Vowel
2 {
3     private char ch;
4     public static void main(String[] args)
5     {
6         char ch1='a';
7         char ch2=ch1;
8         ch2='e';//  

9  

10    Vowel obj1= new Vowel();
11    Vowel obj2=obj1;
12    obj1.ch='i';
13    obj2.ch='o';
14  

15    System.out.println(ch1+":"+ch2);
16    System.out.println(obj1.ch+":"+obj2.ch);
17 }
18 }
```

Handwritten annotations:

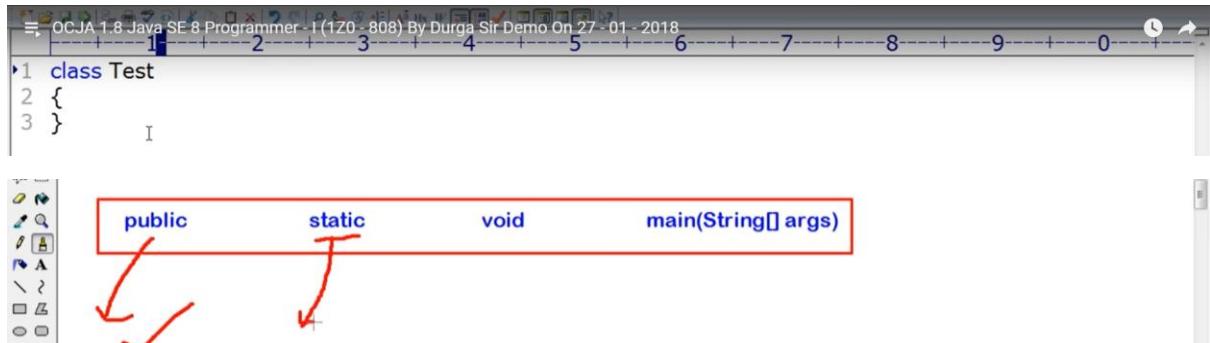
- A red circle highlights the line `char ch1='a';`. An arrow points from this circle to another circle containing the text `ch = x o`.
- Below the first `System.out.println` statement, there are three circles: one under `ch1`, one under `ch2`, and one under the plus sign.
- Below the second `System.out.println` statement, there are three circles: one under `obj1.ch`, one under `obj2.ch`, and one under the plus sign.

Questions pane:

X	Question	Asker
a a i o	Poja Chavan	
a e i o	Rich Cohen	
a e i o u	Govindu Rayapur	
e	dinesh durgasoft	
a a	SANDEEP REDDY	
i o	SANDEEP REDDY	
a e , i o	RAMESWARA RED...	
e a	dinesh durgasoft	
a e i o	john acharya	
a e i o	Poja Chavan	
a e and i o	SANDEEP REDDY	

Send buttons: Send Privately, Send to All

Will the class compile – yes . The class will compile even if we don't hv a main method. At run time JVM will throw error , Main method not available



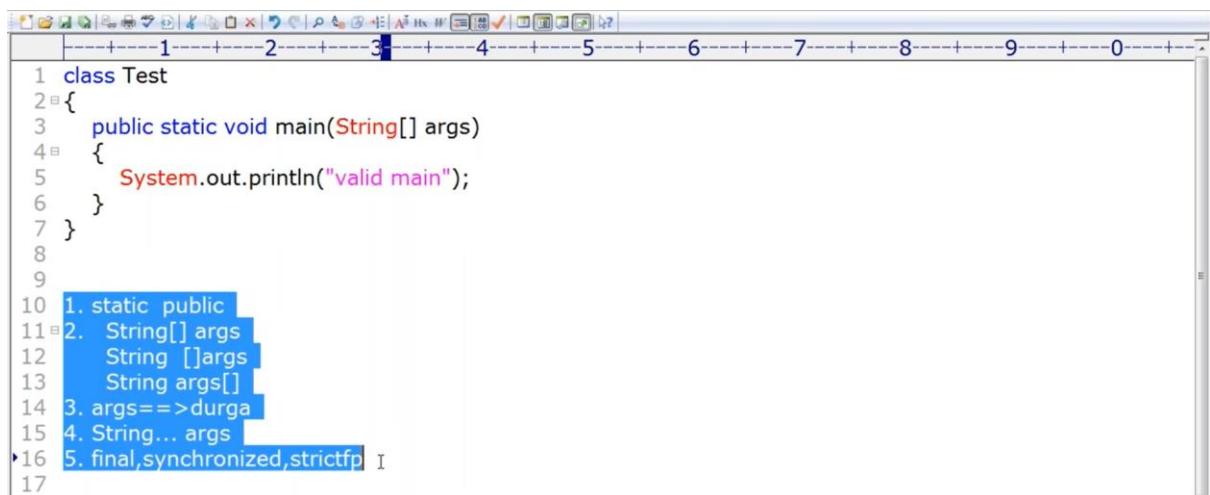
```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir Demo On 27 - 01 - 2018
1 class Test
2 {
3 }
```

Since main method should be accessed by JVM from any where main method is public

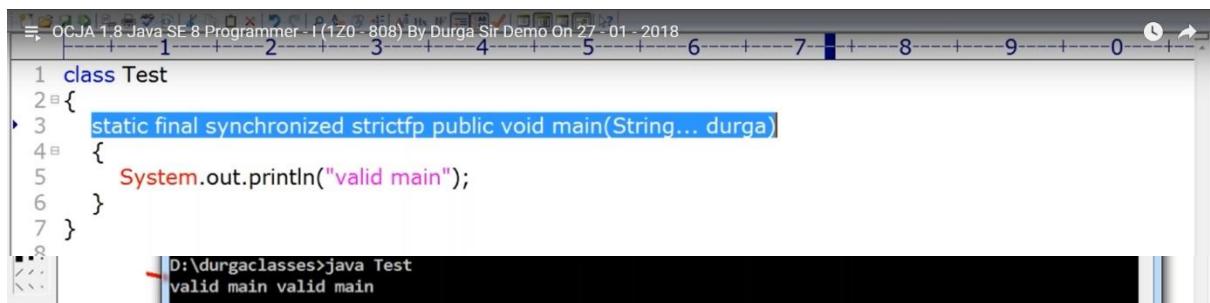
JVM should be accessed by JVM without creating any object , so its static. When its static it needs not object creation to access it

Since it's called by JVM and returning a value to JVM has no value Hence its void

Main method can be final , synchronized and Strictfp



```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir Demo On 27 - 01 - 2018
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("valid main");
6     }
7 }
8
9
10 1. static public
11 2. String[] args
12 3. String []args
13 4. String args[]
14 5. final,synchronized,strictfp
15
16
17
```

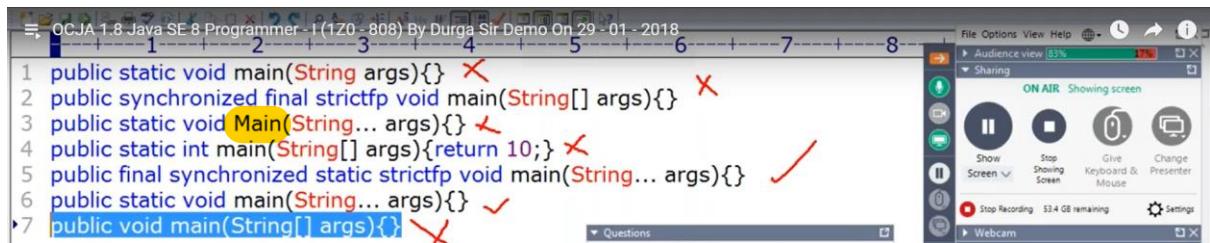


```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir Demo On 27 - 01 - 2018
1 class Test
2 {
3     static final synchronized strictfp public void main(String... durga)
4     {
5         System.out.println("valid main");
6     }
7 }
8
9
```

```
D:\durgaclasses>java Test
valid main valid main
```

Main – should be lower case

String... args – an array of string



The screenshot shows a Java IDE interface. In the code editor, there are seven lines of Java code:

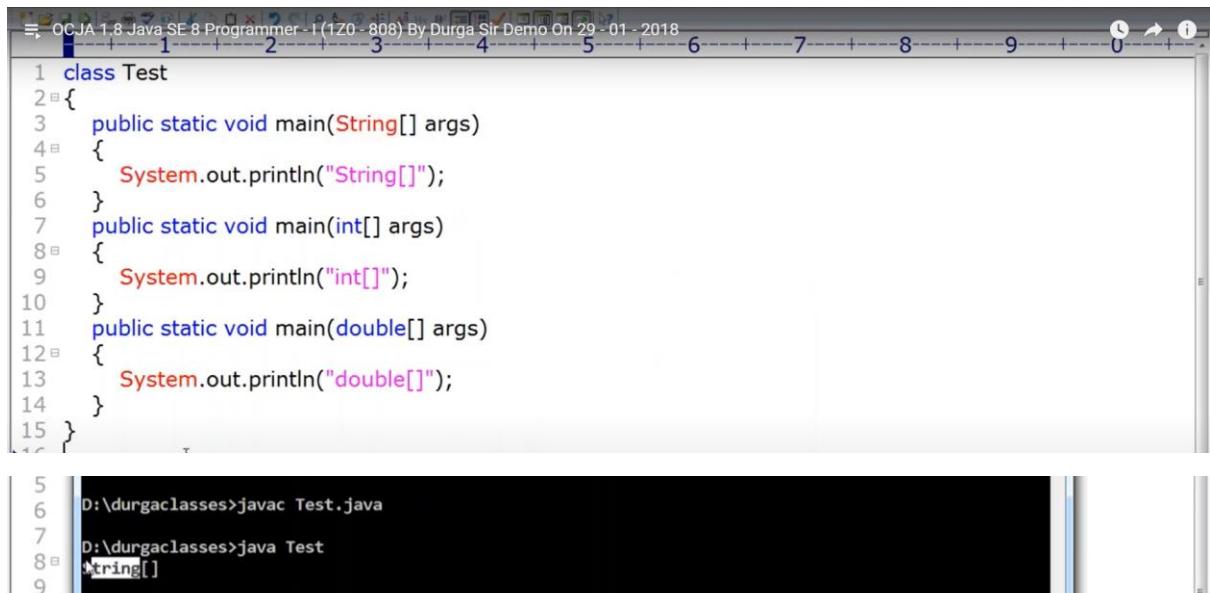
```
1 public static void main(String args){} ✗
2 public synchronized final strictfp void main(String[] args){} ✗
3 public static void Main(String... args){} ✗
4 public static int main(String[] args){return 10;} ✗
5 public final synchronized static strictfp void main(String... args){} ✓
6 public static void main(String... args){} ✓
7 public void main(String[] args){} ✗
```

A yellow circle highlights the third line, "public static void Main(String... args){}". A red checkmark is placed next to the fifth line, "public final synchronized static strictfp void main(String... args){}". A red X is placed next to the first, second, fourth, and seventh lines. The control panel on the right shows "ON AIR Showing screen" and various sharing options.

The above will not throw compile time error. Cos compiled has no knowledge of main method.

These will be treated as normal methods

Overloading main method compilation works fine. JVM will not throw an error but JVM will look for the right main method



The screenshot shows a Java IDE interface with a code editor and a terminal window below it.

Code Editor Content:

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("String[]");
6     }
7     public static void main(int[] args)
8     {
9         System.out.println("int[]");
10    }
11    public static void main(double[] args)
12    {
13        System.out.println("double[]");
14    }
15 }
```

Terminal Window Content:

```
5 D:\durgaclasses>javac Test.java
6
7 D:\durgaclasses>java Test
8 String[]
```

Compile time error methods having same signature and arguments.

The screenshot shows a Java code editor with the following code:

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("String[]");
6     }
7     public static void main(String... args)
8     {
9         System.out.println("String...");
```

A red box highlights the second `main` method signature, and a red arrow points from it to the text "String[] args". A floating window titled "Questions" lists various Java-related topics and their askers.

No compile or run time , line 8 is two dimensional array – var args

The screenshot shows a Java code editor with the same code as above, but the second `main` method signature is underlined with a yellow squiggly line, indicating a warning or error.

Inheritance is acceptable for all static class including main method

The screenshot shows a Java code editor with the following code:

```
1 class P
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Parent Main");
6     }
7 }
8 class C extends P
9 {
10 }
11
12 java P
13 java C
```

This demonstrates that static methods, including the `main` method, can be inherited by subclasses.

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 29 - 01 - 2018
String[]

D:\durgaclasses>javac P.java

D:\durgaclasses>java P
Parent Main

D:\durgaclasses>java C
Parent Main

D:\durgaclasses>_
```

The below will not throw an error. Overriding for main method is accepted. So the below will work fine. But the below is not overriding this is method hiding we will see. But the below will work fine

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 29 - 01 - 2018
1 class P
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Parent Main");
6     }
7 }
8 class C extends P
9 {
10    public static void main(String[] args)
11    {
12        System.out.println("Child Main");
13    }
14 }
```

```
D:\durgaclasses>javac P.java
D:\durgaclasses>java P
Parent Main
D:\durgaclasses>java C
Child Main
D:\durgaclasses>_
```

After 1.7 only the below error

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 29 - 01 - 2018
1 class Test
2 {
3
4 }
5
6 javac Test.java
7 java Test
8 Error: Main method not defined in class Test, please define main method as
9 public static void main(String[] args)
```

Before – No Such method found error

```
C:\Windows\system32\cmd.exe
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 29 - 01 - 2018
Error: Main method not found in class Test, please define the main method as:
1 class
2 {
3     public static void main(String[] args)
4     or a JavaFX application class must extend javafx.application.Application
5
6 D:\durgaclasses>javac -version
javac 1.8.0_31
6 D:\durgaclasses>javac -source 1.6
```

Until 1.6- Static block will execute first then main method

```
C:\Windows\system32\cmd.exe
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 29 - 01 - 2018
1 class Test
2 {
3     static
4     {
5         System.out.println("static block");
6     }
7
8 }
9 first static block will execute and then jvm will search for main ==>1.6
10
```

From 1.7 it would check if main method is there or not, will throw an error if main method is not available. If available, it executes the static block first and then the main method. Whereas before 1.7 until 1.6 JVM executes the static block and throws a run time error saying No such Method found

```
C:\Windows\system32\cmd.exe
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 29 - 01 - 2018
1 class Test
2 {
3     static
4     {
5         System.out.println("static block");
6     }
7     public static void main(String[] args)
8     {
9         System.out.println("main method");
10    }
11
12 }
```

S

1.7 Version Enhancements with respect to main():

Case 1:

Until 1.6v if our class doesn't contain main() method then at runtime we will get Runtime Exception saying NoSuchMethodError:main

But from 1.7 version onwards instead of NoSuchMethodError we will get more meaningful description

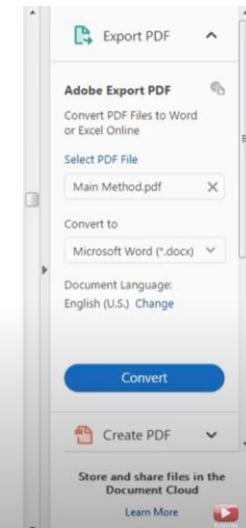
```
I  
class Test {  
}
```

1.6 version:

```
javac Test.java  
java Test  
RE: NoSuchMethodError:main
```

1.7 version:

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



Error: main method not found in class Test, please define the main method as
public static void main(String[] args)

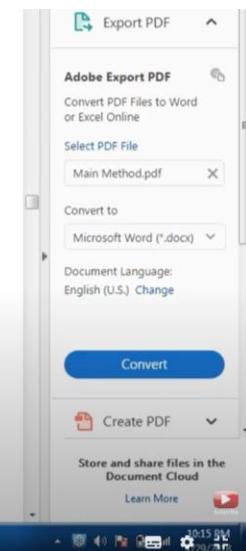
Case 2:

From 1.7 version onwards to start program execution compulsory main method should be required, hence even though the class contains static block if main method not available then won't be executed

```
1) class Test {  
2) static {  
3) System.out.println("static block");  
4) }  
5) }
```

1.6 version:

```
javac Test.java  
java Test  
output:  
static block  
RE: NoSuchMethodError:main
```



1.7 version:

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

Error: main method not found in class Test, please define the main method as
public static void main(String[] args)



Main Method.pdf 12 pages Acrobat Reader DC
File Edit View Insert Tools Help OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 29 - 01 - 2018

Home Tools PYTHON_DURGA.p... Main Method.pdf 12 Sign In

Case 3:

```
1) class Test {  
2) static {  
3) System.out.println("static block");  
4) System.exit(0);  
5) }  
6} 
```

1.6 version:

```
javac Test.java  
java Test  
output :  
static block
```

1.7 version:

```
javac Test.java  
java Test  
Error: main method not found in class Test, please define the main method as  
public static void main(String[] args)
```

Case 4:

This screenshot shows a Java code editor window within Adobe Acrobat Reader DC. The left pane displays two code snippets under 'Case 3' and 'Case 4'. The right pane features a sidebar with conversion tools for PDF files, including options to export to Microsoft Word or create a new PDF. The system tray at the bottom shows the date and time as 38:48 / 1:03:38.

Case 4:

```
1) class Test {  
2) static {  
3) System.out.println("static block");  
4) }  
5) public static void main(String[] args) {  
6) System.out.println("main method");  
7) }  
8} 
```

1.6 version:

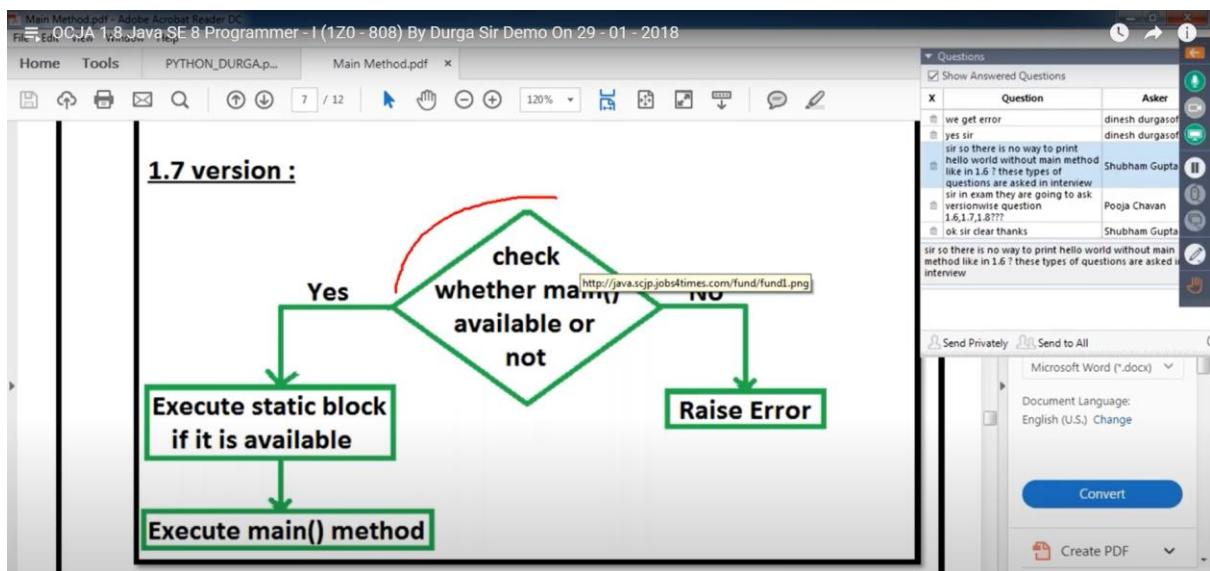
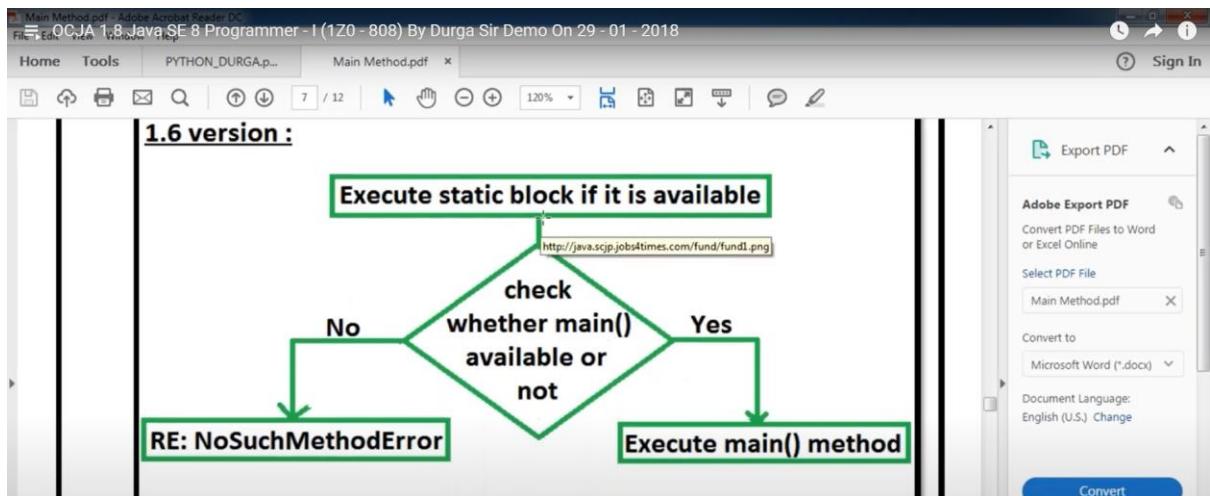
```
javac Test.java  
java Test  
output :  
static block  
main method
```

This screenshot shows a Java code editor window within Adobe Acrobat Reader DC. The left pane displays two code snippets under 'Case 4'. The right pane features a sidebar with conversion tools for PDF files, including options to export to Microsoft Word or create a new PDF. The system tray at the bottom shows the date and time as 38:48 / 1:03:38.

1.7 version:

```
javac Test.java  
java Test  
output :  
static block  
main method
```

This screenshot shows a Java code editor window within Adobe Acrobat Reader DC. The left pane displays two code snippets under '1.7 version'. The right pane features a sidebar with conversion tools for PDF files, including options to export to Microsoft Word or create a new PDF. The system tray at the bottom shows the date and time as 38:48 / 1:03:38.



Command line Args

OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 29 - 01 - 2018

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5     }
6 }
7
8 java Test A B C
9

```

A B C → String[] JVM

Questions

- SHRI EXAM URGY ARE GOING TO ASK VERSIONWISE QUESTION 1.6,1.7,1.8???
- ohhk Pooja Chavan
- yes Shubham Gupta
- s Pooja Chavan
- s Shubham Gupta
- yes Shubham Gupta
- runtime Sai chand Kilari

sir so there is no way to print hello world without main method like in 1.6 ? these types of questions are asked in interview

Send Privately Send to All

```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir Demo On 29 - 01 - 2018  
1 class Test  
2 {  
3     public static void main(String[] args)  
4     {  
5         int x =Integer.parseInt(args[0]);  
6         int y =Integer.parseInt(args[1]);  
7         System.out.println(x+y);  
8     }  
9 }
```

C. is the correct answer

```
OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir Demo On 29 - 01 - 2018  
Press Esc to exit full screen  
1 public static void main(String[] args)  
2 {  
3     System.out.println("Hello "+args[0]);  
4 }  
5 Which set of commands prints Hello Durga in the console?  
6  
A)  
javac Test  
java Test Durga  
7  
B)  
javac Test.java Durga  
java Test  
8  
C)  
javac Test.java  
java Test Durga  
9  
D)  
javac Test.java  
java Test.class Durga
```

OCJA 1.8 Java SE 8 Programmer - I (1Z0-808) By Durga Sir Demo On 29-01-2018

```
67 Q. Consider the code Test.java:  
68  
69 public class Test  
70 {  
71     public static void main(int[] args)  
72     {  
73         System.out.println("int[] main: "+args[0]);  
74     }  
75     public static void main(Object[] args)  
76     {  
77         System.out.println("Object[] main: "+args[0]);  
78     }  
79     public static void main(String[] args)  
80     {  
81         System.out.println("String[] main: "+args[0]);  
82     }  
83 }  
84 and the commands  
85 javac Test.java  
86 java Test 1 2 3  
87  
88 What is the result?  
89 A) int[] main 1  
90 B) Object[] main 1  
91 C) String[] main 1  
92 D) Compilation Fails  
93 E) An Exception raises at runtime
```

Next Topic very imp

Untitled2 * - EditPlus

```
1 operators and assignments:  
2 -----  
3 1Z0-808,  
4 ternary operator==> ?:  
5 == operator and .equals() method  
6  
I
```

PreIncrement and Post increment. Pre Decrement and Post Dec

Y =++x; First increments and assign x to you.

Y=x++; first assigns x to y and then increments x

The screenshot shows a code editor window titled "Untitled2 * - EditPlus". The code is as follows:

```
9 int x = 10;
10 int y = ++x;
11 sop(x); // Line 11 has two cursor positions
12 sop(y); // Line 12 has two cursor positions
13 -----
14 x=10;
15 y=x++;
16 sop(x); // Line 16 has two cursor positions
17 sop(y); // Line 17 has two cursor positions
18 -----
19 x=10;
20 y=--x;
21 sop(x); // Line 21 has two cursor positions
22 sop(y); // Line 22 has two cursor positions
23 -----
24 x=10;
25 y=x--;
26 sop(x); // Line 26 has two cursor positions
27 sop(y); // Line 27 has two cursor positions
28
```

Incremet and dec are not applicable for values- Unexpected type ++10. Required variable. Applicable only for available

The screenshot shows a code editor window titled "OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir Demo On 30 - 01 - 2018". The code is as follows:

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int x = 10;
6         int y = ++10;
7         System.out.println(y);
8     }
9 }
10
```

Compile time error. ++x is 11 . it's a value not variable applying ++11 throws error . above

The screenshot shows a code editor window titled "OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir Demo On 30 - 01 - 2018". The code is as follows:

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int x = 10;
6         int y = ++(++x);
7         System.out.println(y);
8     }
9 }
10
```

In Increment and decrement operation reassign is gonna to happen. So it reassigned , final variable cant be reassigned.

The screenshot shows an IDE interface with a code editor and a terminal window. The code in the editor is:

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         final int x = 10;
6         x++; // x=x+1
7         System.out.println(x);
8     }
9 }
```

The terminal window shows the output of the compilation command:

```
D:\durgaclasses>javac Test.java
Test.java:6: error: cannot assign a value to final variable x
        x++; // x=x+1
               ^
1 error
```

Increment and dec is applicable for all primitives except boolean

The screenshot shows an IDE interface with a code editor. The code is:

```
1 char ch='a';
2 ch++;
3 sop(ch)b
4 -----
5 double d =10.5;
6 d++;
7 sop(d);11.5
8
```

Increment and dec , implicit typecasting will be performed and the value will always be in the range only

The screenshot shows an IDE interface with a code editor. The code is:

```
-- 
11 byte x=127;
12 x++; ==>x=(byte)x+1;
13 sop(b);
14
```

The terminal window shows the output of the compilation command:

```
D:\durgaclasses>javac Test.java
D:\durgaclasses>java Test
-128
```

The below will never end, infinite loop

The screenshot shows an IDE interface with a code editor. The code is:

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         byte b =0;
6         while(b++ <128)
7         {
8             System.out.println(b);
9         }
10    }
11 }
12
```

Arithmetic operator

```
9 except boolean
10
11
12 byte x=127;
13 x++ ;==>x=(byte)(x+1);
14 sop(b);
15 -----
16 +,-,*,/ and %
17 |
```

Systematically the below is correct. Any arithmetic

```
20
21 sop(10/0);
22 sop(10/0.0);
23 sop(0/0);
24 sop(0/0.0);
```

Result type for any arithmetic operation the result type is

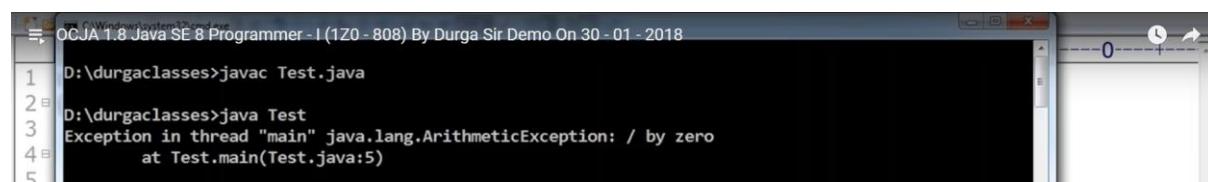
=Max(int, type(a),type(b))

1. Max(int,int,int) → int
2. Maz(int, int, double) -> double
3. Max(int,int,int) → int
4. Maz(int, int, double) -> double

In integral rep there is no way to represent infinity so in 1 case , which is int when we divide a no by 0 , which is infinity throws arithmetic exception. Whereas in case 4 double has a way to rep infinity so it will not throw arithmetic exception



```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 30 - 01 - 2018
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         System.out.println(10/0);
6     }
7 }
```



```
D:\durgaclasses>javac Test.java
D:\durgaclasses>java Test
Exception in thread "main" java.lang.ArithmaticException: / by zero
        at Test.main(Test.java:5)
```

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5         System.out.println(10/0.0);
6     }
7 }

D:\durgaclasses>java Test
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Test.main(Test.java:5)

D:\durgaclasses>javac Test.java

D:\durgaclasses>java Test
Infinity

```

In general math $0/0$ is undefined. So integral rep there is no way to define undefined values. So it throws Arithmetic Exception. Case 2

Whereas Floating point rep has a way to rep undefined values which is NaN case 4

```

17
18 a and b
19 result type=max(int,type(a),type(b))
20
21
22
23 sop(10/0); //int==>AE
24 sop(10/0.0); //double==>Infinity
25 sop(0/0); //int==>AE
26 sop(0/0.0); //double==>NaN
27

```

/ and % operation alone will cause arithmetic exception. Run Time exception AE

```

29
30 => ArithmeticException==>
31     possible only in Integral arithmetic but not in floating point arithmetic
32     /, %
33     RuntimeException but not ce
34

```

+ acts as arithmetic operator and as concatenation in string. + operator alone supports operator overloading. Since + operator supports operator overloading we cant say java as a whole supports operator overloading

```

34
35 String concatenation operator(+):
36 -----
37 10+20==>30
38 "ab"+"cd"==>abcd
39 I
40

```

String + no – String , String+no -> String

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 30 - 01 - 2018  
Press Esc to exit full screen  
1 class Test  
2 {  
3     public static void main(String[] args)  
4     {  
5         String a="durga";  
6         int b=10;  
7         int c =20;  
8         int d=30;  
9         System.out.println(a+b+c+d); //durga102030  
10        System.out.println(b+a+c+d); //10durga2030  
11        System.out.println(b+c+a+d); //30durga30  
12        System.out.println(b+c+d+a); //60durga  
13        System.out.println(a+b+(c+d)); //durga1050  
14    }  
15 }  
16
```

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 30 - 01 - 2018  
Press Esc to exit full screen  
1 class Test  
2 {  
3     public static void main(String[] args)  
4     {  
5         System.out.println("5+2=" + 4+3);  
6         System.out.println("5+2=" +(4+3));  
7     }  
8 }  
9  
D:\durgaclasses>javac Test.java  
D:\durgaclasses>java Test  
5+2=43  
5+2=7
```

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 30 - 01 - 2018  
Press Esc to exit full screen  
1 class Test  
2 {  
3     public static void main(String[] args)  
4     {  
5         System.out.println("Result A:" + 4+5);  
6         System.out.println("Result B:" +(4)+(5));  
7     }  
8 }
```



```
D:\durgaclasses>javac Test.java  
D:\durgaclasses>java Test  
Result A:45  
Result B:45
```

You can apply relational operators on any type other than Boolean. For Boolean and objects relational ops throw compile time error. Line 45 throws error

```
40 Relational operators(<,<=,>,>=)
41 -----
42 sop(10<20);
43 sop(10<10.0);
44 sop(10<'a');
45 sop(true<false);
46
```

The screenshot shows a Java code editor with the following code:

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         System.out.println(10<20);
6         System.out.println(10<10.0);
7         System.out.println(10<'a');
8     }
9 }
10
11
```

The lines containing the relational operators `<`, `<10.0`, and `<'a'` are highlighted with yellow boxes.

The screenshot shows a terminal window with the following output:

```
D:\durgaclasses>java Test
true
false
true
```

The screenshot shows a Java code editor with the following code:

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         System.out.println(10<20);
6         System.out.println(10<10.0);
7         System.out.println(10<'a');
8         System.out.println(true<false);
9     }
10 }
```

The line `System.out.println(true<false);` is highlighted with a yellow box.

The screenshot shows a terminal window with the following output:

```
D:\durgaclasses>javac Test.java
Test.java:8: error: bad operand types for binary operator '<'
        System.out.println(true<false);
                           ^
first type: boolean
second type: boolean
1 error
```

The line `System.out.println(true<false);` is highlighted with a yellow box.

The screenshot shows a terminal window with the following output:

```
D:\durgaclasses>javac Test.java
Test.java:8: error: bad operand types for binary operator '<'
        System.out.println("durga"<"durga123");
                           ^
first type: String
second type: String
1 error
```

The line `System.out.println("durga"<"durga123");` is highlighted with a yellow box.

10>20 – true: True<30 , boolean on relational throws compile time error. Nesting of relational operator throws error

```
| 6     System.out.println(10<20<30);
```

Equality Operator:

The screenshot shows a Java code editor with the following code:

```
1 increment and decrement operators
2 arithmetic operators
3 String concatenation operator
4 Relational operators
5 Equality operators
6 -----
7 == !=
```

The line `7 == !=` is highlighted.

Equality op can be used for any primitive type. Line 12 , int gets promoted to bigger data type double so would return true. Can apply equality to boolean as well.

The screenshot shows a Java code editor with the following code:

```
7 == !=
8
9 10==20
10 'a'==10
11 'a'==97
12 10==10.0
13
14 10.0==10.0
15
16 false==false
```

The line `12 10==10.0` is highlighted.

Equality op can be applied to object types as well if both ref points to same object. == ops are used to compare ref value or add value

The screenshot shows a Java code editor with the following code:

```
20 r1==r2==>true
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
```

A red box highlights the line `r1==r2==>true`. Inside this box, there is a hand-drawn diagram showing two circles labeled r_1 and r_2 , with arrows pointing from each label to its respective circle.

To the right of the code editor is a messaging interface. The message history includes:

- true
- sir 10.0==10.0 also?
- true
- false
- sture
- true
- ok sir
- 10.0==10 ??
- same value
- both pointing to same obj
- refer to same obj
- mem add same
- when they have same value

Gautam Lohiya, Shubham Gupta, Rajat Gupta, Ashirbad Rout, Poja Chavan, Govindu Rayapur, Amit Kumar, Shubham Gupta, Govindu Rayapur, Shubham Gupta, RAMESWARA REDDY, Kalyan KK, Poja Chavan, Shubham Gupta

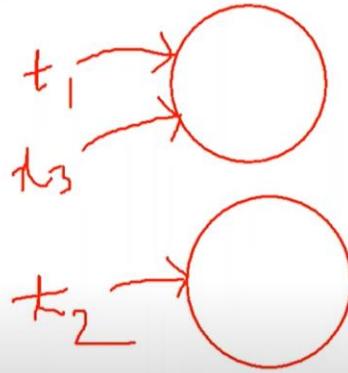
Good evening Sir!!

Send Privately Send to All

```

20 r1==r2====>true
21
22
23 Thread t1= new Thread();
24 Thread t2 = new Thread();
25 Thread t3=t1;
26
27
28 t1==t2
29
30 t1==t3
31
32
33
34

```



Pooja Chavan
Amit Kumar
Ashirbad Rout
Gautam Lohiya
SrujananReddy Govar...
Kalyan KK
RAMESWARA REDDY
Pooja Chavan
Rajat Gupta
Shubham Gupta
Ashirbad Rout
Kalyan KK
Govindu Rayapur
RAMESWARA REDDY
Amit Kumar

Good evening Siri!

Send Privately Send to All

```

28 Address comparison/reference comparison
29

```

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5         Thread t1= new Thread();
6         Thread t2 = new Thread();
7         Thread t3=t1;
8         System.out.println(t1==t2);
9         System.out.println(t1==t3);
10    }
11 }

```

```

D:\durgaclasses>javac Test.java
D:\durgaclasses>java Test
false
true

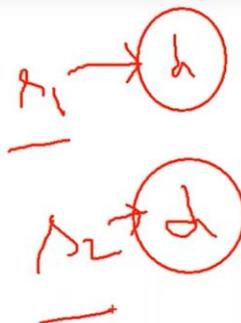
```

Below is false as new creates a new object and the add ref are diff

```

1 OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir Demo On 31-01-2018
2
3 class Test
4 {
5     public static void main(String[] args)
6     {
7         String s1= new String("durga");
8         String s2= new String("durga");
9         System.out.println(s1==s2);
10    }
11 }

```



X	Question	Asker
false		Shubham Gupta
false		Pooja Chavan
false		Amit Kumar
false		Rajat Gupta
true		RAMESWARA REDDY
false		Suhana babu
true		Sai Chand Kilaru
true		Deepankaj Yadav
false		Rajat Gupta
false		Kalyan KK
diff		Shubham Gupta
diff		Amit Kumar
diff		Govindu Rayapur
diff		Kalyan KK
sorry its false		RAMESWARA REDDY
ortrue		Sumit Bhattacharya

Good evening Siri!

These are two diff object types and returns false but,

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         Object o = new Object();
6         String s = new String("durga");
7         Thread t = new Thread();
8         System.out.println(o==s);
9         System.out.println(o==t);
10    }
11 }
```

There is a relation between Object -> String and Object -> Thread . But there is no relation btw Thread and String , u cant apply equality operator on objects with no relation. So compile time error on line 10. There should be relation btw argument types

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         Object o = new Object();
6         String s = new String("durga");
7         Thread t = new Thread();
8         System.out.println(o==s);
9         System.out.println(o==t);
10        System.out.println(t==s);
11    }
12 }
```

D:\durgaclasses>javac Test.java
Test.java:10: error: incomparable types: Thread and String
System.out.println(t==s);
^
1 error

Exam equals specific to String only not other objects

29
30
31
32 Difference between == operator and equals() method:
33 -----
34
35

Even if object is diff if content is same equals() returns true

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
```

Difference between == operator and equals() method:

I
== reference comparison
equals() method content comparison

Line 7 is false and Line 8 true

```
1 2 3 4 5 6 7 8 9 10 11
```

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

Compile time error no relation btw string and String buffer

```
1 2 3 4 5 6 7 8 9 10 11
```

```
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));
    }
}
```

Equals method does not check for relation but if the object is different, it returns false

```
1 2 3 4 5 6 7 8 9 10 11
```

```
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
    }
}
```

```

45
46 sir but in this type is same. it return true
47 Thread t=new thread();
48 Thread t2=t;
49 sop(t.equals(t2))
50

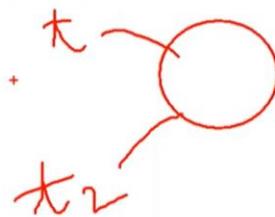
```

T1 and T2 content are the same, since they point to same object

```

46 sir but in this type is same. it return true
47 Thread t=new thread();
48 Thread t2=t;
49 sop(t.equals(t2))
50
51

```



true	Rajat Gupta
1	Shubham Gupta
1	Pooja Chavan
2	Rajat Gupta
1	Ashibad Rout
1	Rajat Gupta
1	Gowindu Rayapur
1	Deepankaj Yadav

sir but in this type is same. it return true Thread t=new new
thread(); Thread t2=t; sop(t.equals(t2))

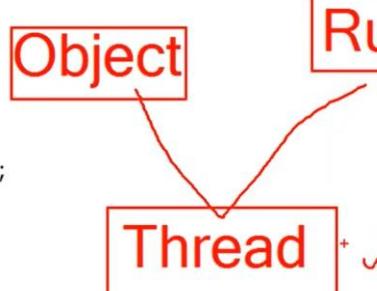
Instanceof operator – instanceof is a reserved word and operator should be lower case . reserved words are lower case

All true

```

51 -----
52 instanceof operator
53
54
55
56 Thread t = new Thread();
57 sop(t instanceof Thread);
58 sop(t instanceof Runnable);
59 sop(t instanceof Object);
60
61
62
63
64

```



X	Question	Asker
capital	john acharya	
small	Rajat Gupta	
O	Amit Kumar	RAMESWARA R...
small	Shubham Gupta	
true	Shubham Gupta	
all are true	Shubham Gupta	
true	Pooja Chavan	
true	Gowindu Rayapur	
true	john acharya	
true	Pooja Chavan	
true	john acharya	
true	Gowindu Rayapur	
true	john acharya	
true	Pooja Chavan	
true true true	RAMESWARA R...	

```

D:\durgaclasses>java Test
true
true
true

```

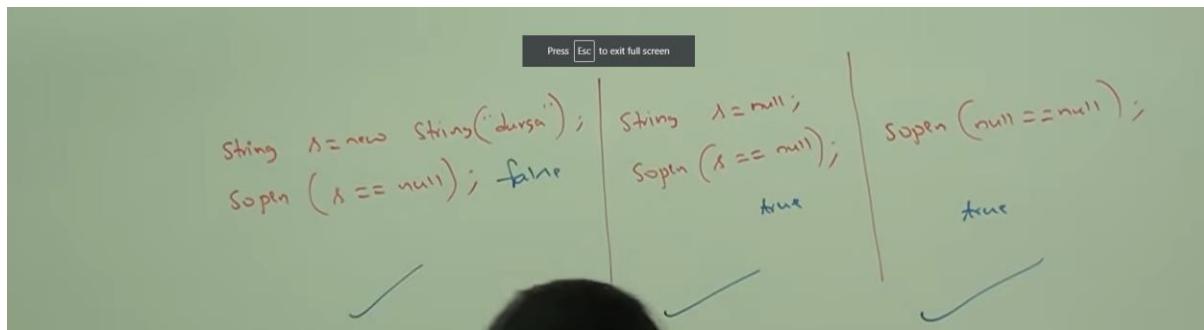
If you want to use instanceof operator there should be some relation. Else it will throw

Inconvertible types; cannot cast 'java.lang.String' to 'java.lang.Thread'

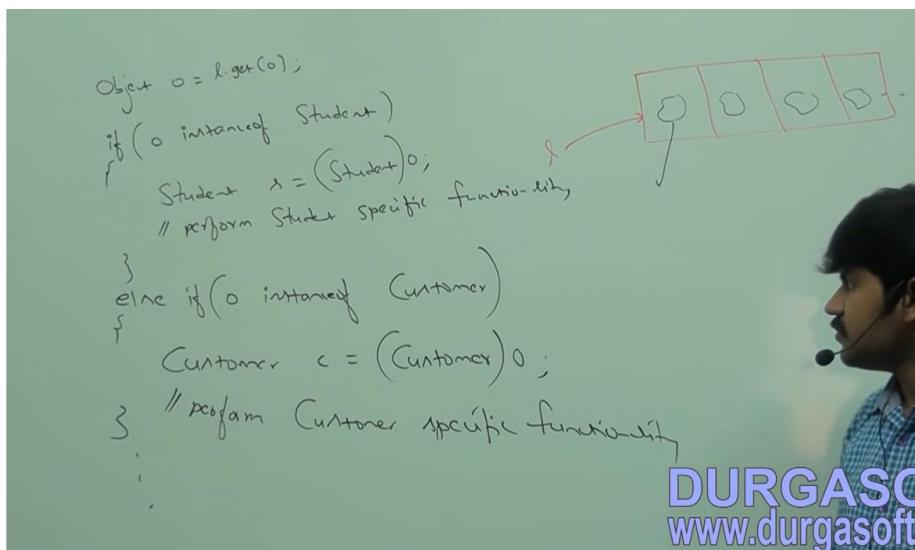
A screenshot of a Java IDE interface. On the left, a code editor shows a class named 'Test' with the following code:

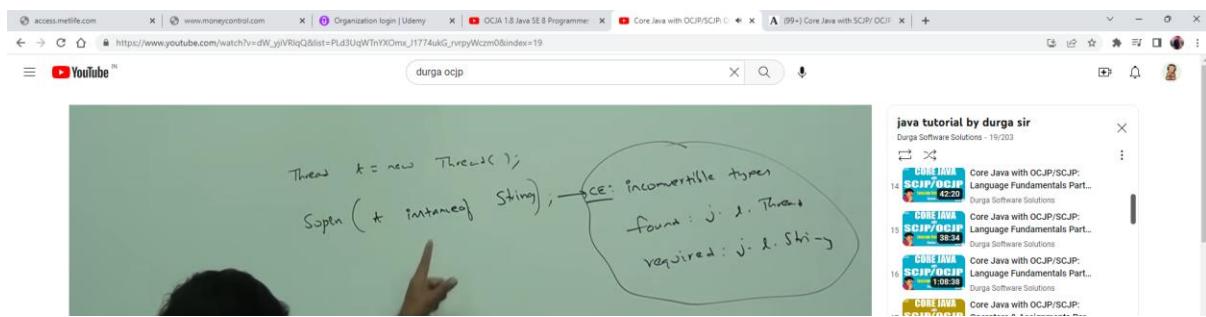
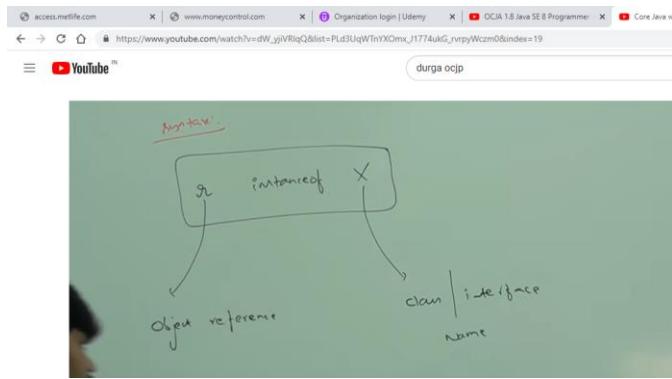
```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         Thread t = new Thread();
6         System.out.println(t instanceof Thread);
7         System.out.println(t instanceof Runnable);
8         System.out.println(t instanceof Object);
9         System.out.println(t instanceof String);
10    }
11 }
12
```

The line `System.out.println(t instanceof String);` is highlighted with a red box and has a red arrow pointing from it to the question list on the right. The question list is titled 'Questions' and contains a table with columns 'X', 'Question', and 'Asker'. It lists various questions and their answers, such as 'true', 'false', and 'null'.

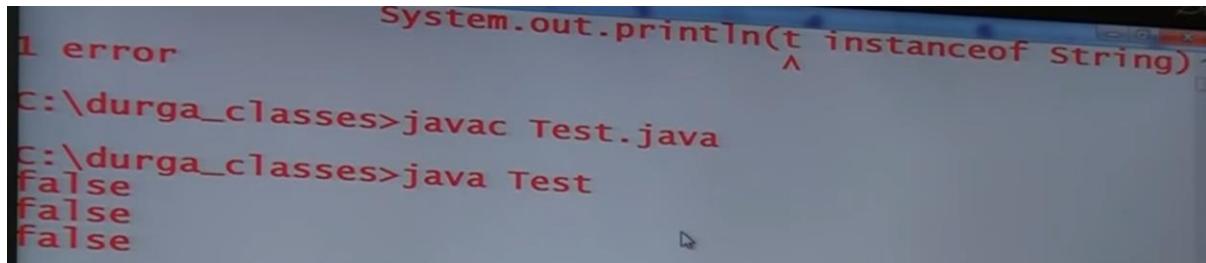
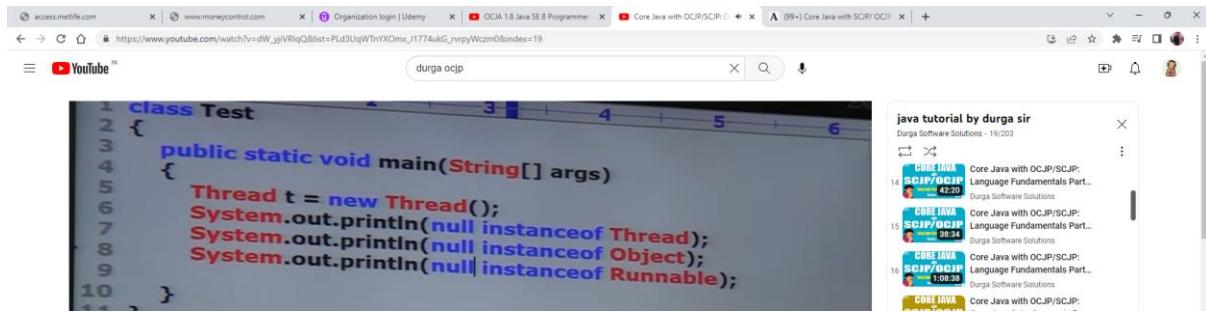


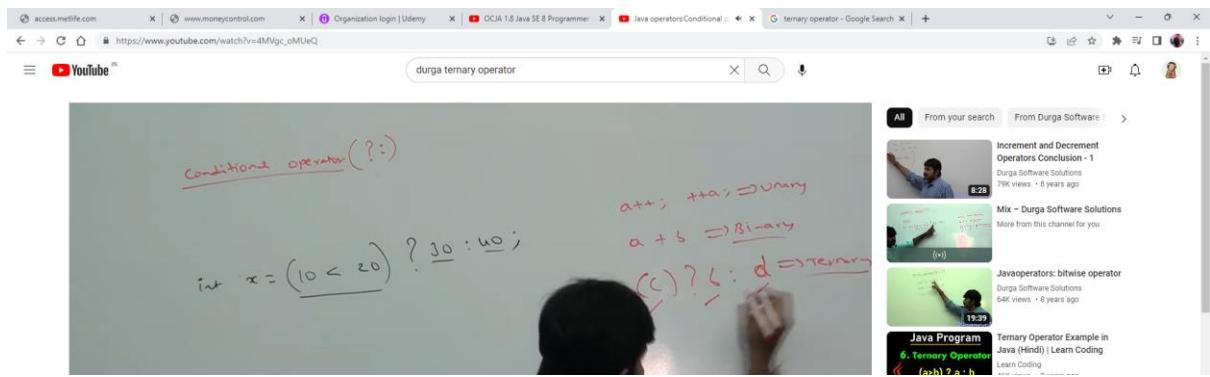
ArrayList can be of anytype



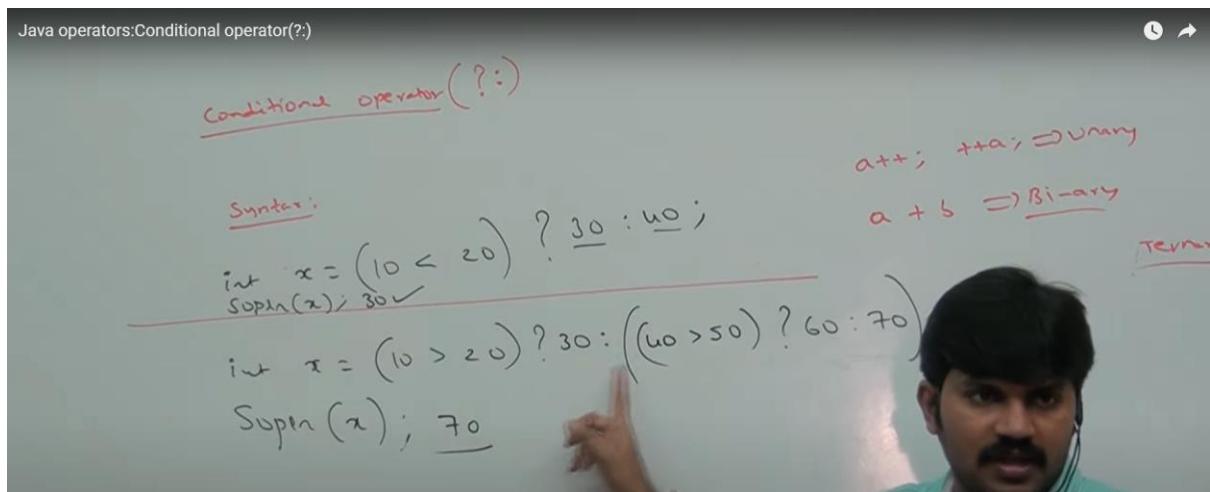


For any class or interface x -> null instance of x , is always false





Nesting of conditional operator ternary operator is possible



pdfcoffee.com_2-operators-amp-assignmentspdf-free.pdf - Adobe Acrobat Reader DC (32-bit)

File Edit View Sign Window Help

Home Tools Core_Java_with_SCJP... DurgaSoft-SCJP-no... pdfcoffee.com_2-o... 22 / 26

Conditional Operator (?)

The only possible ternary operator in java is conditional operator

```
Ex 1 :
int x=(10>20)?30:40;
System.out.println(x); //40
```

```
Ex 2 :
int x=(10>20)?30:((40>50)?60:70);
System.out.println(x); //70
```

Nesting of conditional operator is possible

```
int x=(10>20)?30:(100>20)?40:50;
System.out.println(x); //40
```

```
int x=(10>20)?30:(100>20)?40:50;
System.out.println(x); //50
```

```
int a=10,b=20;
byte c1=(10>20)?30:40;
byte c2=(10>20)?30:40;
System.out.println(c1); //40
System.out.println(c2); //30
```

```
int a=10,b=20;
byte c1=(a>b)?30:40;
byte c2=(a>b)?30:40;
System.out.println(c1);
System.out.println(c2);
```

```
E:\scjp>javac OperatorsDemo.java
OperatorsDemo.java:6: possible loss of precision
        found : int
               required: byte
                byte c1=(a>b)?30:40;
```

new operator :

BIT Mostly sunny

Search

ENG IN 12-02-2016 10:16

```

1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8-----+-----9-----+-----0-----+
5
6 Bitwise operator:
7 -----
8 &==>if both arguments are true
9 |==>If atleast one argument is true
10 ^==>X-OR==>If both arguments are different
11
12 sop(true&true)==>true
13 sop(true|false)==>true
14 sop(true^false)==>true
15 sop(true^true)==>false

```

```

OCJA 1.8 Java SE 8 Programmer - I (1Z0-808) By Durga Sir Demo On 01-02-2018
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8-----+-----9-----+-----0-----+
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         System.out.println(true&false);
6         System.out.println(true|false);
7         System.out.println(true^false);
8     }
9 }

```

```

D:\durgaclasses>java Test
false
true
true
true

```

If we apply bitwise operator on nos, the operation is performed bitwise and returns a number.

Bitwise operator cant be applied only on Boolean and itegrals not floating pt

```

16 sop(4&5)// 4
17 sop(4|5)// 5
18 sop(4^5)// 1
19
20 100
21 101
22 100
23 101
24 100
25 101
26
27

```

these are bitwise not logical Shubham Gupta

Send Privately Send to All

In 22 col 4 37 00 PC ANSI

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5         System.out.println(4&5);
6         System.out.println(4|5);
7         System.out.println(4^5);
8     }
9 }

```

```

4 D:\durgaclasses>javac Test.java
5
6 D:\durgaclasses>java Test
4
5
8
1

```

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5         System.out.println(4.0&5); t
6
7
8
9
10
11

```

```

D:\durgaclasses>javac Test.java
Test.java:5: error: bad operand types for binary operator '&'
        System.out.println(4.0&5);
                           ^
      first type:  double
      second type: int
1 error

```

A internally means 95 and 97

Output -65 perform bitwise comparison

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5         System.out.println('a' & 'A');
6
7     }
8 }
9

```

Bitwise complement op ~ is applicable only for integral nt boolean below is an error

```

33 bit-wise complement operator(~):
34 -----
35 sop(~true)
36

```

1s complement 0 becomes 1 and 1 becomes 0

30 ^
31 -----
32
33 bit-wise complement operator(~):
34 -----
35 sop(~true)
36 sop(~4)

37 | 0 0
38 | 1 1
39 | 0 1
40 | 1 1
41 | 1 1
42 | 1 1
43 | 1 1
44 | 1 1
45 | 1 1

3

Questions

X	Question	Asker	R...
valid		john acharya	
-5 is ans		Shubham Gupta	
-5		Shubham Gupta	
-5		Govindu Rayapur	
1		Deepankaj Yadav	
11		john acharya	
false		Amit Kumar	
-4		Ramanathan KRISHN...	
-5		Shubham Gupta	
45		Pooja Chavan	
5		Pooja Chavan	

Send Privately Send to All

Untitled2 Test.java In 37 col1 2 00 PC ANSI

Applicable only for boolean not others . True -> false and false is true

37
38 boolean complement operator(!):
39 -----
40 sop(!true)===>
41 sop(!4)===>CE

42
43
44
45
46 &
47 |
48 ~
49 !

42
43
44
45
46 &
47 |
48 ~
49 !

ce CE error oh s yes clear Ok sir What you called title of videos? Govindu Rayapur Pooja Chavan Sai chand Kilari Shubham Gupta Pooja Chavan Shubham Gupta Amit Kumar Kalyan TC Ramanathan KRISHN... Deepankaj Yadav

Send Privately Send to All

If & and | - in the below both username and pwds will be evaluated and then it comes out even when 1 condition fails whereas in && || even if 1 fails it will not execute the second condition

```

55
56
57 short circuit operators(&&, ||)
58 -----
59
60 & and      &&
61 | and      ||
62
63
64 if(usernames & pwds) 1+
65 { 10
66   perform this operation
67 }
68 else
69 {
70 ...
71 }
72 }
73

```

The screenshot shows a Java code editor with the following code:

```

55
56
57 short circuit operators(&&, ||)
58 -----
59
60 & and      &&
61 | and      ||
62
63
64 if(usernames & pwds) 1+
65 { 10
66   perform this operation
67 }
68 else
69 {
70 ...
71 }
72 }
73

```

Annotations in red:

- A red arrow points from the text "1+" to the "&" operator.
- A red arrow points from the number "10" to the opening brace of the if block.

To the right of the code editor is a "Questions" panel showing a list of user questions and answers.

```

64
65 x && y ==> y will be evaluated iff x is true
66
67 x || y ==> y will be evaluated iff x is false
68
69

```

The screenshot shows a Java code editor with the following code:

```

64
65 x && y ==> y will be evaluated iff x is true
66
67 x || y ==> y will be evaluated iff x is false
68
69

```

Annotations in yellow:

- "x && y ==> y will be evaluated iff x is true"
- "x || y ==> y will be evaluated iff x is false"

&& || ops only applicable for Boolean and not integrals . whereas & | applicable for integers and boolean

The screenshot shows a Java code editor with the following code:

```

64
65 x && y ==> y will be evaluated iff x is true
66
67 x || y ==> y will be evaluated iff x is false
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84

```

Handwritten red annotations:

- A large circle is drawn around the first two lines of code: "x && y ==> y will be evaluated iff x is true" and "x || y ==> y will be evaluated iff x is false".
- Red arrows point from the variables "x" and "y" in these lines to a large circle labeled "boolean".
- Red arrows point from the operators "&&" and "||" to another large circle labeled "integers".

To the right of the code editor is a "Questions" panel showing a list of user questions and answers.

Very important - & will perform both .oversight

The screenshot shows an IDE interface with a Java code editor and a question list. The code is as follows:

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int x = 10;
6         int y = 15;
7         if(++x<10 && ++y>15)
8         {
9             x++;
10        }
11        else
12        {
13            y++;
14        }
15        System.out.println(x+".."+y);
16    }
17 }
```

Annotations and handwritten notes:

- A yellow oval highlights the condition `if(++x<10 && ++y>15)`.
- Red boxes highlight the increment operators `++x` and `++y`.
- A red bracket underlines the entire condition `if(++x<10 && ++y>15)`.
- A red arrow points from the handwritten note $11 - 17$ to the condition.
- A handwritten note $11 - 17$ is written below the code editor.

The screenshot shows an IDE interface with a Java code editor and a question list. The code is identical to the one above:

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int x = 10;
6         int y = 15;
7         if(++x<10 && ++y>15)
8         {
9             x++;
10        }
11        else
12        {
13            y++;
14        }
15        System.out.println(x+".."+y);
16    }
17 }
```

Annotations and handwritten notes:

- A yellow oval highlights the condition `if(++x<10 && ++y>15)`.
- Red boxes highlight the increment operators `++x` and `++y`.
- A red bracket underlines the entire condition `if(++x<10 && ++y>15)`.
- A red arrow points from the handwritten note $11 - 16$ to the condition.
- A handwritten note $11 - 16$ is written below the code editor.

```

70 Assignment operators:
71 -----
72 1. Simple assignment:
73 -----
74 x=10;
75
76 2.Chained assignments
77 -----
78 a=b=c=d=20;
79
80 3. Compound Assignment operator:
81 -----
82 a += 20;

```

Not allowed in below

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int a,b,c,d;
6         a=b=c=d=20;
7         System.out.println(a,b,c,d);
8     }
9 }
10

```

Assignment is right associate , first assigns 20 to d and d is assigned to c , c is assigned to be so on

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int a,b,c,d;
6         a=b=c=d=20;
7         System.out.println(a+".."+b+".."+c+".."+d);
8     }
9 }
10

```

X	Question	Answer
python	Shubham Gupta	
no	John Acharya	
no	Sai Chand Kilari	
no	Pooja Chavan	
nooooooo	Govindu Rayapur	
no	Amit Kumar	
becauseof,	Sai Chand Kilari	
20..20..20	Kalyan TC	
20 20 20	Shubham Gupta	
20202020	Govindu Rayapur	
CE	Pooja Chavan	

Right Associative

A is declared as in whereas others b ,c and d are not declared. Cannot find symbols b,c and d. Chained assignment is not possible at time of declaration

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int a=b=c=d=20;
6         System.out.println(a+..+b+..+c+..+d);
7     }
8 }

```

```

8 D:\durgaclasses>javac Test.java
9 D:\durgaclasses>java Test
10 -160...-180...200..10
11

```

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int a,b,c,d;
6         a=b=c=d=20;
7         a += b -= c *= d /= 2;
8         System.out.println(a+..+b+..+c+..+d);
9     }
10 }
11

```

Handwritten annotations:

- $a = 20 - 180$
- -160
- -10
- 200

Red circled code: $a += b -= c *= d /= 2;$

Red checkmark: \checkmark

Right side panel: Questions

X	Question	Asker
200 -200 200 10	Pooja Chavan	
bcd is 20	Amit Kumar	
a will be calculate	Amit Kumar	
a=20,b=0,c=20,d=10	Deepankaj Yadav	
for a	Amit Kumar	
20 180 200 10	RAMESWARA REDDY	
20	Pooja Chavan	
20	RAMESWARA REDDY	
20	Deepankaj Yadav	
20	john acharya	
20	Govindu Rayapur	

OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 01-02-2018

```

91
92 Conditional operator (?:):
93 -----
94
95
96 x=(b)?c:d;
97
98
99
100
101

```

The diagram shows the expression `x=(b)?c:d;` with annotations:

- $x++ \rightarrow \text{Unary}$
- $a+b \rightarrow \text{Binary}$
- A box around `(b)?c:d` is labeled **Ternary**.
- The expression is broken down into three parts: `(b)`, `?c`, and `:d`. The `(b)` part is labeled **1**, the `?c` part is labeled **2**, and the `:d` part is labeled **3**.

Questions pane (right side):

X	Question	Asker
20		RAMESWARA REDDY
20		Deepankaj Yadav
20		john acharya
20		Govindu Rayapur
-180,-180,200,10		Kalyan TC
-180,-180,200,10		Kalyan TC
ohhhhhhhhhh		Shubham Gupta
awesommm		dinesh durga soft
s		Pooja Chavan
yes		Kalyan TC
hurrah		john acharya

Send Privately | Send to All

70

OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 01-02-2018

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int x =(10>20)?30:((40>50)?60:70);
6         System.out.println(x);
7     }
8 }
9 Nesting of conditional operator is allowed....
10

```

OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 01-02-2018

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int x =(10>20)?30:((40>50)?60:70);
6         System.out.println(x);
7     }
8 }

```

OCJA 1.8 Java SE 8 Programmer - I (120 - 808) By Durga Sir Demo On 01-02-2018

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int x = (10>20)?30:(40>50)?60:70;
6         System.out.println(x);
7     }
8 }
```

70 ✓

Send Privately Send to All

Untitled2 Test.java 59:32 / 1:01:25

In 7 col 6 9 00 PC Ansel

S