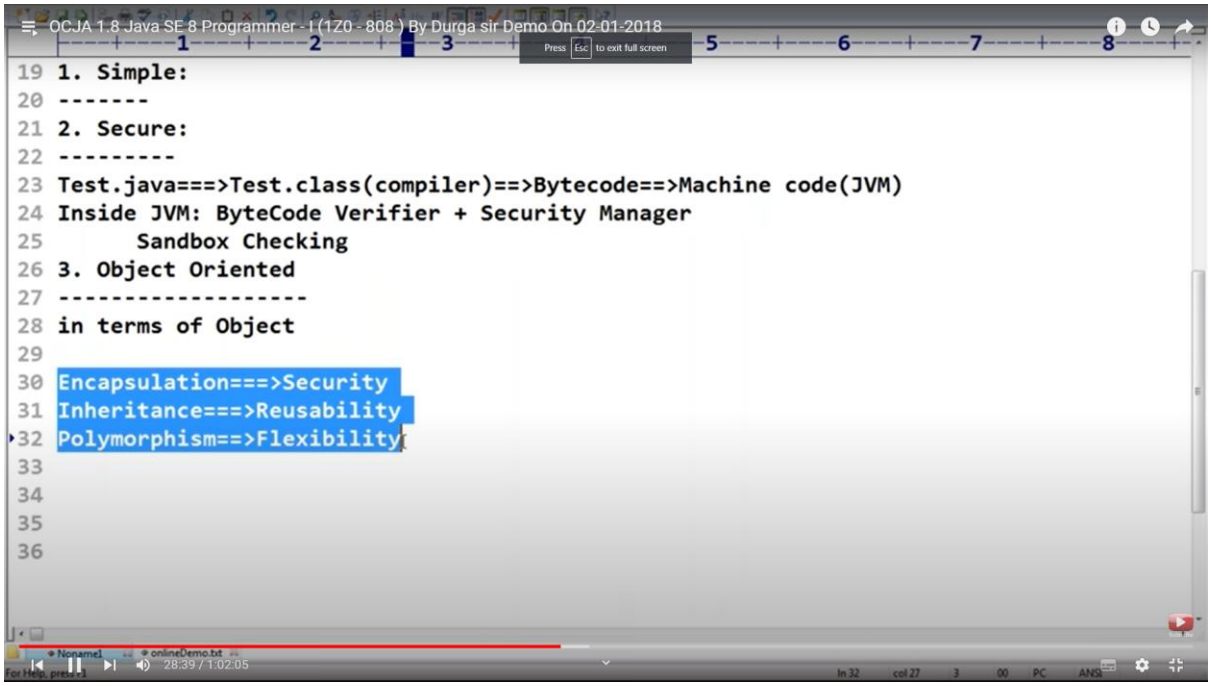
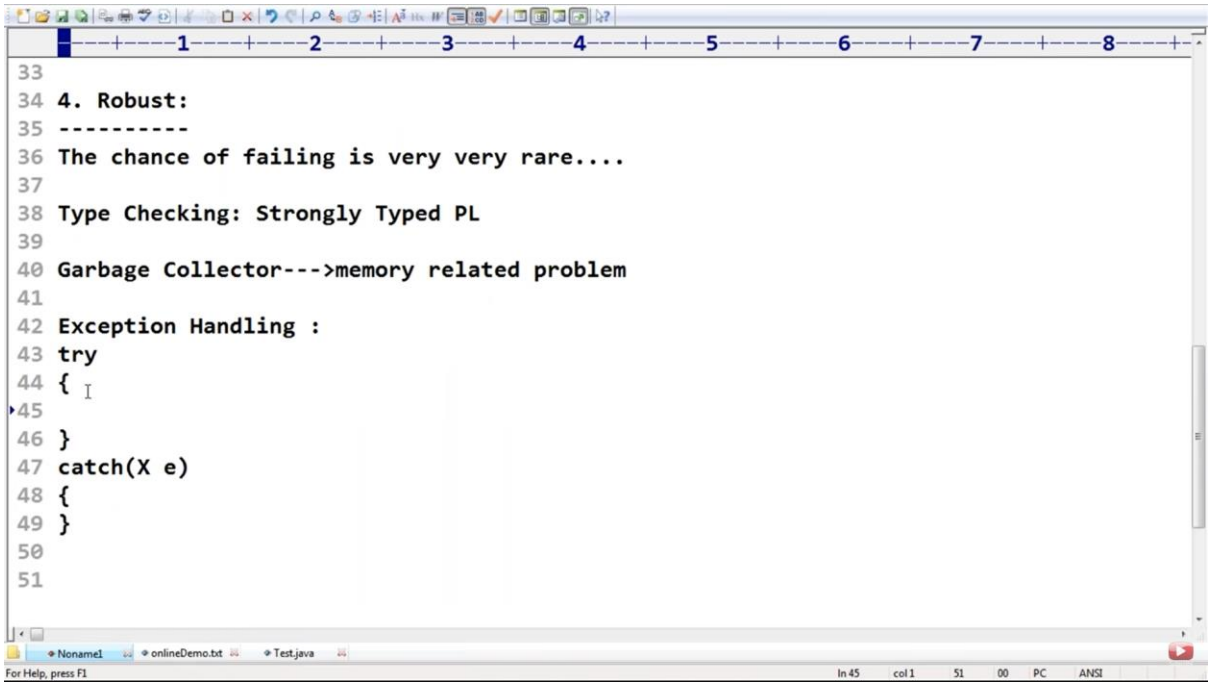
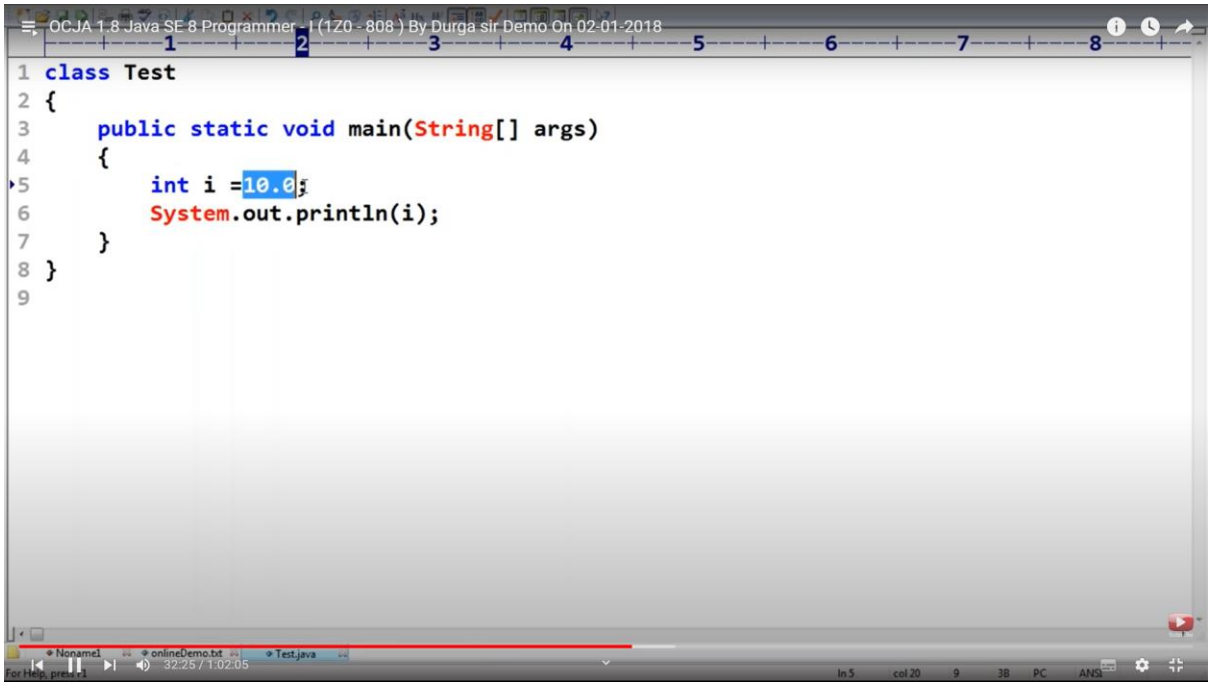
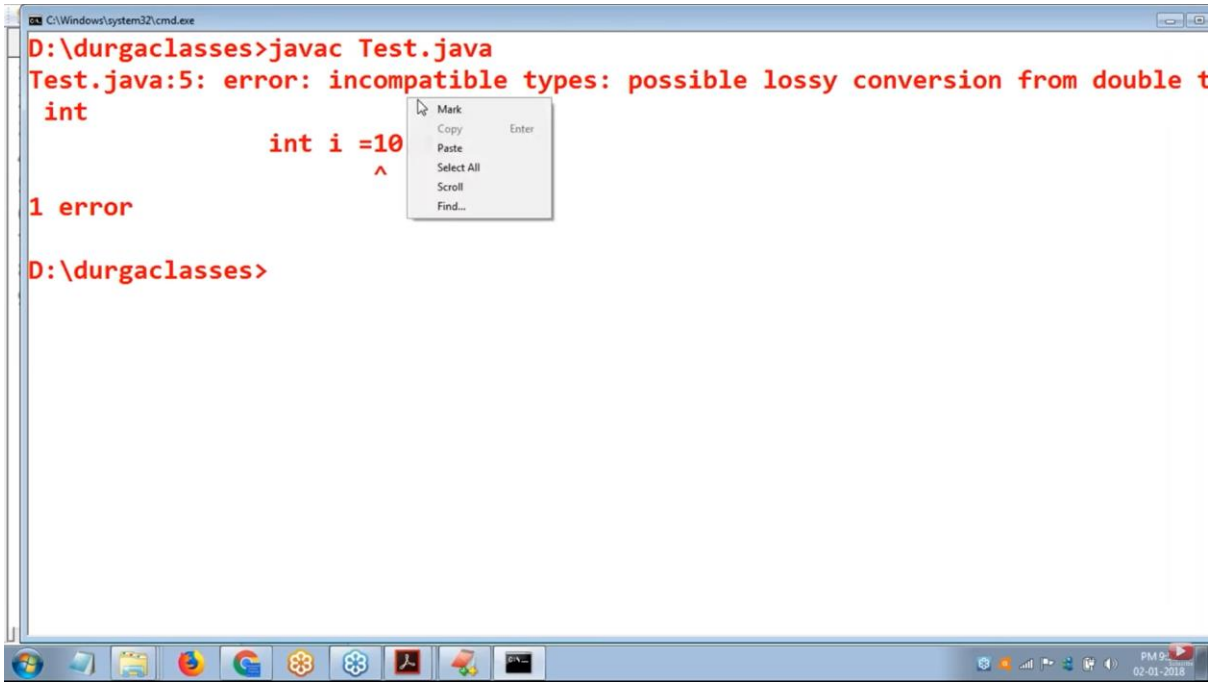

OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga sir Demo On 02-01-2018
Press Esc to exit full screen
1. Simple:
2. Secure:
Test.java===>Test.class(compiler)==>Bytecode==>Machine code(JVM)
Inside JVM: ByteCode Verifier + Security Manager
Sandbox Checking
3. Object Oriented
in terms of Object
Encapsulation===>Security
Inheritance===>Reusability
Polymorphism==>Flexibility
Noname1
onlineDemo.txt
28:39 / 1:02:05

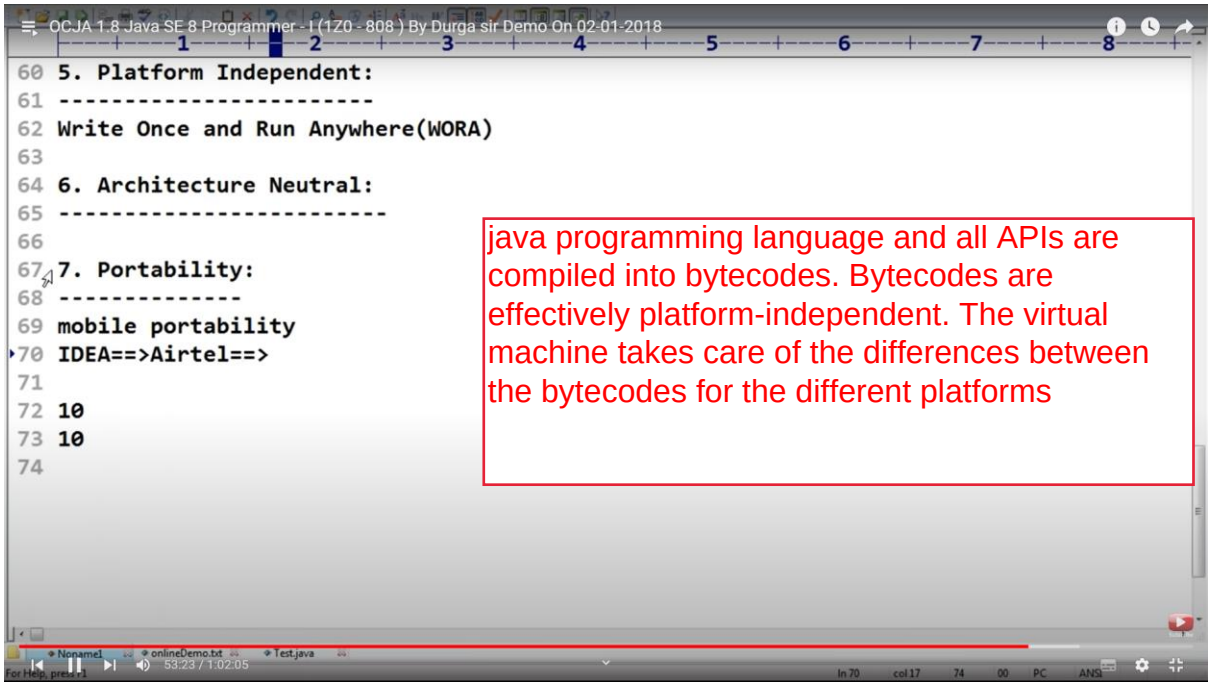

4. Robust:
The chance of failing is very very rare....
Type Checking: Strongly Typed PL
Garbage Collector--->memory related problem
Exception Handling :
try
{
}
catch(X e)
{
}
Noname1
onlineDemo.txt
Test.java
For Help, press F1

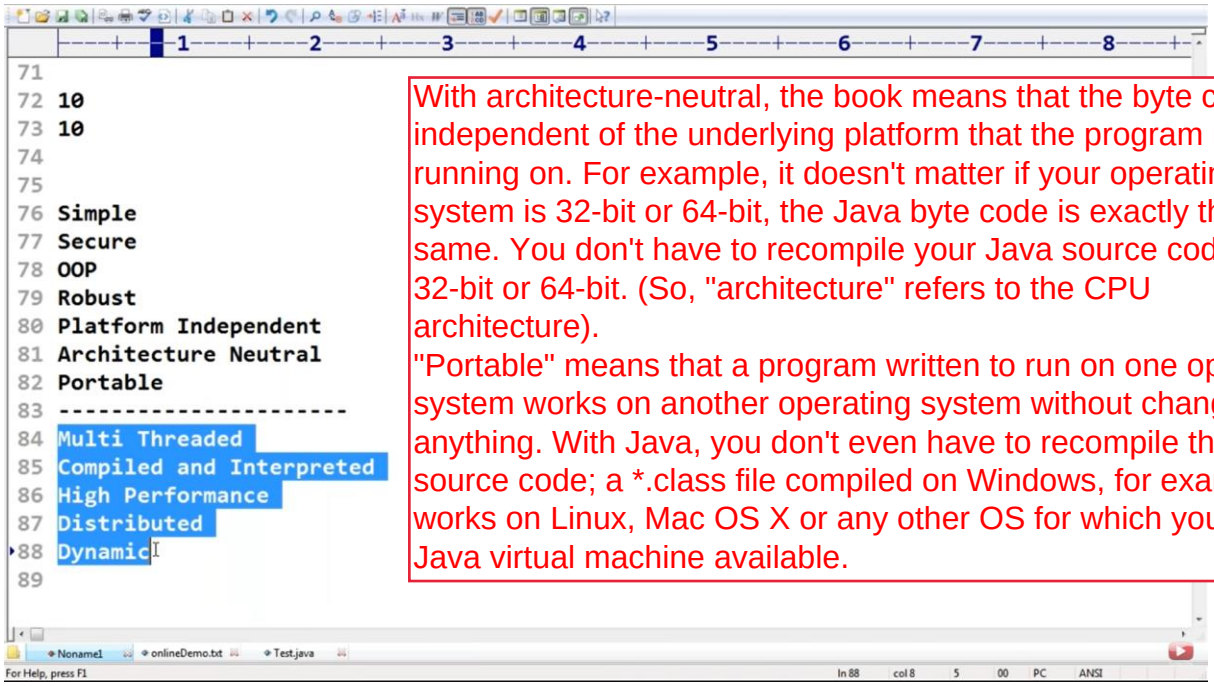


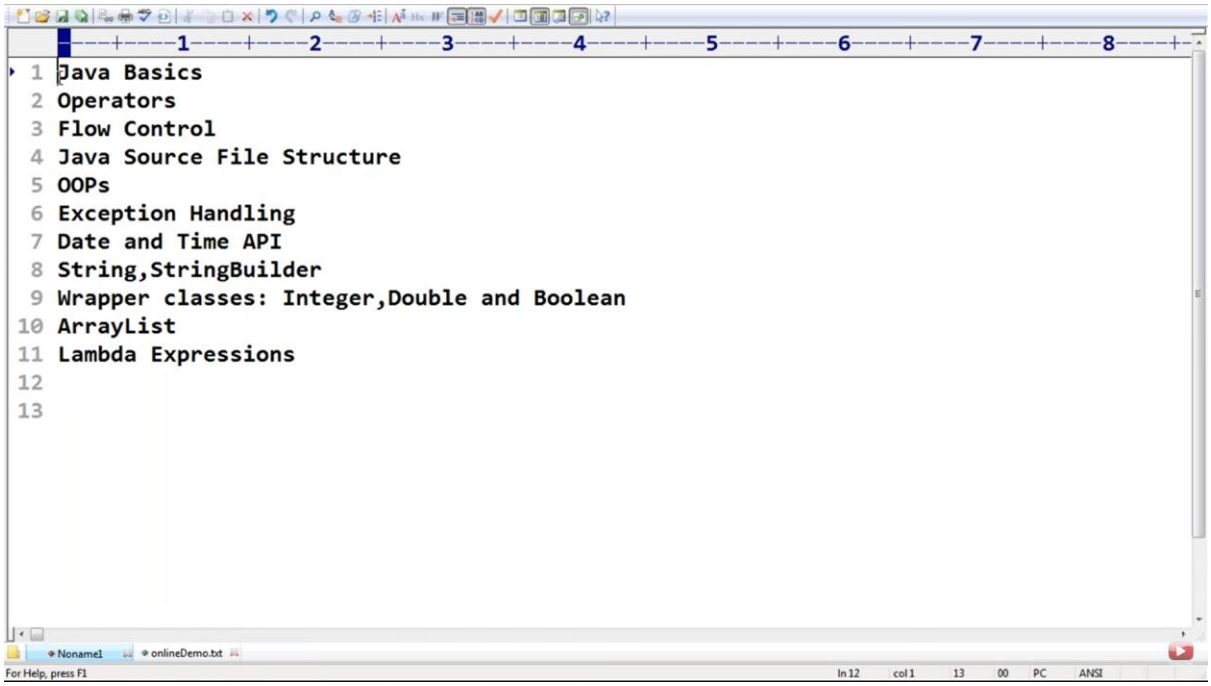
Compile time error

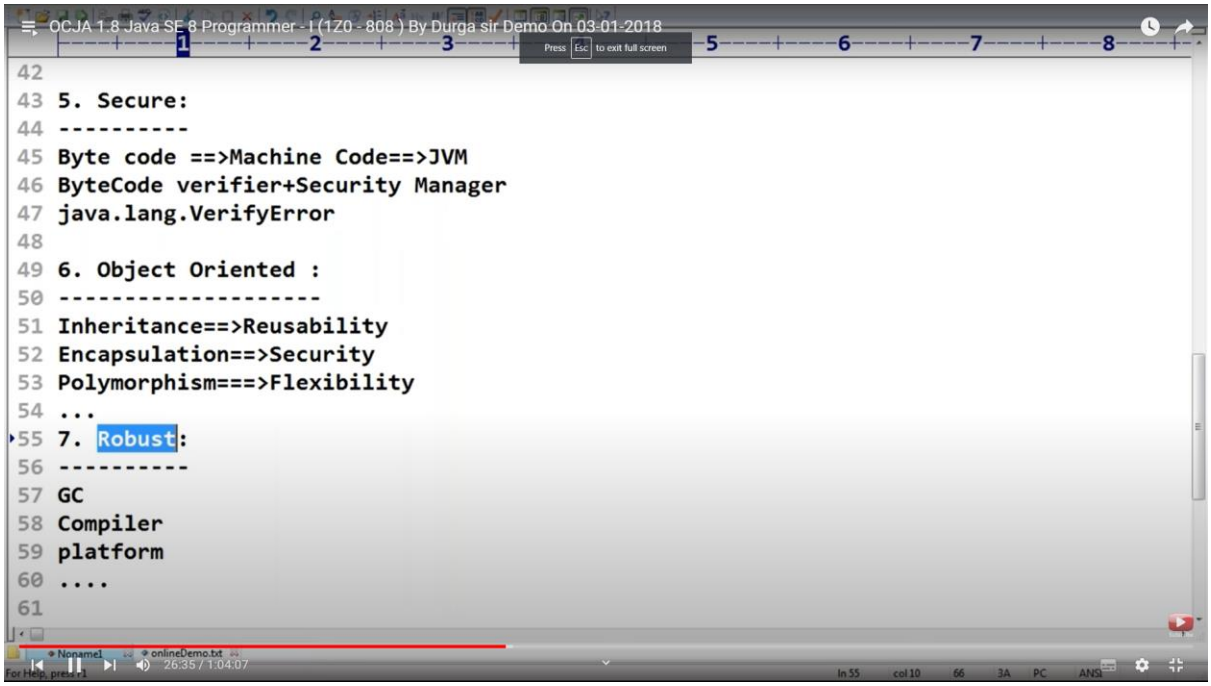


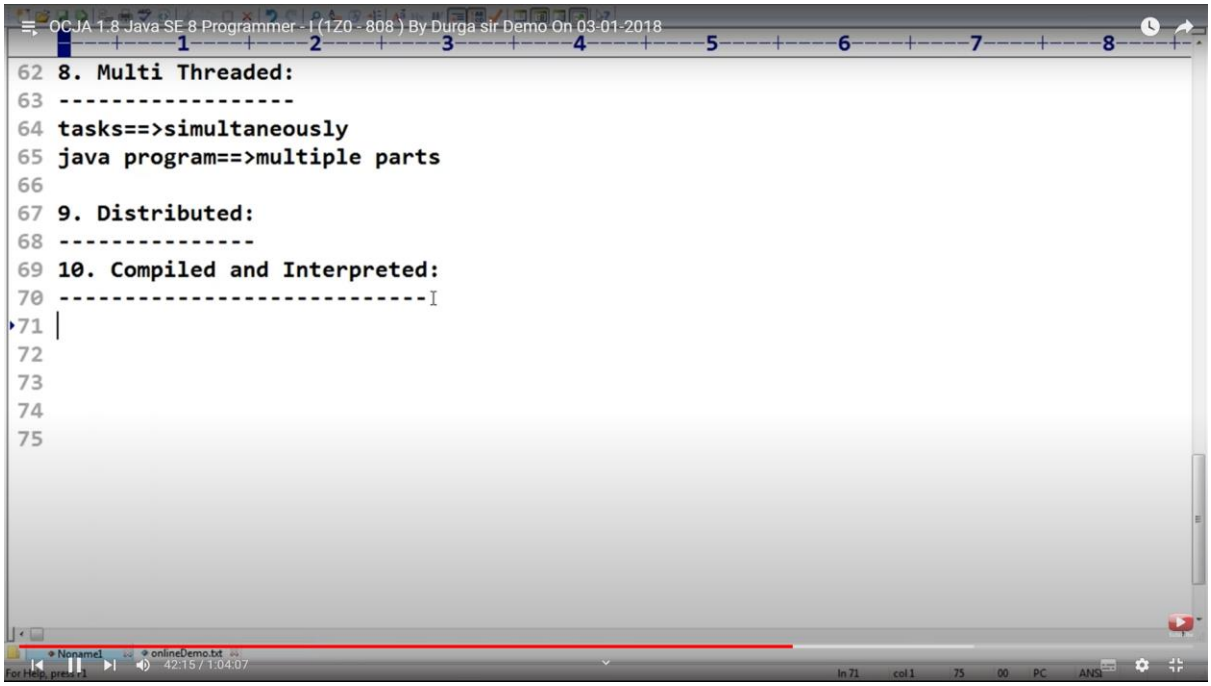
Since jave ais platform independent its easily portable and can be executed in any machine

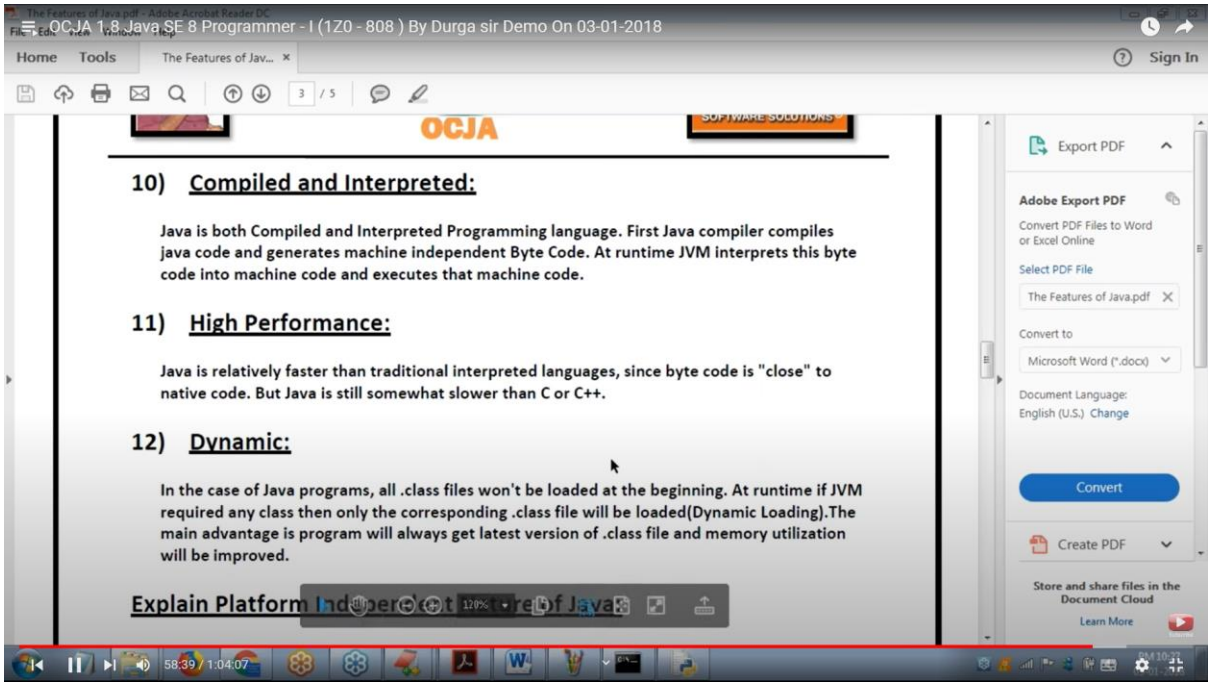

OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga sir Demo On 02-01-2018
5. Platform Independent:
Write Once and Run Anywhere(WORA)
6. Architecture Neutral:
7. Portability:
mobile portability
IDEA==>Airtel==>
10
10
java programming language and all APIs are compiled into bytecodes. Bytecodes are effectively platform-independent. The virtual machine takes care of the differences between the bytecodes for the different platforms

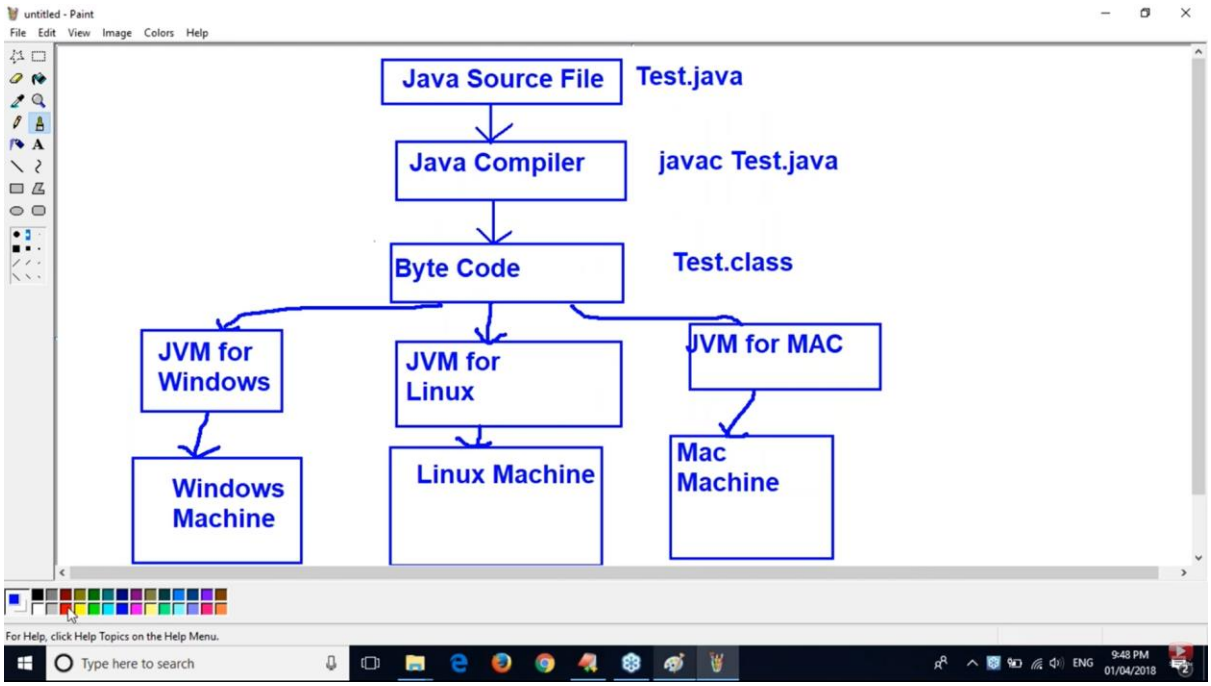

10
10
Simple
Secure
OOP
Robust
Platform Independent
Architecture Neutral
Portable
Multi Threaded
Compiled and Interpreted
High Performance
Distributed
Dynamic
With architecture-neutral, the book means that the byte code is independent of the underlying platform that the program is running on. For example, it doesn't matter if your operating system is 32-bit or 64-bit, the Java byte code is exactly the same. You don't have to recompile your Java source code for 32-bit or 64-bit. (So, "architecture" refers to the CPU architecture).
"Portable" means that a program written to run on one operating system works on another operating system without changing anything. With Java, you don't even have to recompile the source code; a *.class file compiled on Windows, for example, works on Linux, Mac OS X or any other OS for which you have a Java virtual machine available.

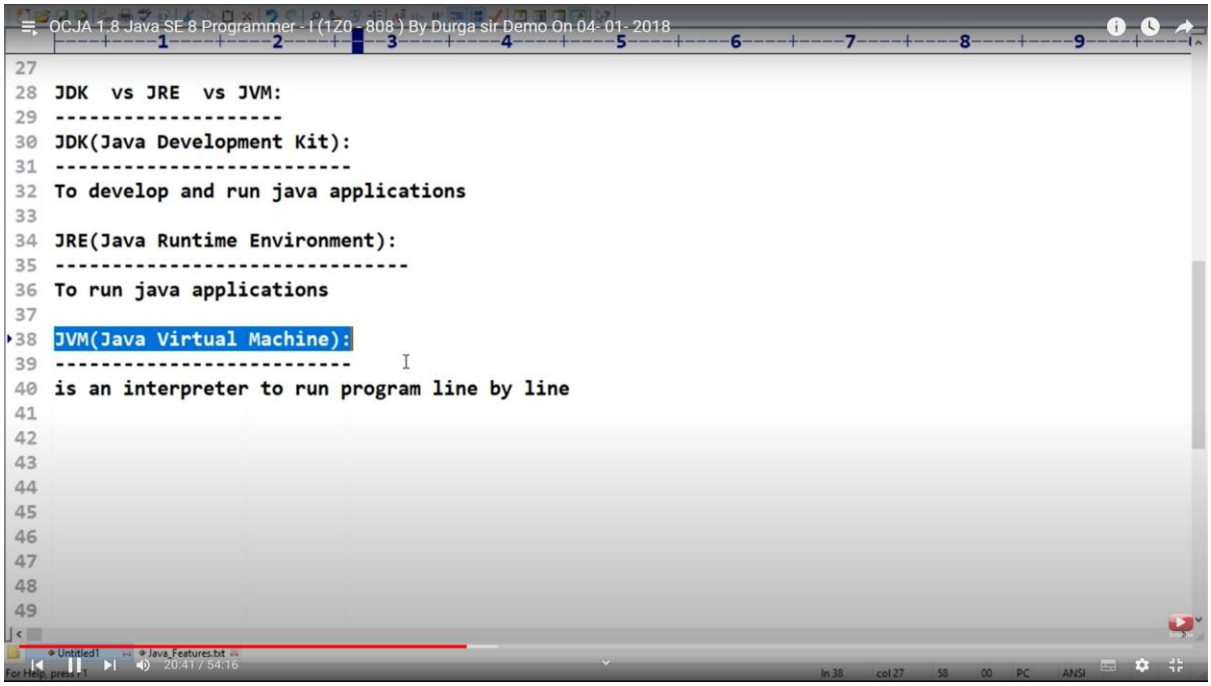

1 Java Basics
2 Operators
3 Flow Control
4 Java Source File Structure
5 OOPs
6 Exception Handling
7 Date and Time API
8 String,StringBuilder
9 Wrapper classes: Integer,Double and Boolean
10 ArrayList
11 Lambda Expressions
12
13
Noname1
onlineDemo.txt
For Help, press F1


OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga sir Demo On 03-01-2018
Press Esc to exit full screen
42
43 5. Secure:
44 ----------
45 Byte code ==>Machine Code==>JVM
46 ByteCode verifier+Security Manager
47 java.lang.VerifyError
48
49 6. Object Oriented :
50 --------------------
51 Inheritance==>Reusability
52 Encapsulation==>Security
53 Polymorphism===>Flexibility
54 ...
55 7. Robust:
56 ----------
57 GC
58 Compiler
59 platform
60 ....
61
Noname1
onlineDemo.txt
26:35 / 1:04:07

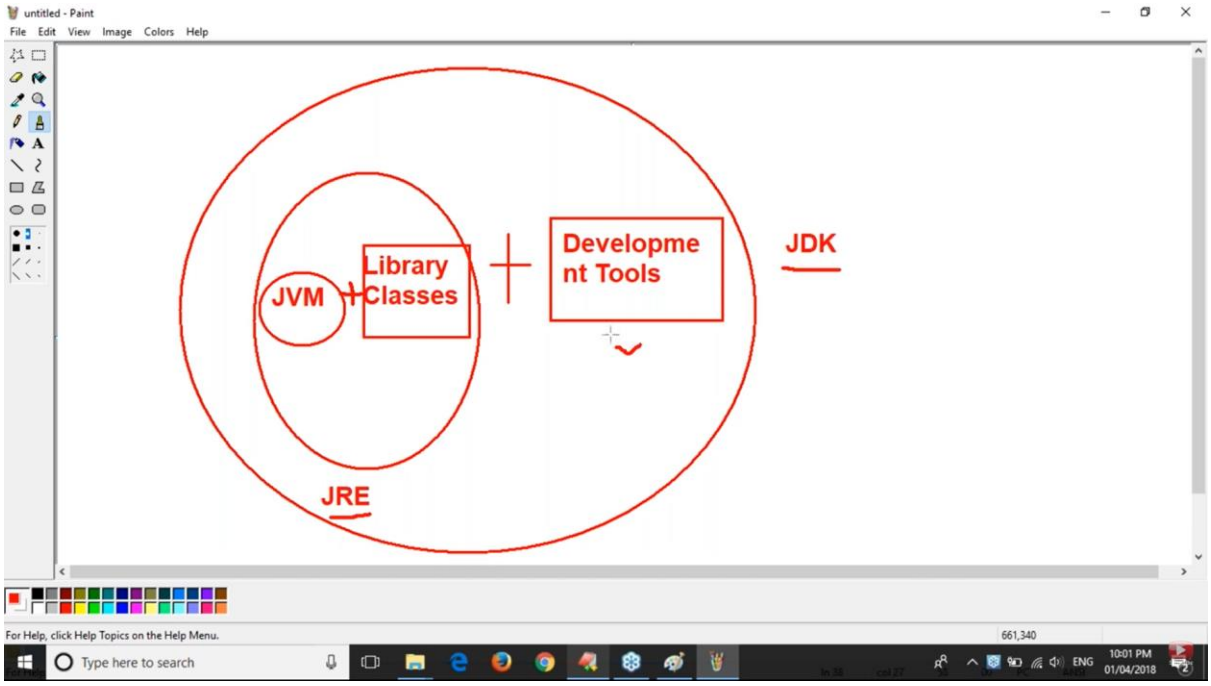

OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga sir Demo On 03-01-2018
8. Multi Threaded:
tasks==>simultaneously
java program==>multiple parts
9. Distributed:
10. Compiled and Interpreted:

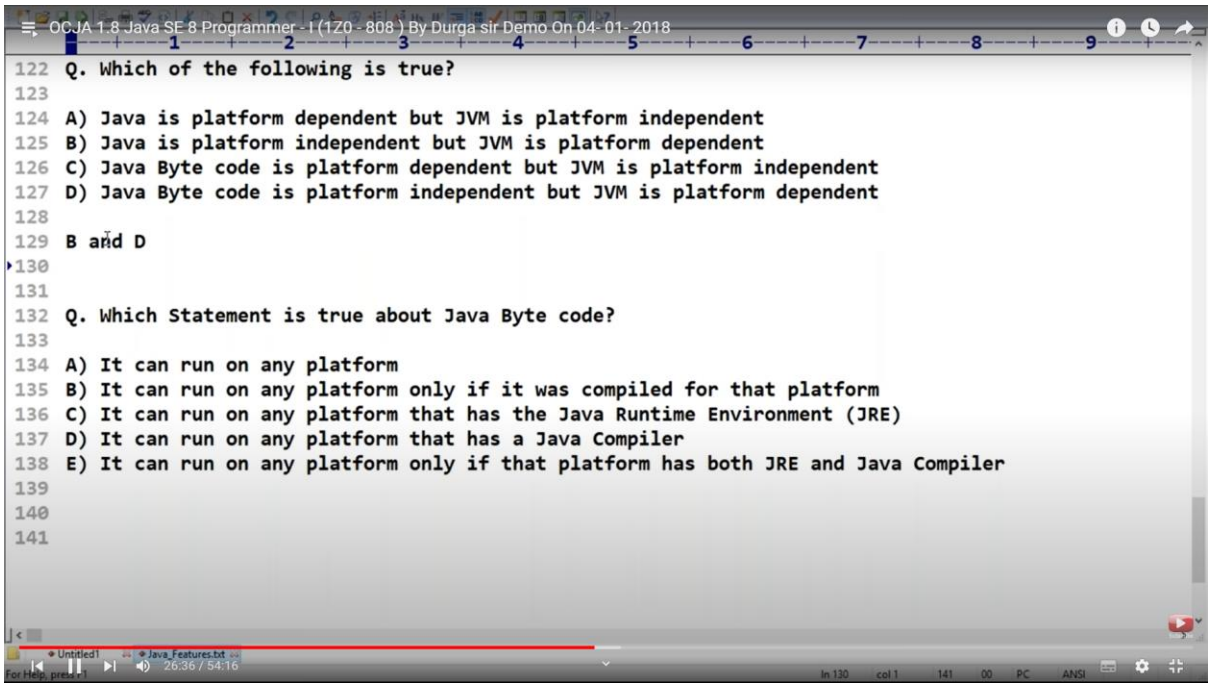

OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga sir Demo On 03-01-2018
Home
Tools
Sign In
OCJA
10) Compiled and Interpreted:
Java is both Compiled and Interpreted Programming language. First Java compiler compiles java code and generates machine independent Byte Code. At runtime JVM interprets this byte code into machine code and executes that machine code.
11) High Performance:
Java is relatively faster than traditional interpreted languages, since byte code is "close" to native code. But Java is still somewhat slower than C or C++.
12) Dynamic:
In the case of Java programs, all .class files won't be loaded at the beginning. At runtime if JVM required any class then only the corresponding .class file will be loaded(Dynamic Loading).The main advantage is program will always get latest version of .class file and memory utilization will be improved.
Export PDF
Adobe Export PDF
Convert PDF Files to Word or Excel Online
Select PDF File
The Features of Java.pdf
Convert to
Microsoft Word (*.docx)
Document Language:
English (U.S.) Change
Convert
Create PDF
Store and share files in the Document Cloud
Learn More

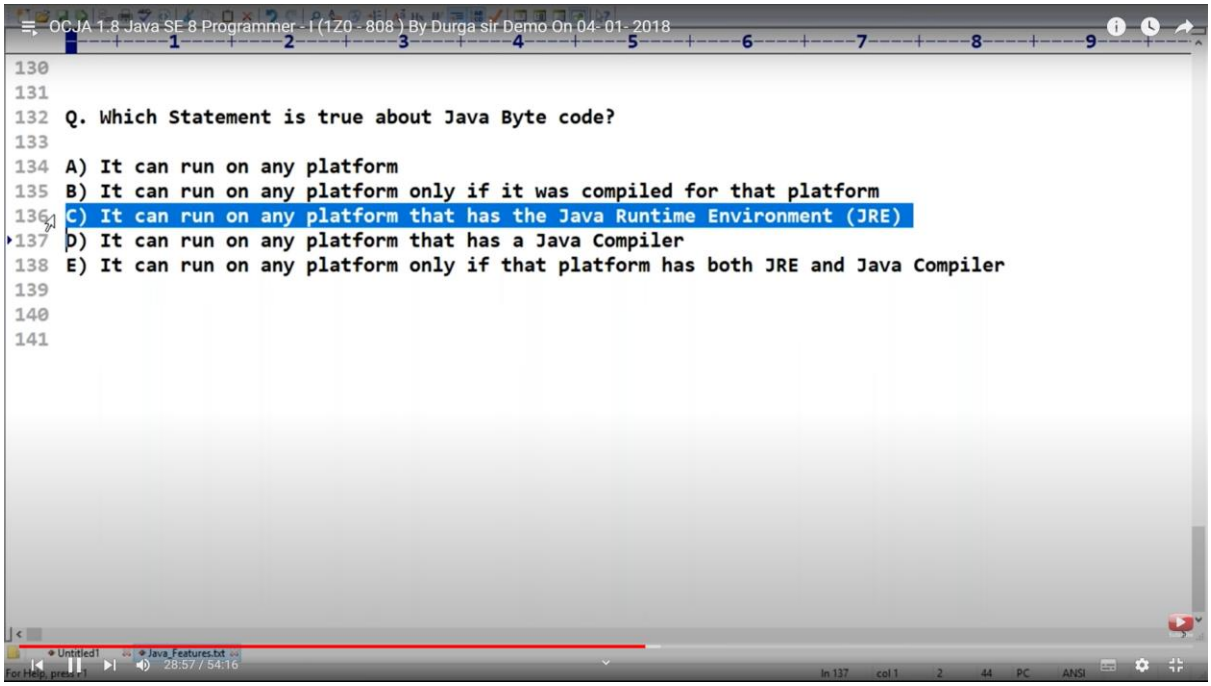

untitled - Paint
File Edit View Image Colors Help
Java Source File
Test.java
Java Compiler
javac Test.java
Byte Code
Test.class
JVM for Windows
JVM for Linux
JVM for MAC
Windows Machine
Linux Machine
Mac Machine
For Help, click Help Topics on the Help Menu.
Type here to search
ENG
9:48 PM
01/04/2018


OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga sir Demo On 04- 01- 2018
JDK vs JRE vs JVM:
JDK(Java Development Kit):
To develop and run java applications
JRE(Java Runtime Environment):
To run java applications
JVM(Java Virtual Machine):
is an interpreter to run program line by line
Untitled1
Java_Features.txt
20:41 / 54:16

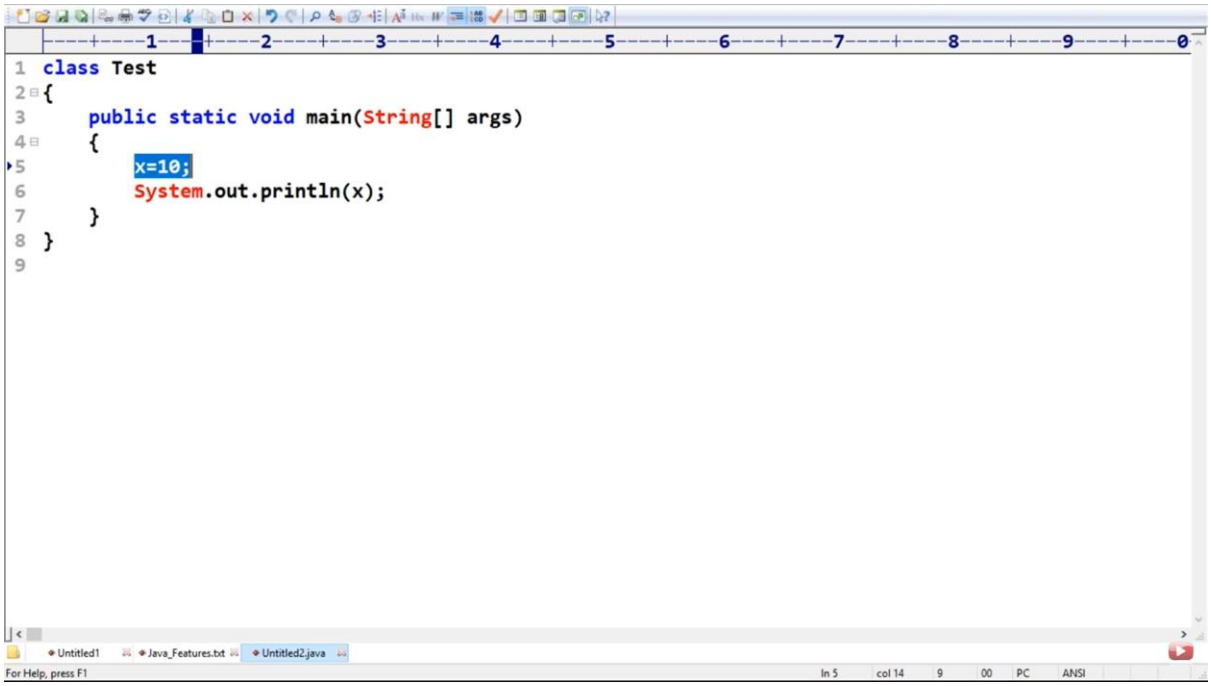
All development tools in bin folder compiler and debugger part of Development tools

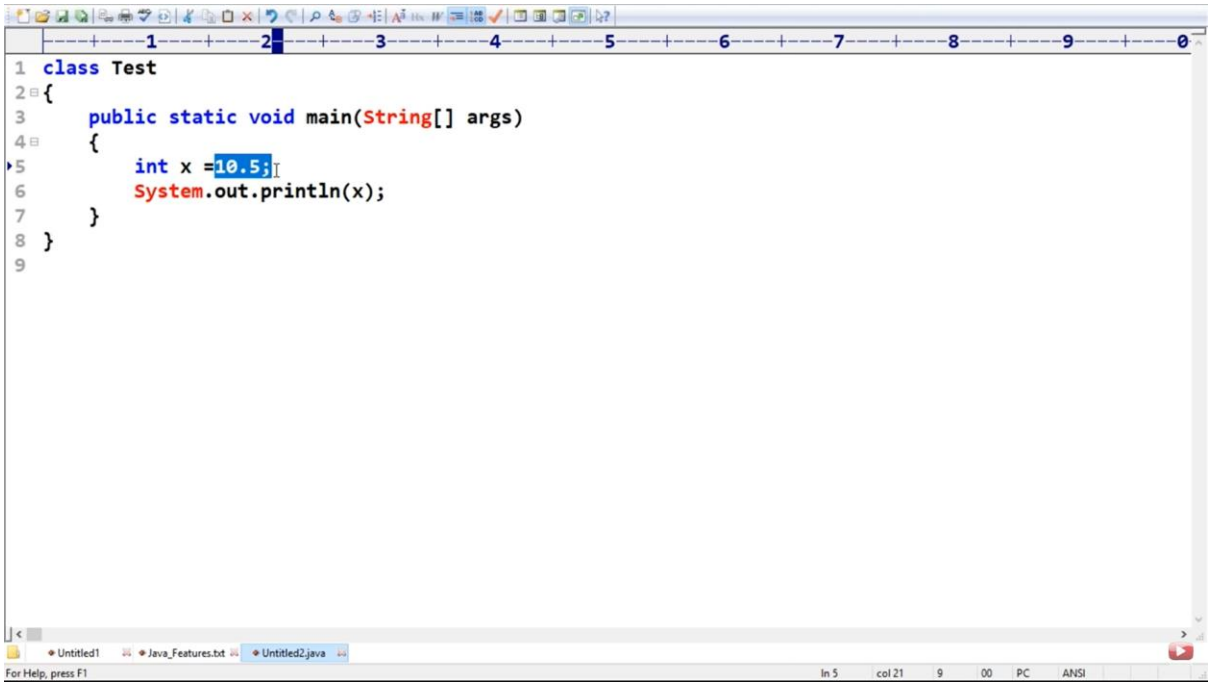




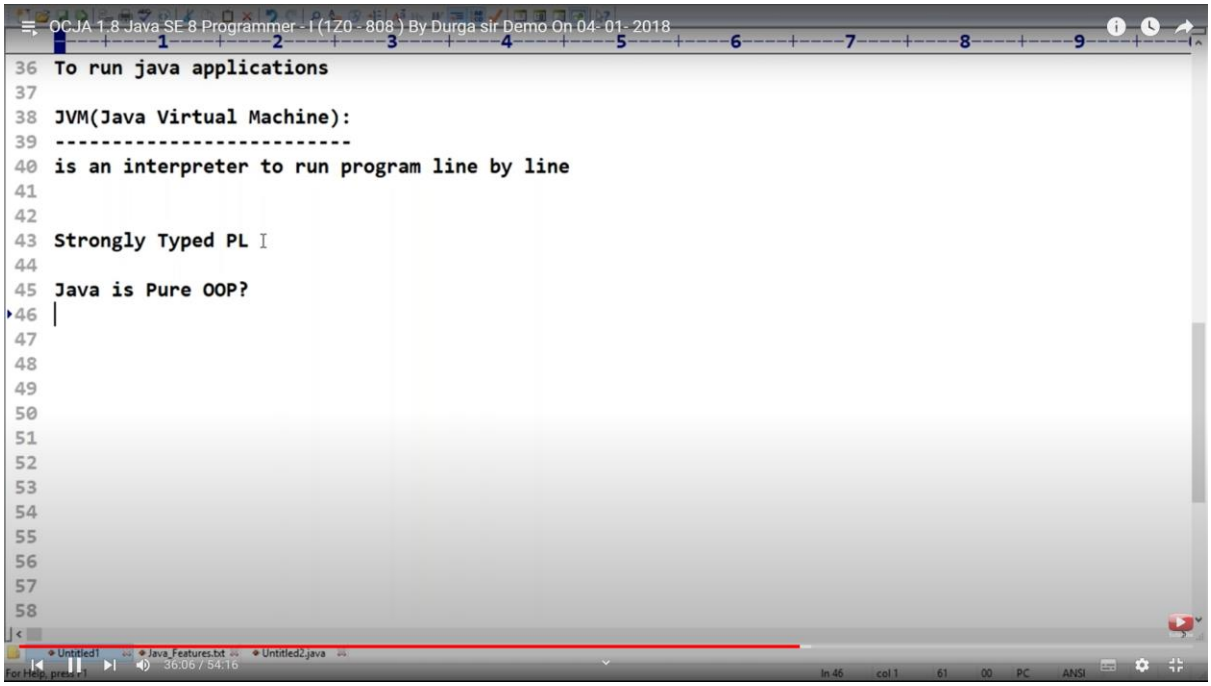


Invalid type missing

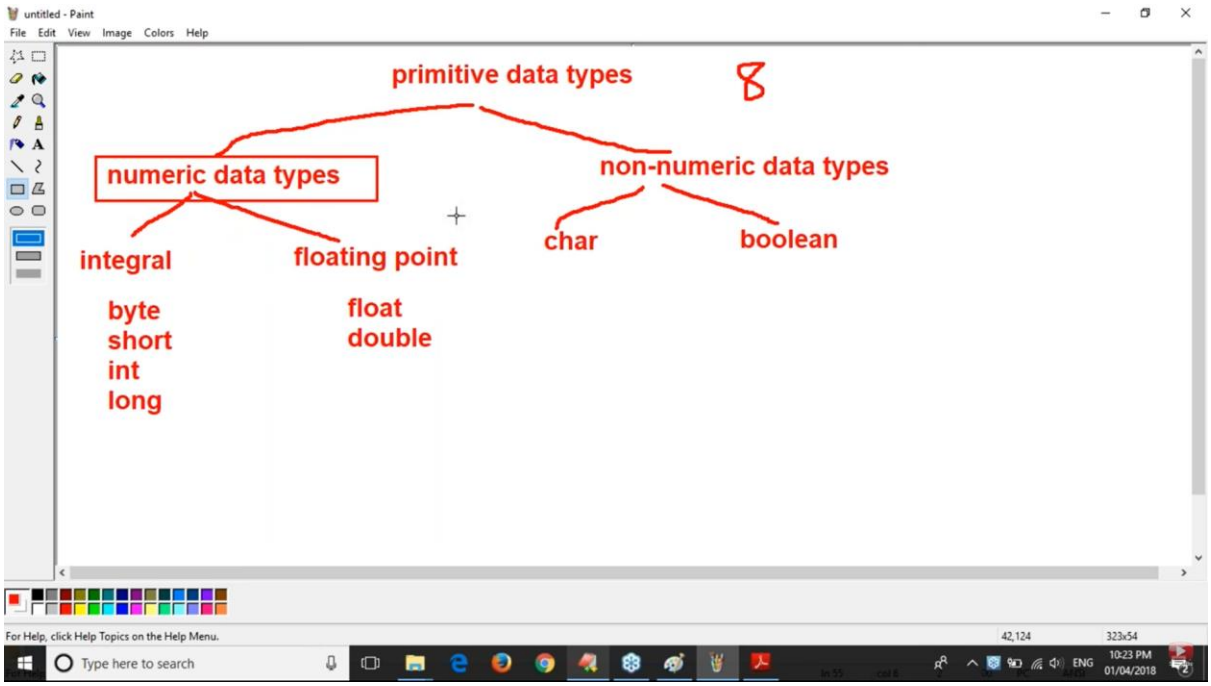


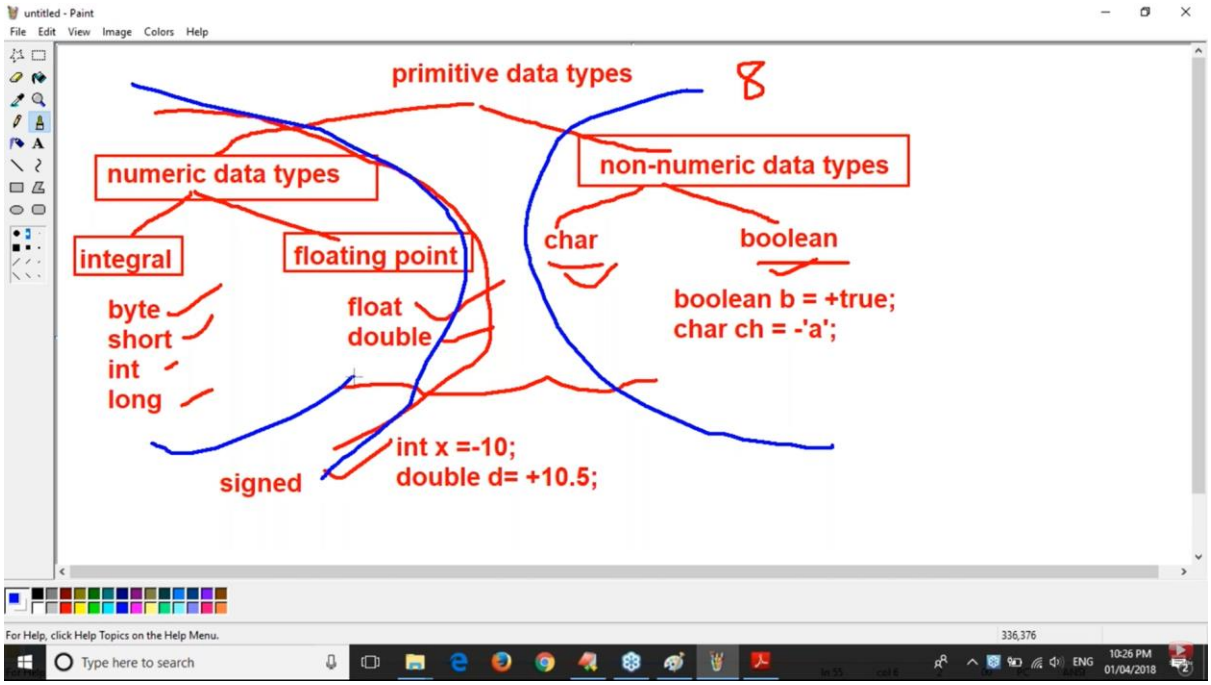


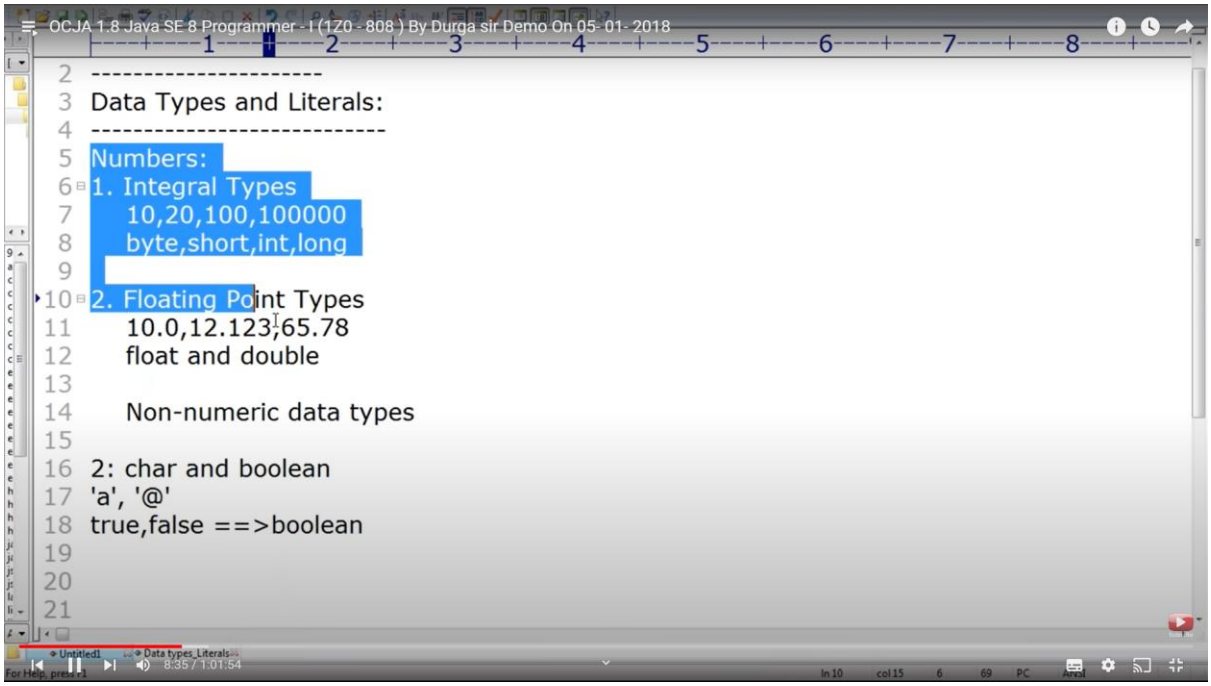
```
1  class Test
2  {
3      public static void main(String[] args)
4      {
5          int x =10.5;
6          System.out.println(x);
7      }
8  }
9
```



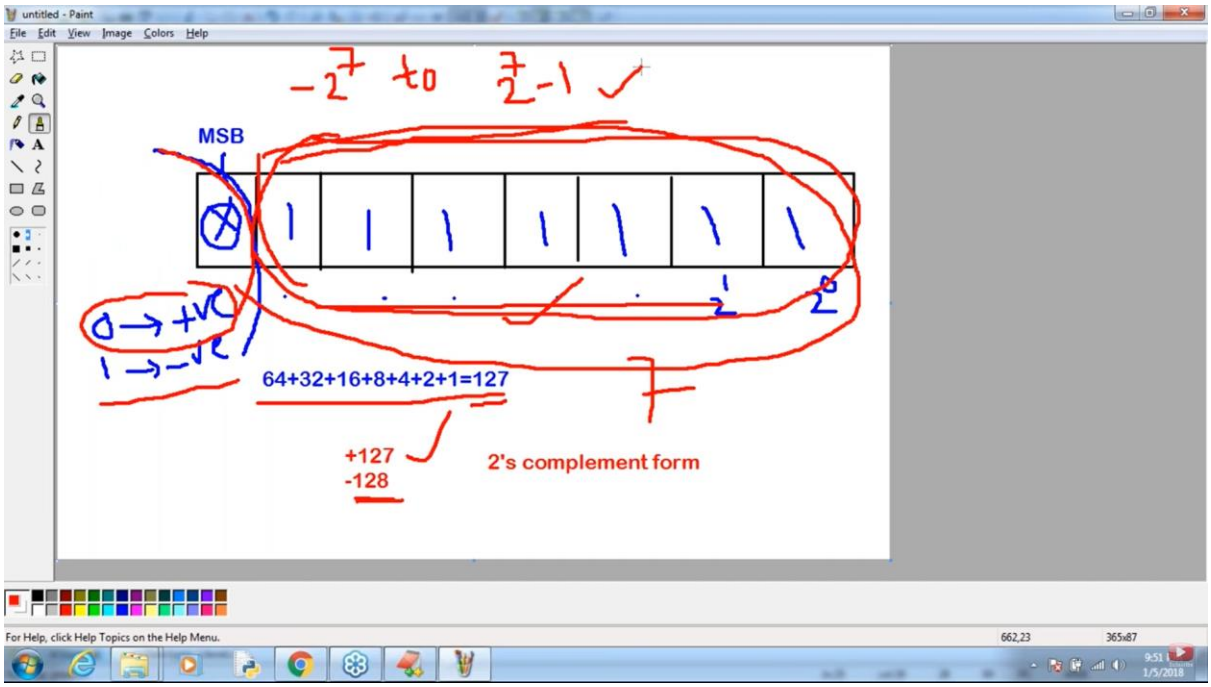

```
36 To run java applications
37
38 JVM(Java Virtual Machine):
39 -------------------------
40 is an interpreter to run program line by line
41
42
43 Strongly Typed PL
44
45 Java is Pure OOP?
46
47
48
49
50
51
52
53
54
55
56
57
58
```

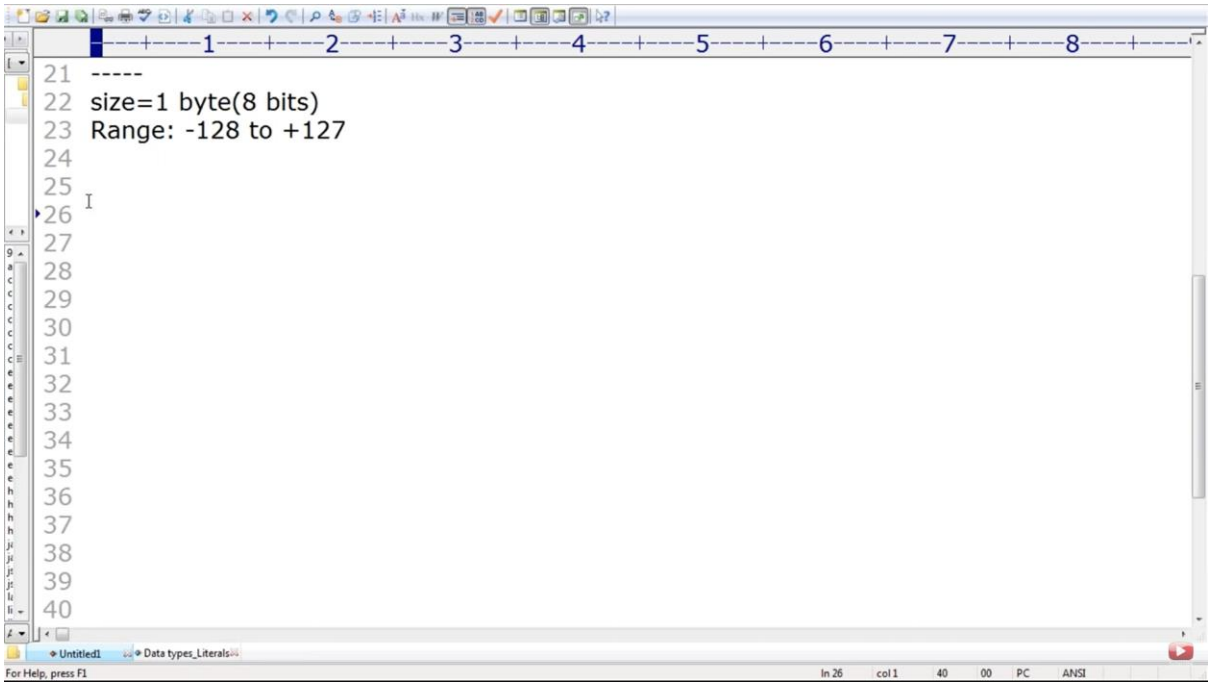

untitled - Paint
File Edit View Image Colors Help
primitive data types
numeric data types
non-numeric data types
integral
floating point
char
boolean
byte
short
int
long
float
double
For Help, click Help Topics on the Help Menu.
Type here to search
ENG
10:23 PM
01/04/2018

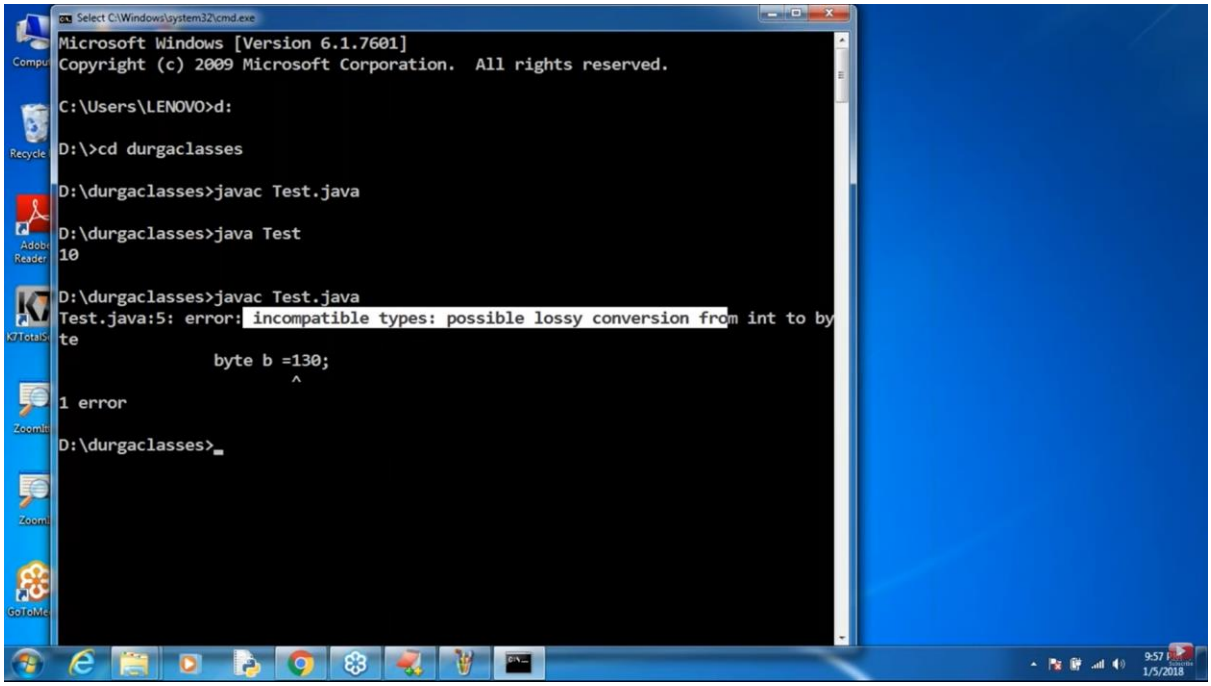

untitled - Paint
File Edit View Image Colors Help
primitive data types
numeric data types
non-numeric data types
integral
floating point
char
boolean
byte
short
int
long
float
double
boolean b = +true;
char ch = -'a';
int x =-10;
double d= +10.5;
signed
For Help, click Help Topics on the Help Menu.
Type here to search
ENG
10:26 PM
01/04/2018



Positive nos are directly stored in memory whereas negative nos are rep in 2's compliment so we can store upto 128

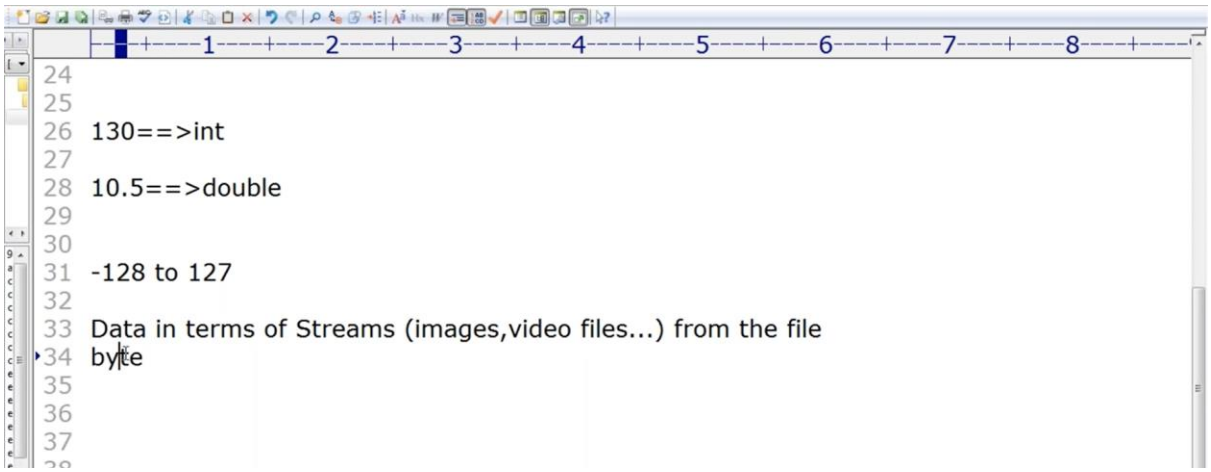






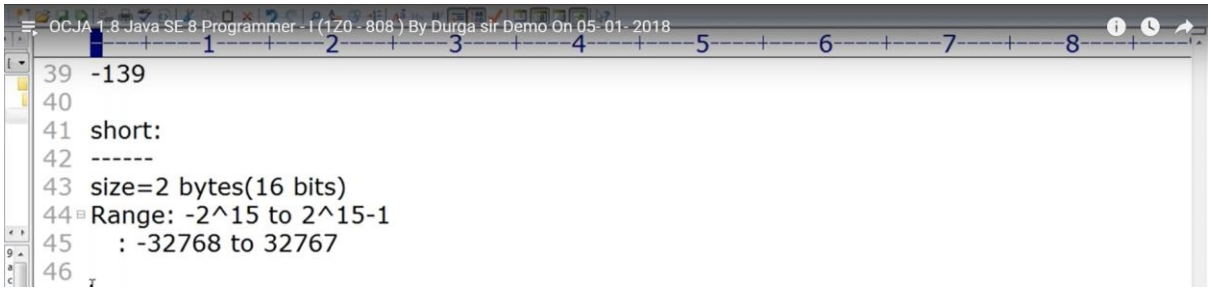
By default every integeral type is int and every floating point are double

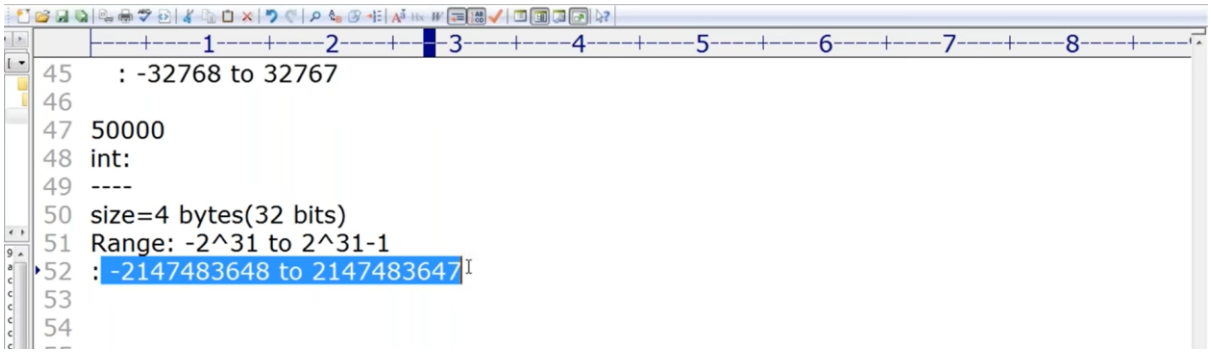
Wherever files or streams are rep in bytes



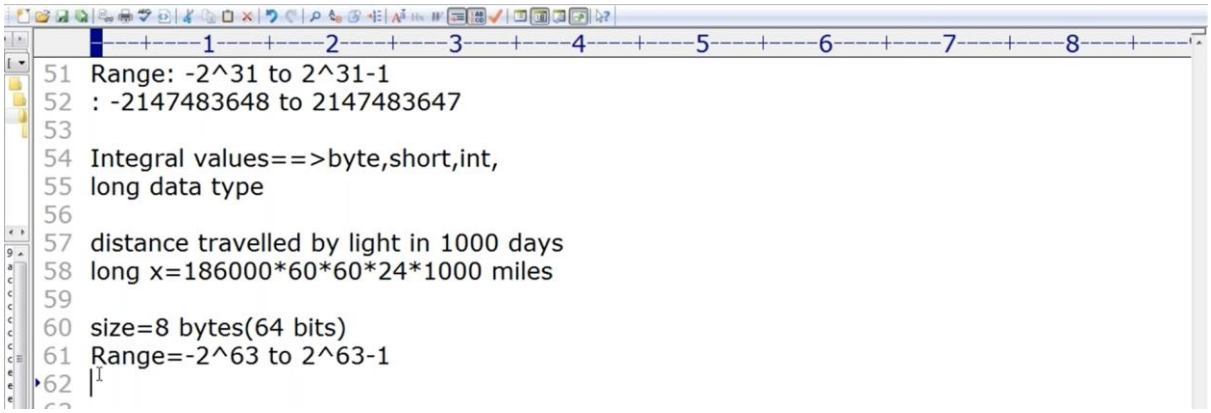
Rarely used data type is Short – 2bytes

Olden days we were using 16 bit processor, now java 9 using 64 bits . we are not using short anymore. But can be used

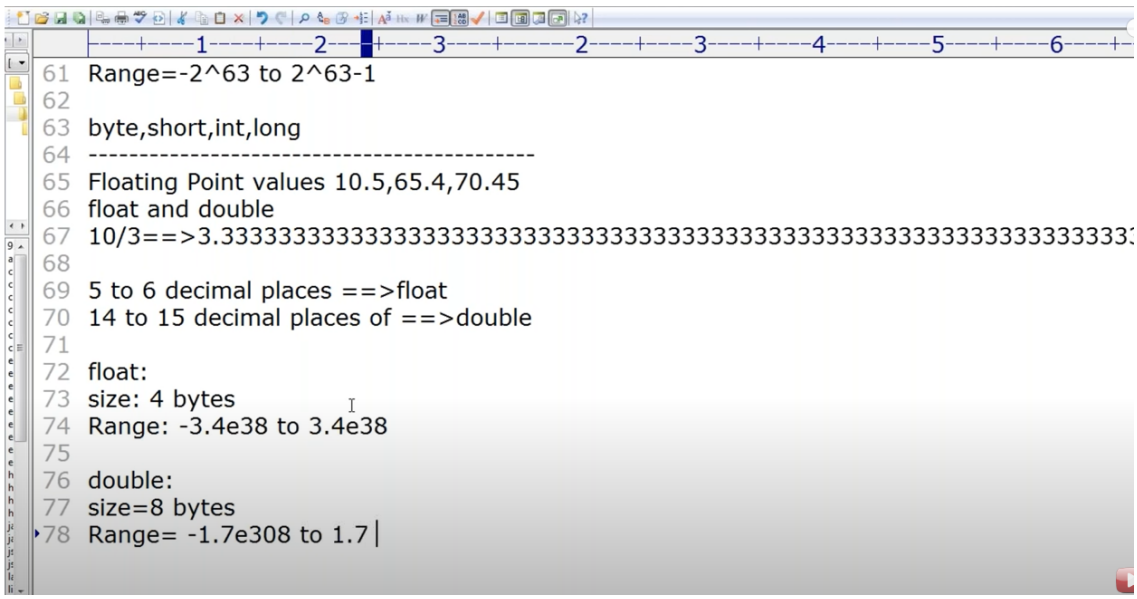




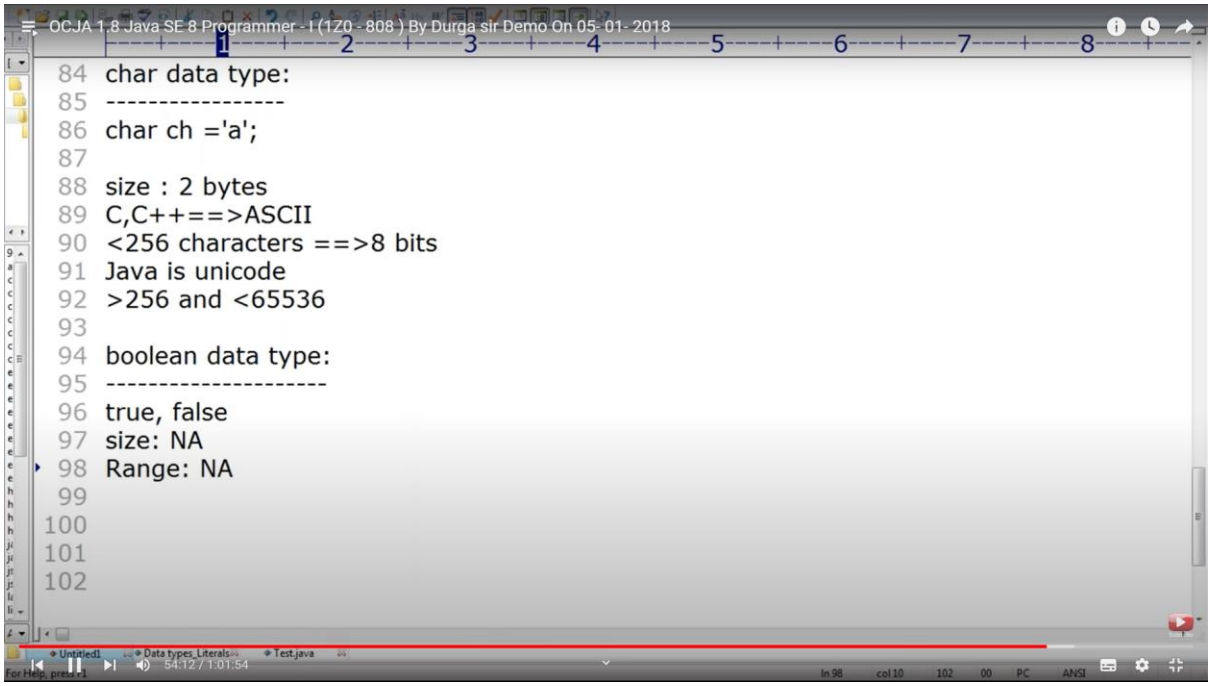
```
45    : -32768 to 32767
46
47  50000
48  int:
49  ----
50  size=4 bytes(32 bits)
51  Range: -2^31 to 2^31-1
52  : -2147483648 to 2147483647
53
54
```

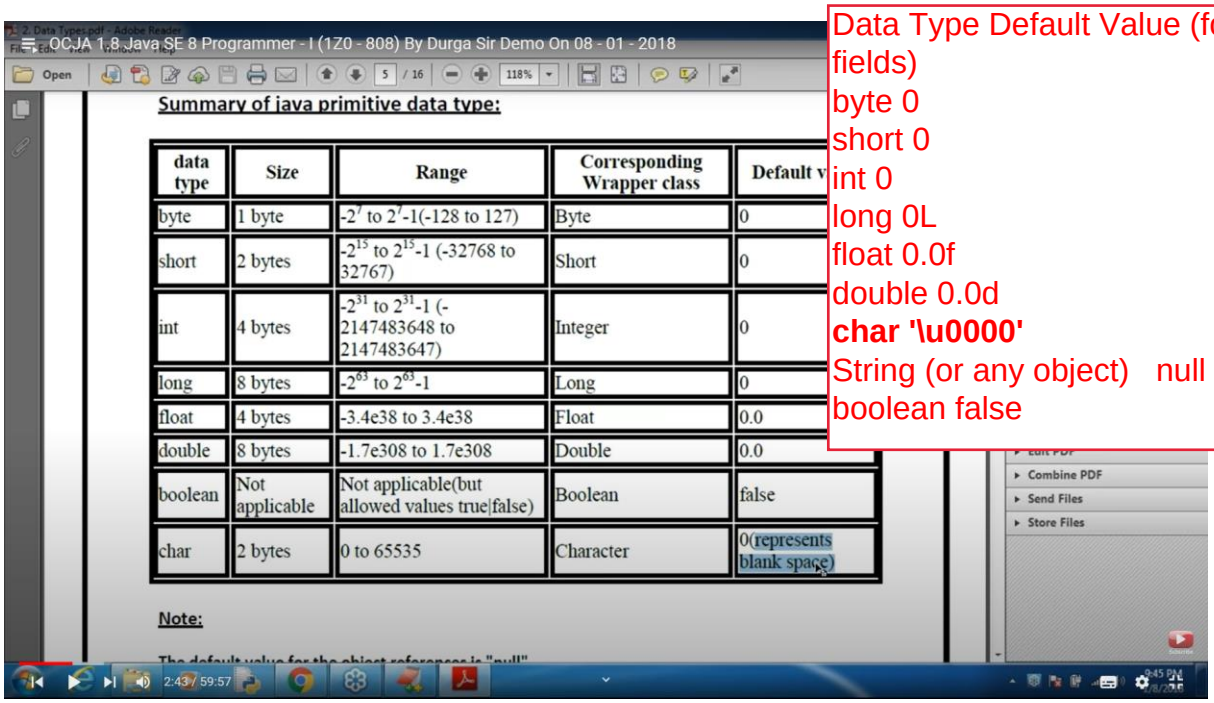


```
51  Range: -2^31 to 2^31-1
52  : -2147483648 to 2147483647
53
54  Integral values==>byte,short,int,
55  long data type
56
57  distance travelled by light in 1000 days
58  long x=186000*60*60*24*1000 miles
59
60  size=8 bytes(64 bits)
61  Range=-2^63 to 2^63-1
62
```



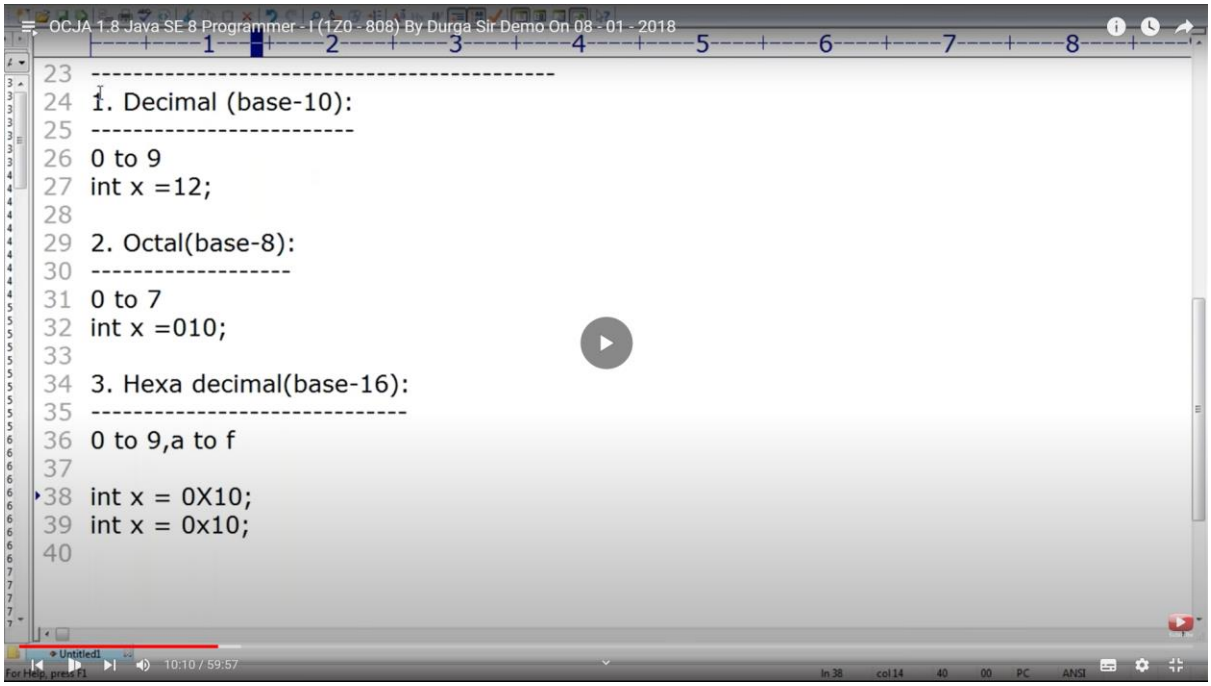
```
61  Range=-2^63 to 2^63-1
62
63  byte,short,int,long
64  ----------------------------------------
65  Floating Point values 10.5,65.4,70.45
66  float and double
67  10/3==>3.333333333333333333333333333333333333333333333333333333333333333333333333333333
68
69  5 to 6 decimal places ==>float
70  14 to 15 decimal places of ==>double
71
72  float:
73  size: 4 bytes
74  Range: -3.4e38 to 3.4e38
75
76  double:
77  size=8 bytes
78  Range= -1.7e308 to 1.7
```



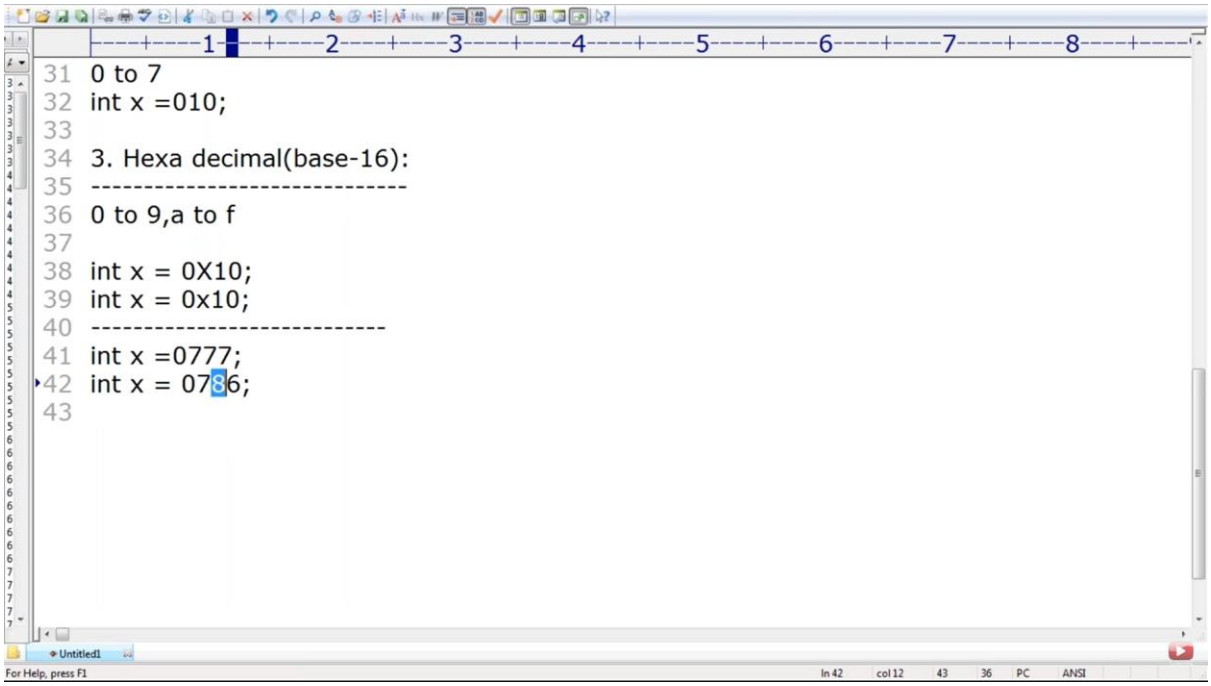


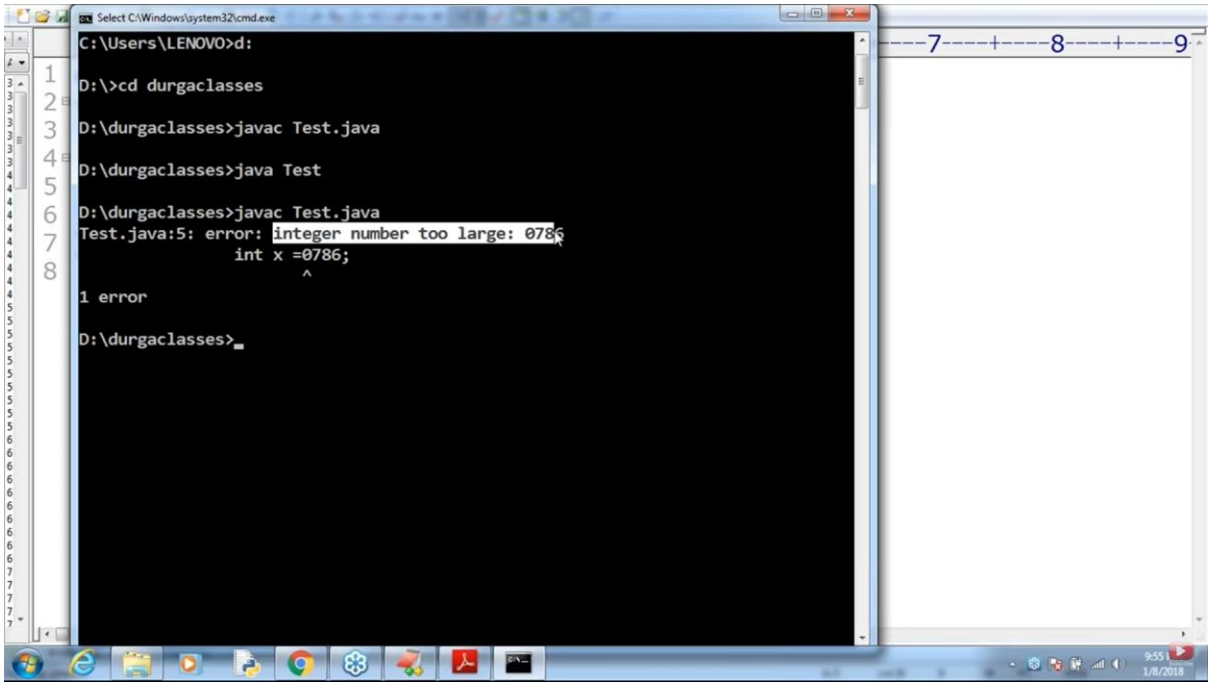
data type	Size	Range	Corresponding Wrapper class	Default v
byte	1 byte	-2^7 to 2^7-1(-128 to 127)	Byte	0
short	2 bytes	-2^{15} to 2^{15}-1 (-32768 to 32767)	Short	0
int	4 bytes	-2^{31} to 2^{31}-1 (-2147483648 to 2147483647)	Integer	0
long	8 bytes	-2^{63} to 2^{63}-1	Long	0
float	4 bytes	-3.4e38 to 3.4e38	Float	0.0
double	8 bytes	-1.7e308 to 1.7e308	Double	0.0
boolean	Not applicable	Not applicable(but allowed values true|false)	Boolean	false
char	2 bytes	0 to 65535	Character	0(represents blank space)

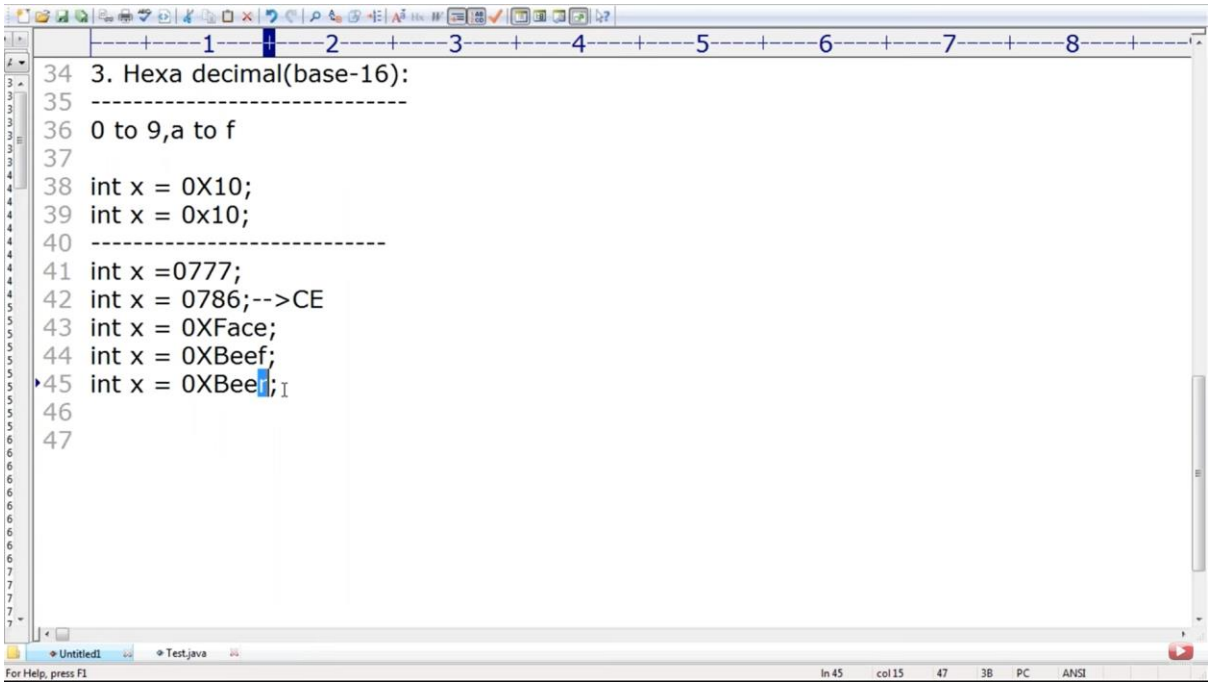
Data Type Default Value (for fields)
byte 0
short 0
int 0
long 0L
float 0.0f
double 0.0d
char '\u0000'
String (or any object) null
boolean false



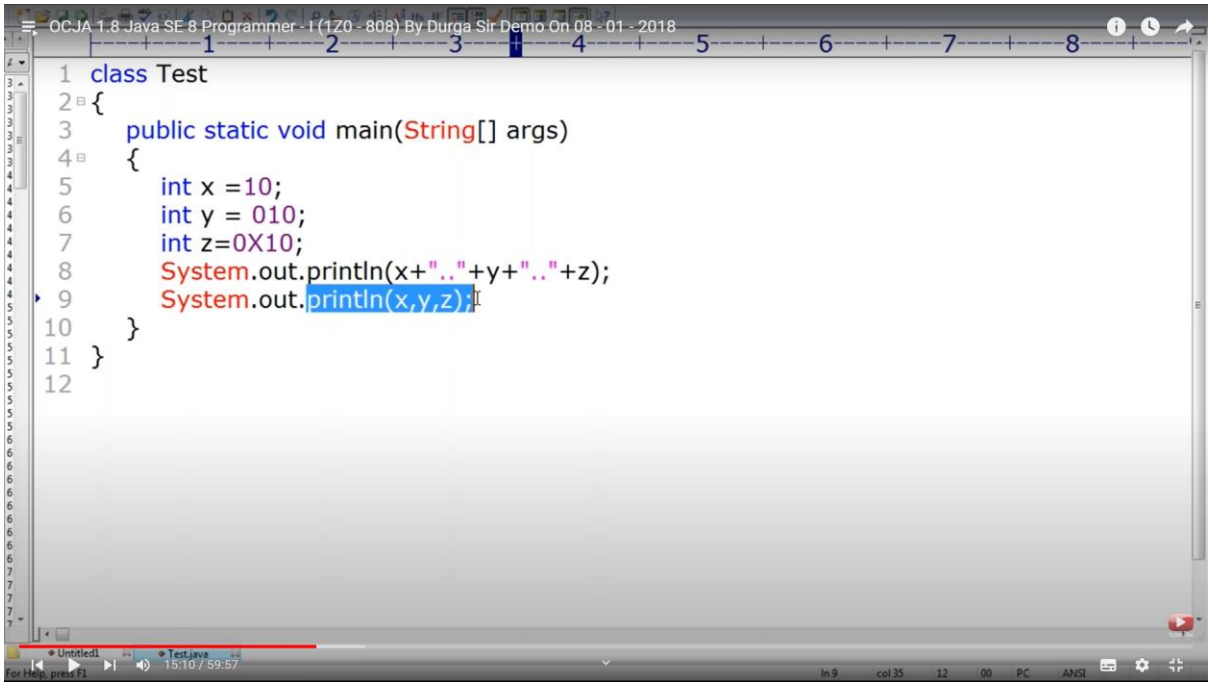
In octal digit 8 is not allowed

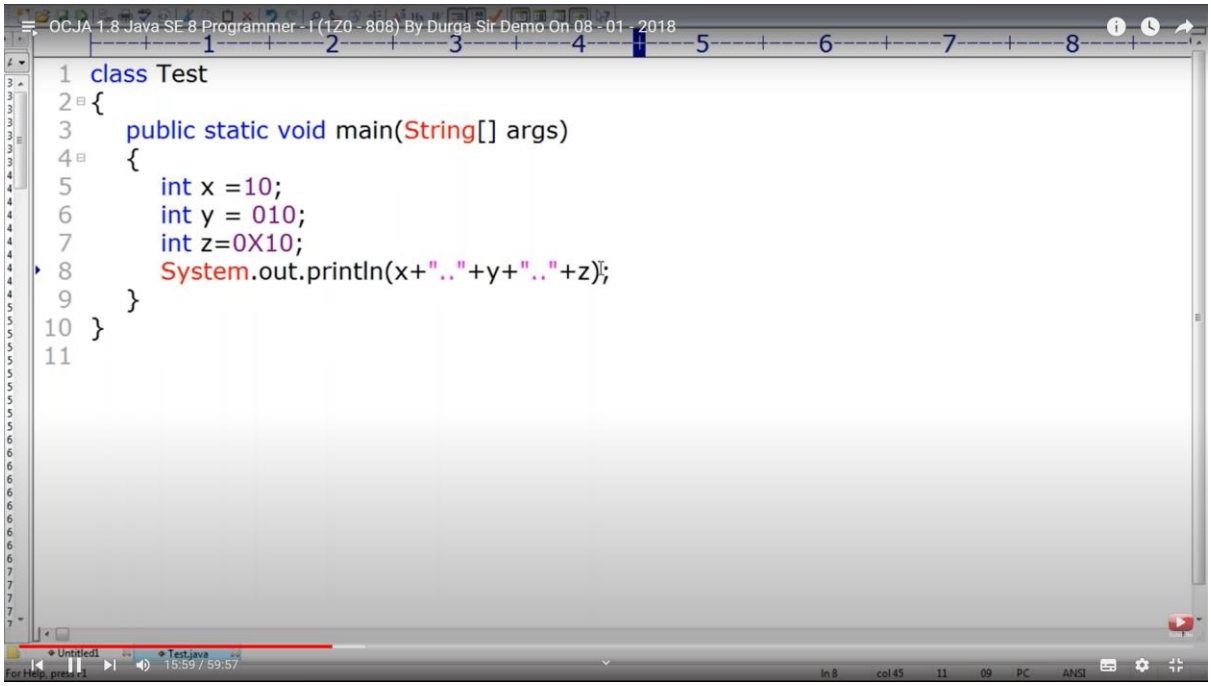



Select C:\Windows\system32\cmd.exe
C:\Users\LENOVO>d:
D:\>cd durgaclasses
D:\durgaclasses>javac Test.java
D:\durgaclasses>java Test
D:\durgaclasses>javac Test.java
Test.java:5: error: integer number too large: 0786
int x =0786;
^
1 error
D:\durgaclasses>

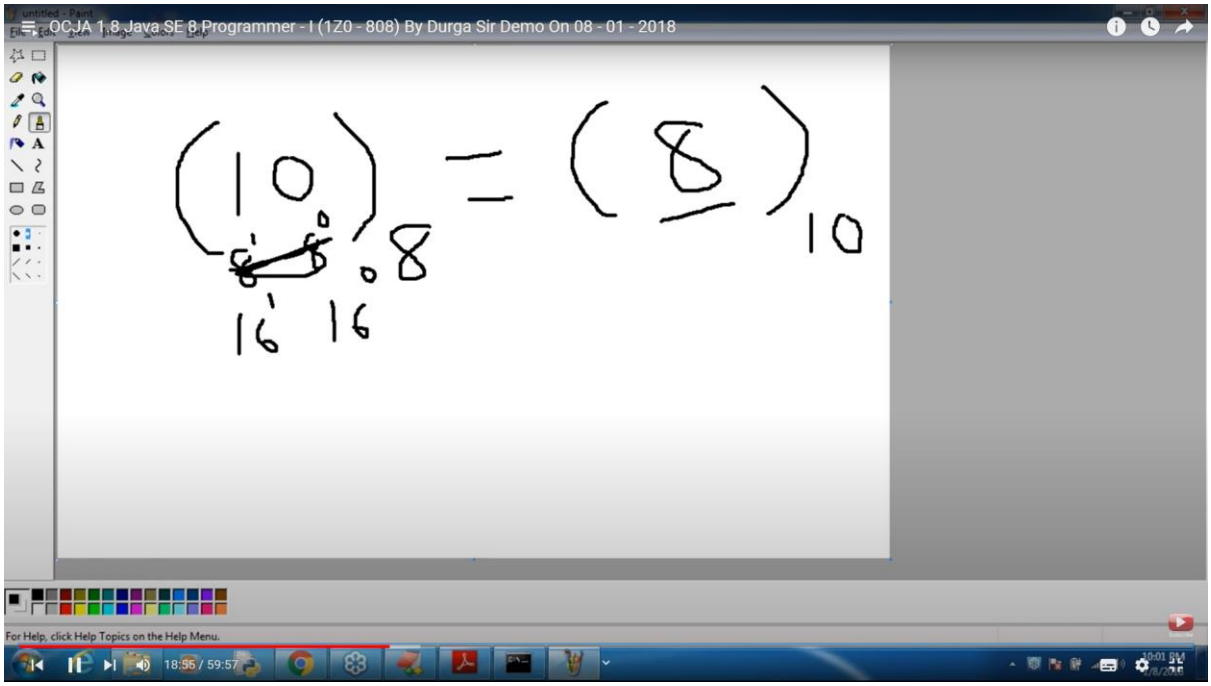

3. Hexa decimal(base-16):
0 to 9,a to f
int x = 0X10;
int x = 0x10;
int x =0777;
int x = 0786;-->CE
int x = 0XFace;
int x = 0XBeef;
int x = 0XBeef;
Untitled1
Test.java
For Help, press F1

This type or multiple arugumnts in line 9 are not allower, we can pass only 1 variable like line 8

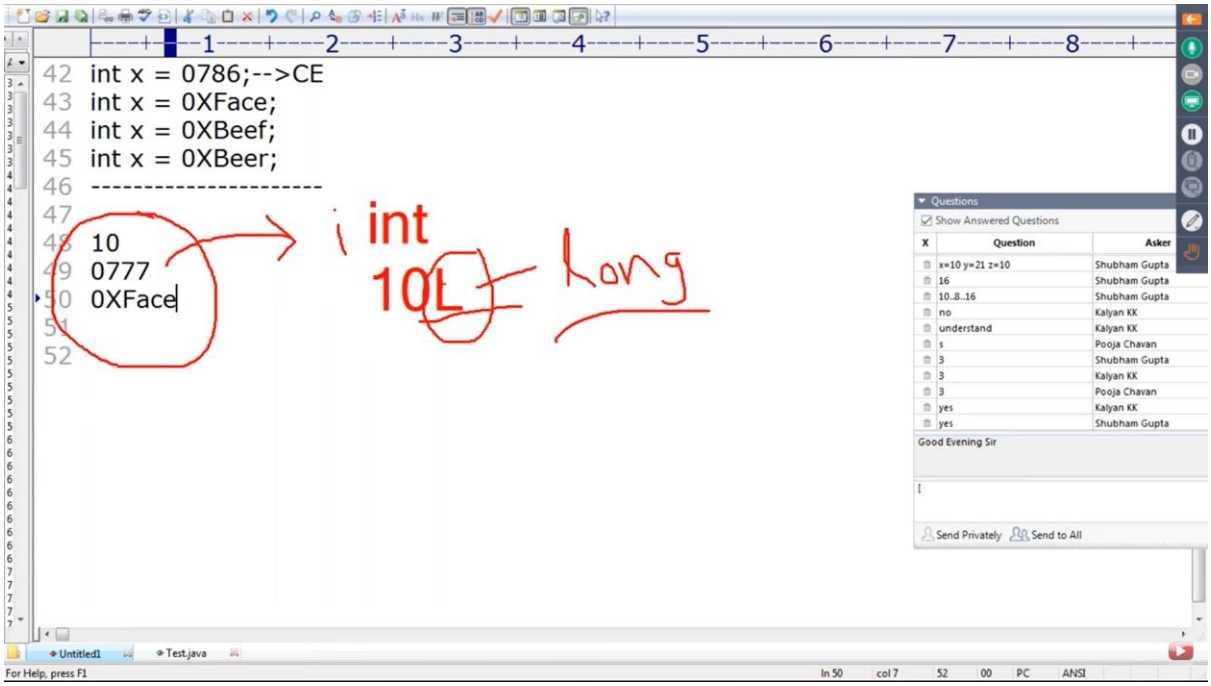


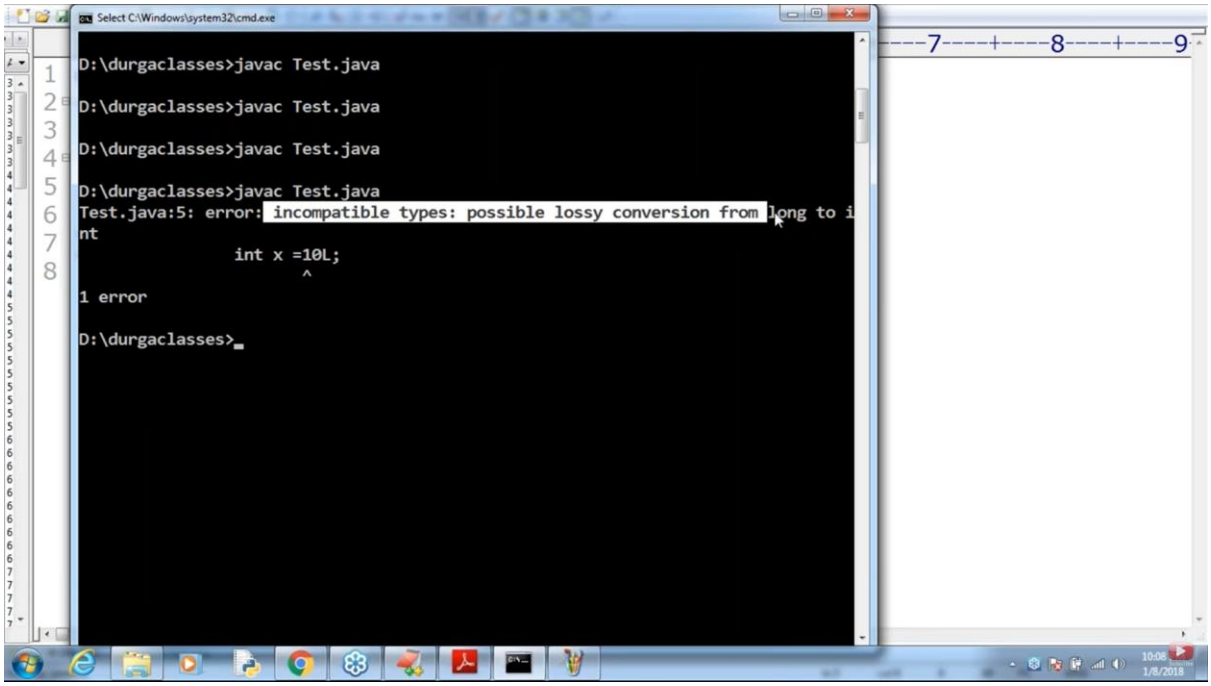


System will consider only in decimal value

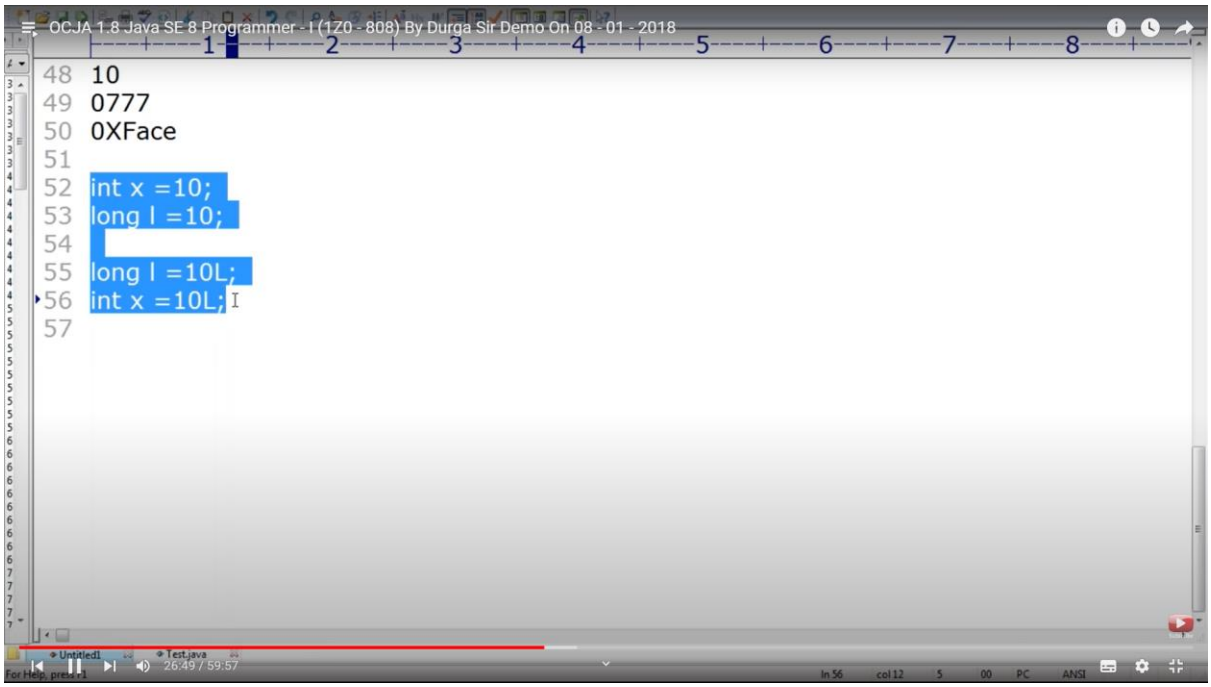

OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 08 - 01 - 2018
(10) = (8)10
8
16 16
For Help, click Help Topics on the Help Menu.
18:55 / 59:57


1 error
D:\durgaclasses>javac Test.java
D:\durgaclasses>java Test
10..8..16

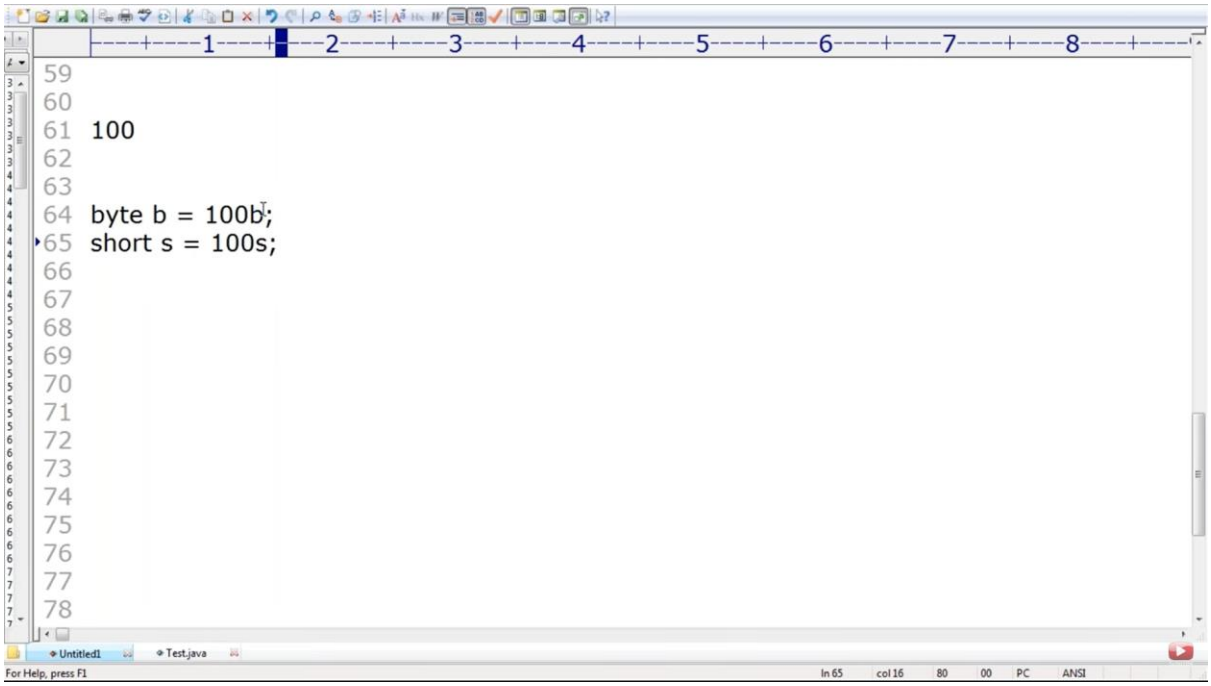

int x = 0786;-->CE
int x = 0XFace;
int x = 0XBeef;
int x = 0XBeer;
10
0777
0XFace
int
10L
long
Questions
Show Answered Questions
Question
Asker
x=10 y=21 z=10
Shubham Gupta
16
Shubham Gupta
10..8..16
Shubham Gupta
no
Kalyan KK
understand
Kalyan KK
s
Pooja Chavan
3
Shubham Gupta
3
Kalyan KK
3
Pooja Chavan
yes
Kalyan KK
yes
Shubham Gupta
Good Evening Sir
Send Privately
Send to All
Untitled1
Test.java
For Help, press F1

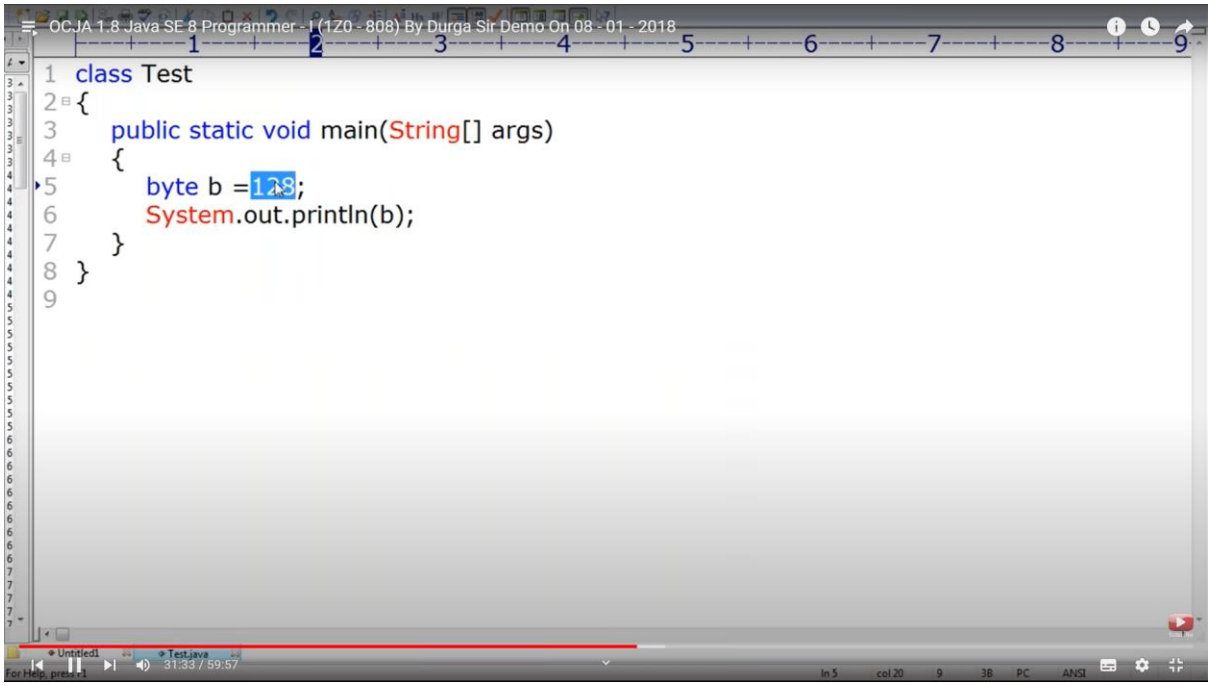


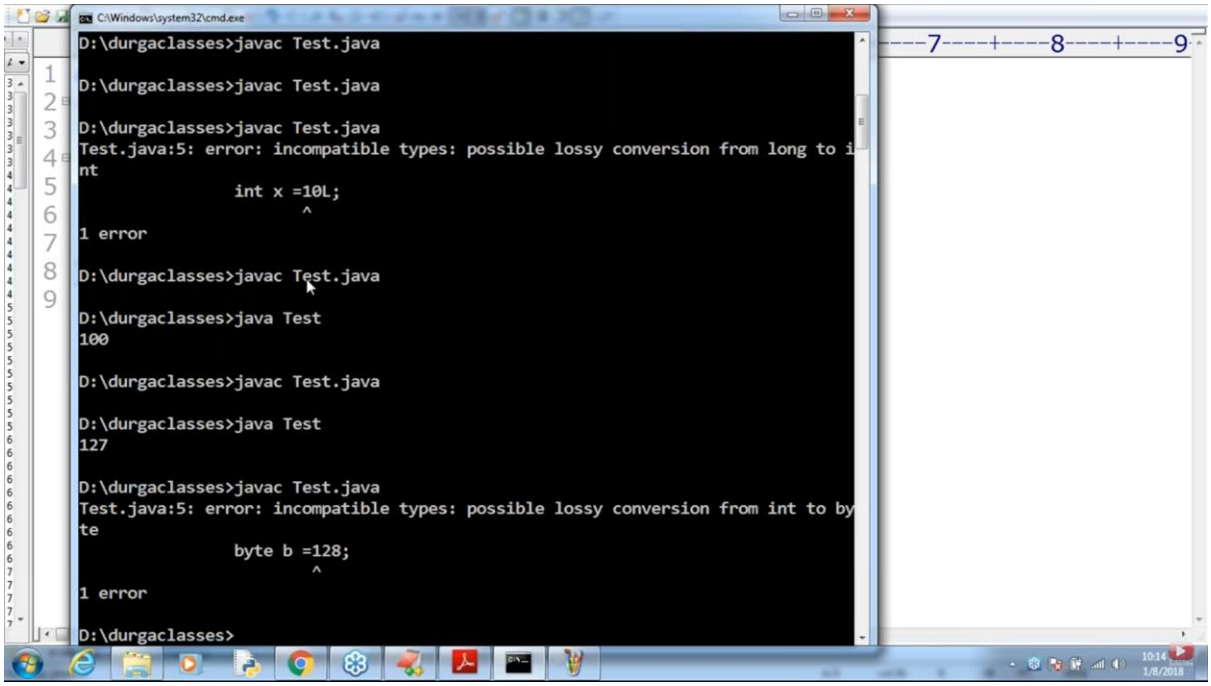
By Default every integral (octal, decimal and Hexa) is of int type. If you want to specify as Long suffix with L

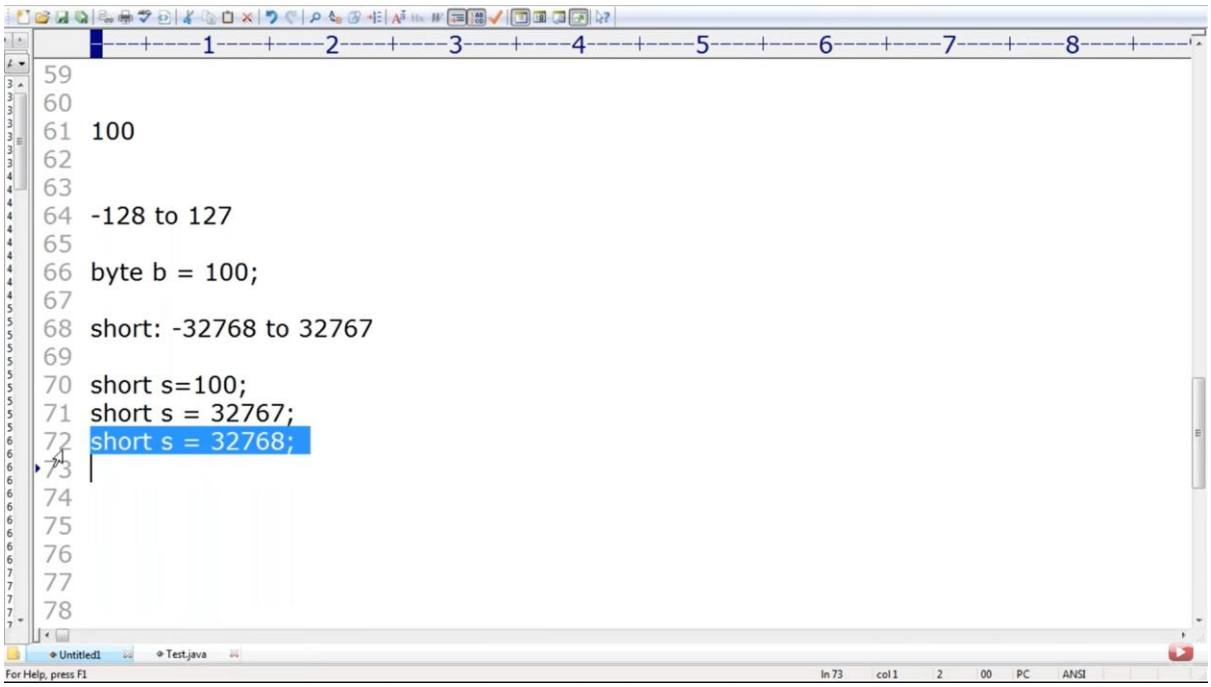


The below is wrong. Cant explicitly specify byte or short

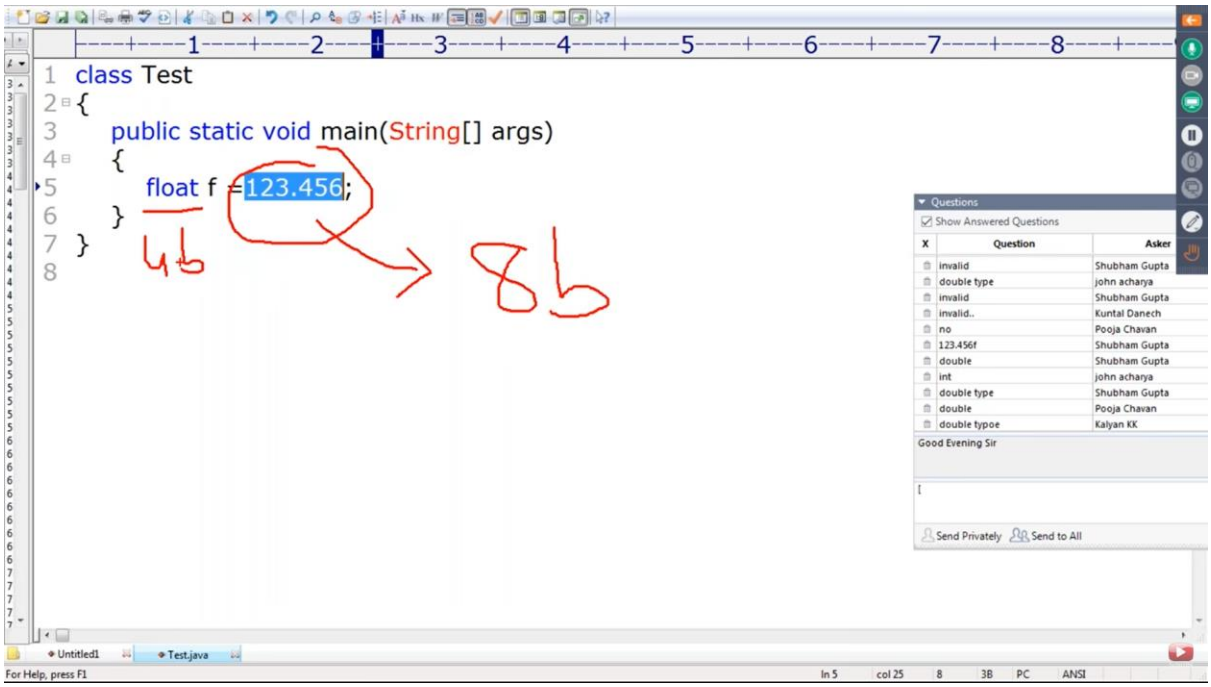

100
byte b = 100b;
short s = 100s;
Untitled1
Test.java
For Help, press F1
ln 65
col 16


OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 08 - 01 - 2018
class Test
{
public static void main(String[] args)
{
byte b =128;
System.out.println(b);
}
}
Untitled1
Test.java
31:33 / 59:57
ln 5
col 20


C:\Windows\system32\cmd.exe
D:\durgaclasses>javac Test.java
D:\durgaclasses>javac Test.java
D:\durgaclasses>javac Test.java
Test.java:5: error: incompatible types: possible lossy conversion from long to i
nt
int x =10L;
^
1 error
D:\durgaclasses>javac Test.java
D:\durgaclasses>java Test
100
D:\durgaclasses>javac Test.java
D:\durgaclasses>java Test
127
D:\durgaclasses>javac Test.java
Test.java:5: error: incompatible types: possible lossy conversion from int to by
te
byte b =128;
^
1 error
D:\durgaclasses>

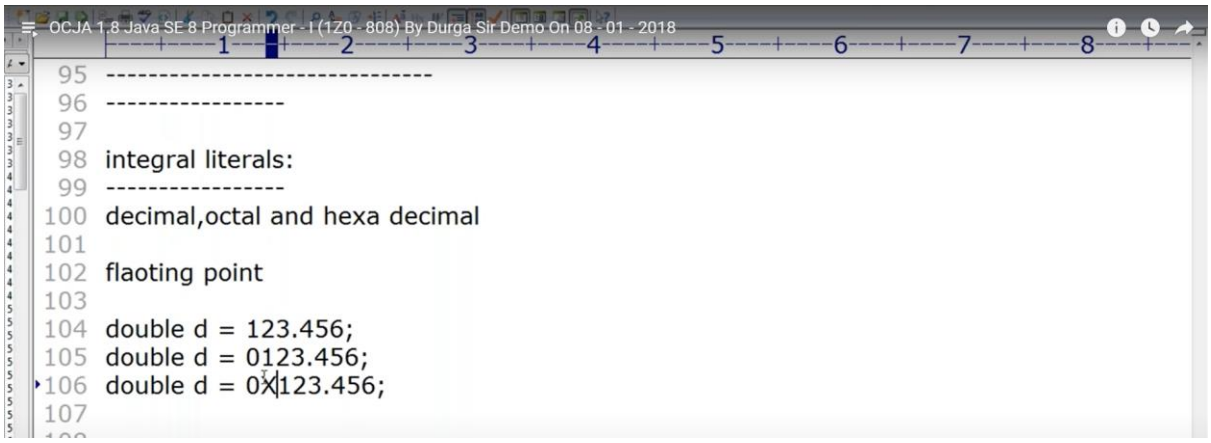

100
-128 to 127
byte b = 100;
short: -32768 to 32767
short s=100;
short s = 32767;
short s = 32768;
Untitled1
Test.java
For Help, press F1

By default floating point literals are of double type cant assign to float



Suffix with small f or capital F

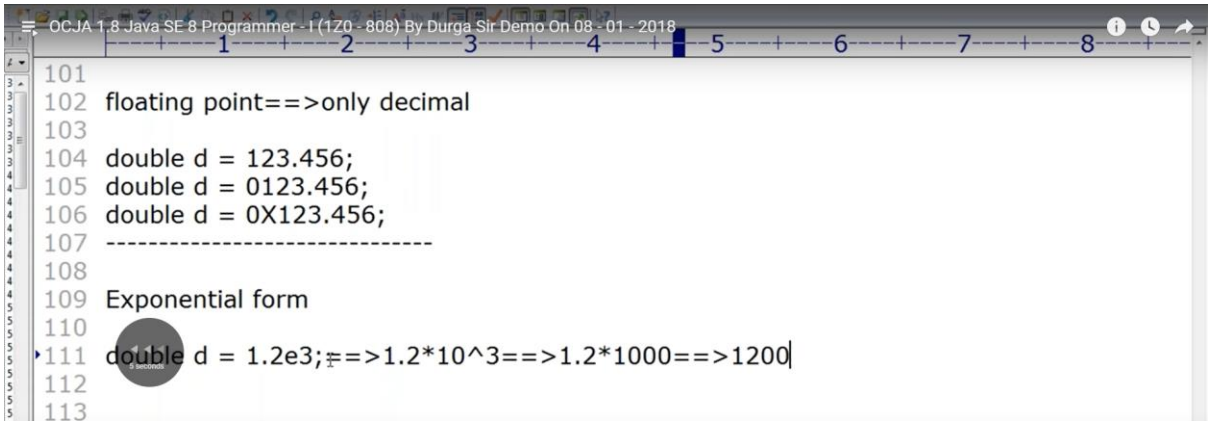
Line 90 alone error



Floating Point only supports decimal.

Double d = 0123.456 will be considered as floating point and ignore the 0 on printing the 0 will be truncated .So its valid.

Whereas 0x → x is an invalid character and will throw an error.

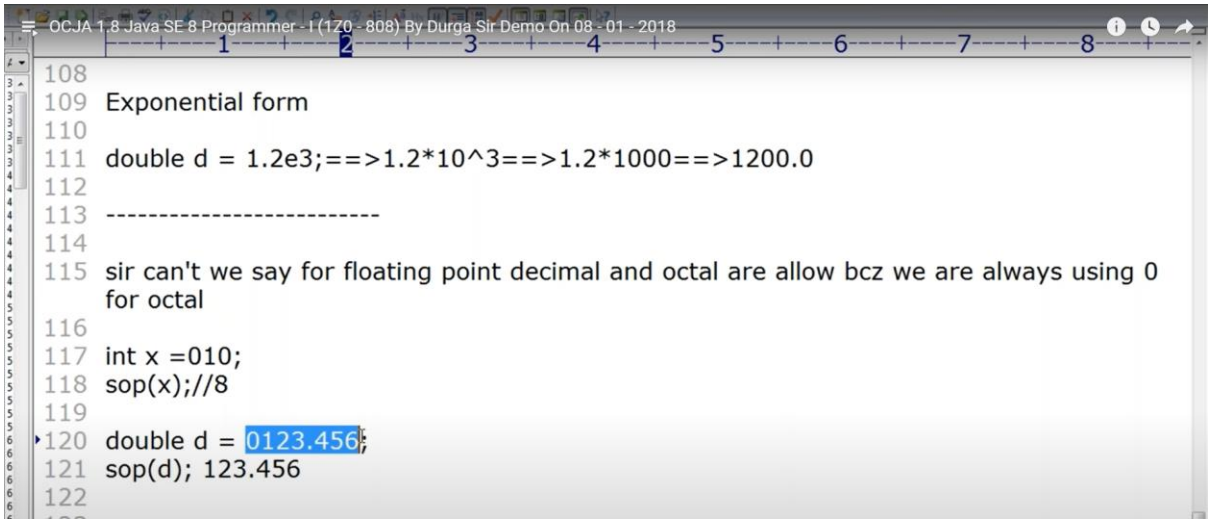


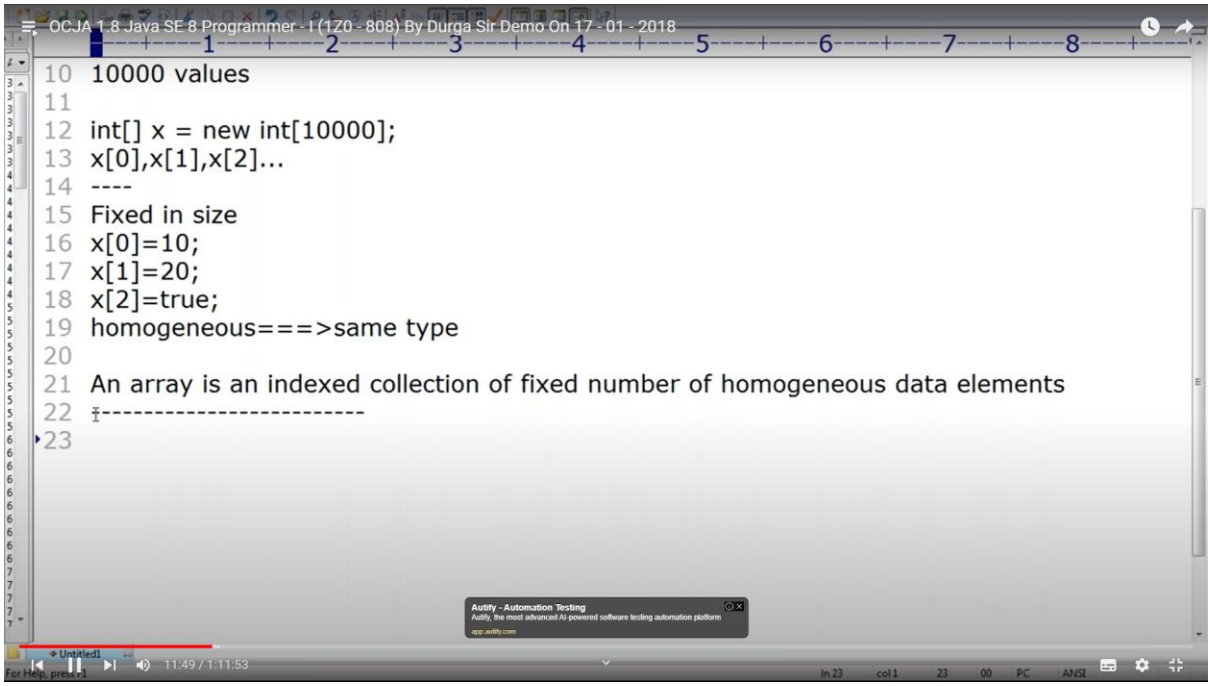
```
D:\durgaclasses>javac Test.java

D:\durgaclasses>java Test
1200.0
```

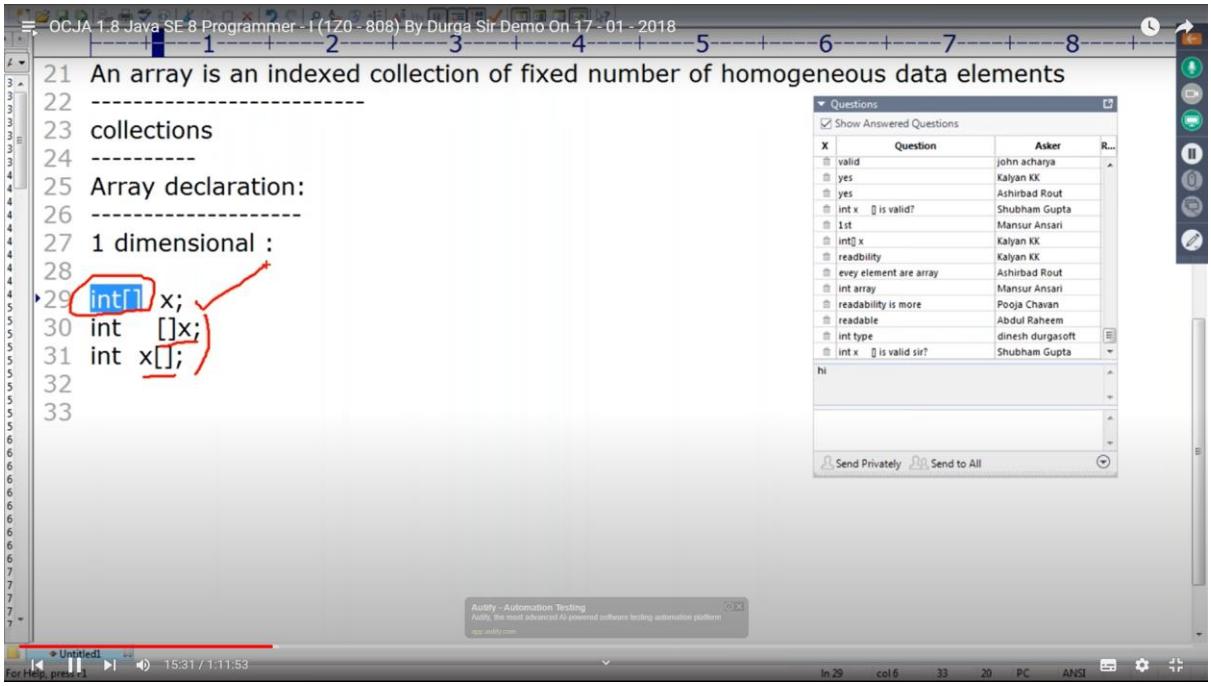
```
class Test
{
    public static void main(String[] args)
    {
        float f = 1.2e3;
        System.out.println(f);
    }
}
```

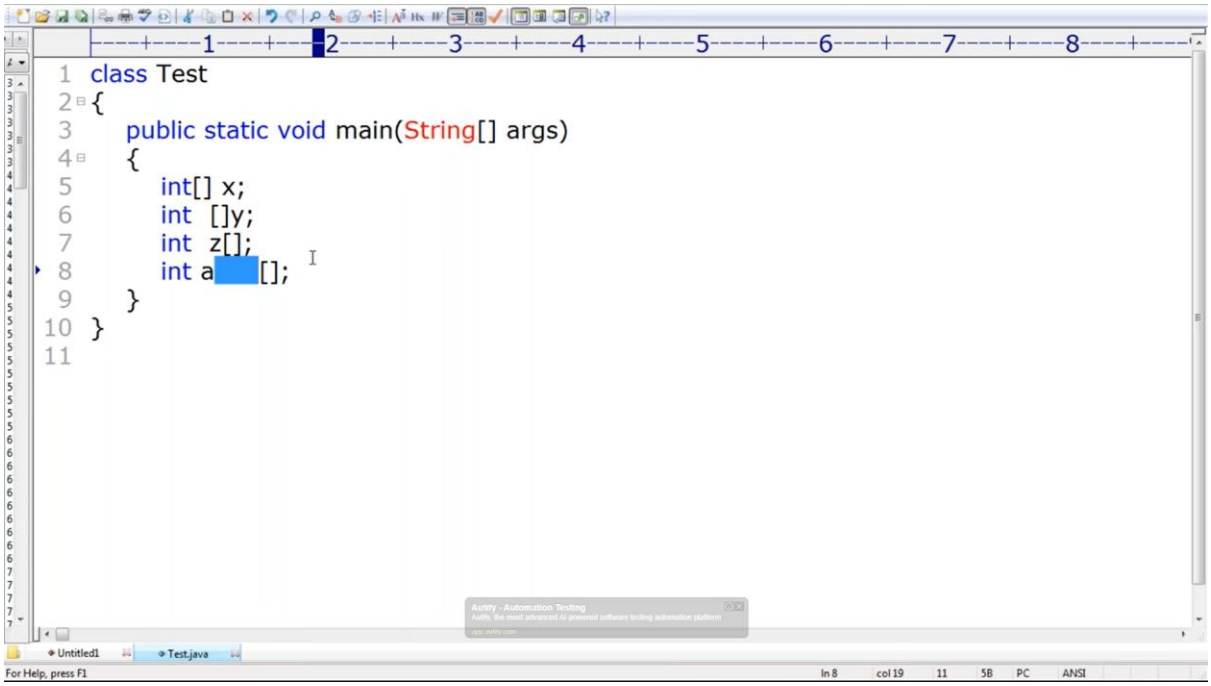
```
C:\Windows\system32\cmd.exe
                    ^
1 error

D:\durgaclasses>javac Test.java

D:\durgaclasses>java Test
1200.0

D:\durgaclasses>javac Test.java
Test.java:5: error: incompatible types: possible lossy conversion from double to
 float
                float f = 1.2e3;
                          ^
1 error

D:\durgaclasses>javac Test.java

D:\durgaclasses>java Test
1200.0
```

```
class Test
{
    public static void main(String[] args)
    {
        float f = 1.2e3F;
        System.out.println(f);
    }
}
```


OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 08 - 01 - 2018
108
109 Exponential form
110
111 double d = 1.2e3;==>1.2*10^3==>1.2*1000==>1200.0
112
113 ---
114
115 sir can't we say for floating point decimal and octal are allow bcz we are always using 0 for octal
116
117 int x =010;
118 sop(x);//8
119
120 double d = 0123.456;
121 sop(d); 123.456
122


OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 17 - 01 - 2018
10 10000 values
11
12 int[] x = new int[10000];
13 x[0],x[1],x[2]...
14 ----
15 Fixed in size
16 x[0]=10;
17 x[1]=20;
18 x[2]=true;
19 homogeneous===>same type
20
21 An array is an indexed collection of fixed number of homogeneous data elements
22 ----------
23
Autify - Automation Testing
Autify, the most advanced AI-powered software testing automation platform
app.autify.com
11:49 / 1:11:53





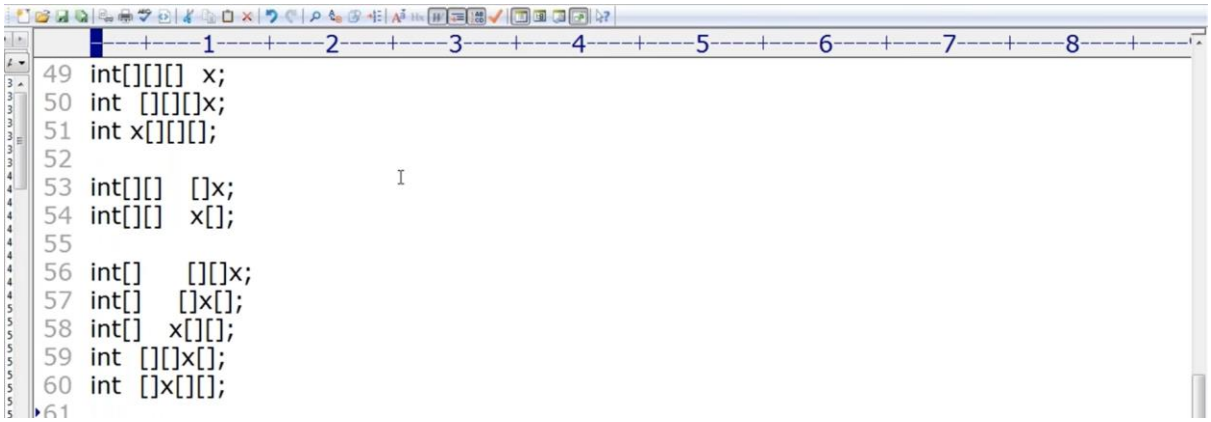
Space ignored by compiler

Cant specify size at time of declartion but can be done at the time of creation

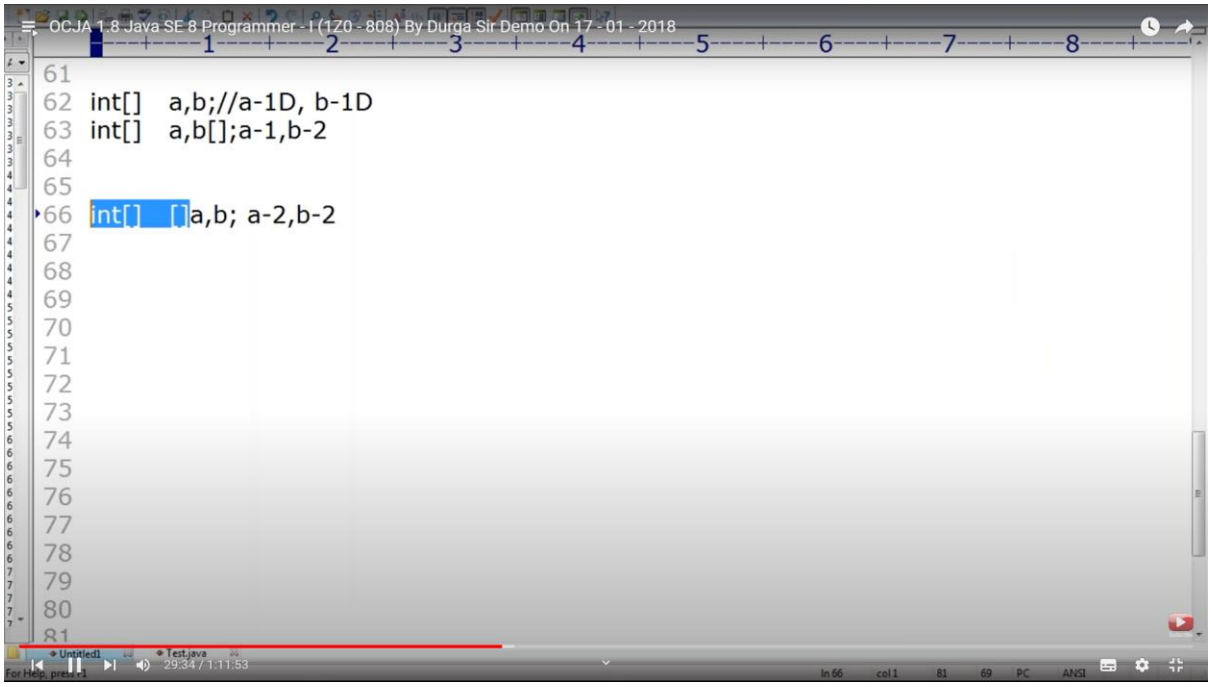
```
33
34
35 int[4] x;
```

```
6     }
7 }
8
D:\durgaclasses>javac Test.java
Test.java:5: error: ']' expected
                int[4] x;
                    ^
Test.java:5: error: not a statement
                int[4] x;
                  ^
Test.java:5: error: illegal start of expression
                int[4] x;
                    ^
3 errors
```

All the below are valid

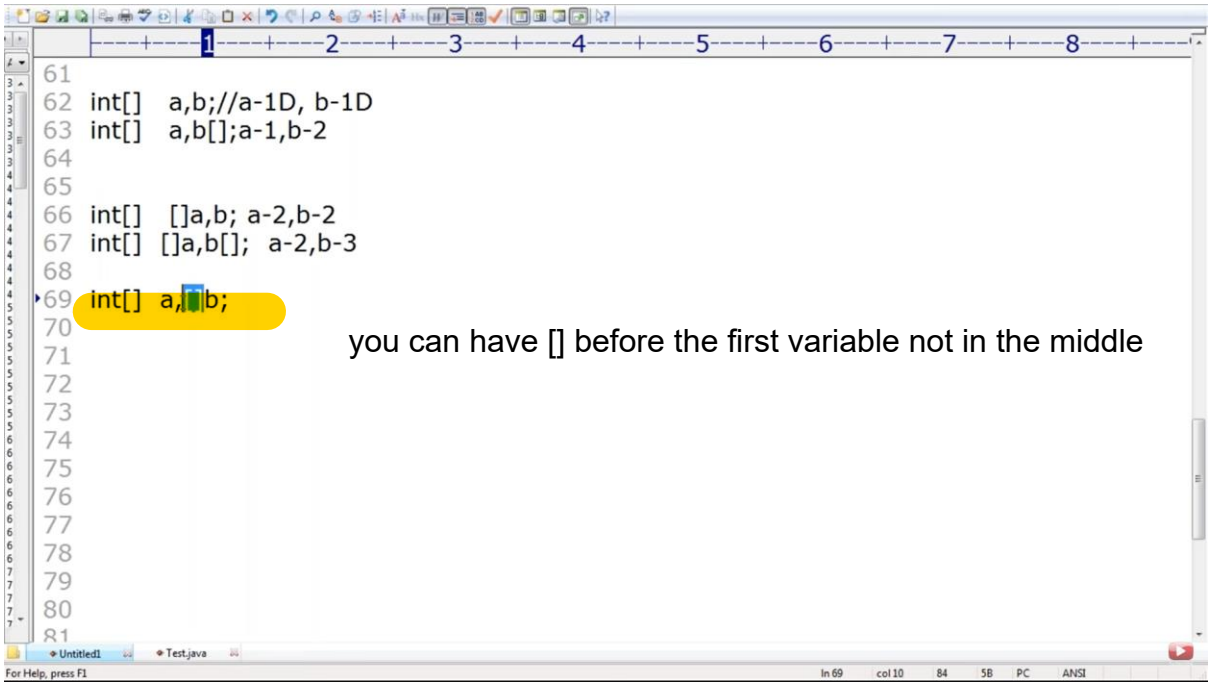


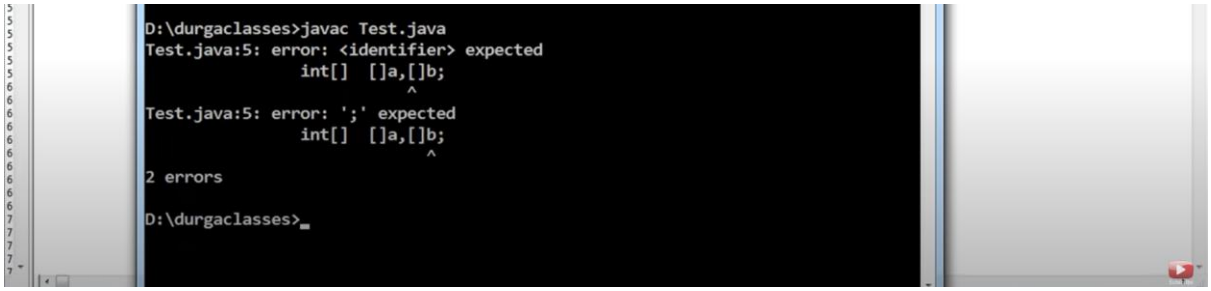
Sppace will be ignored both are 2D



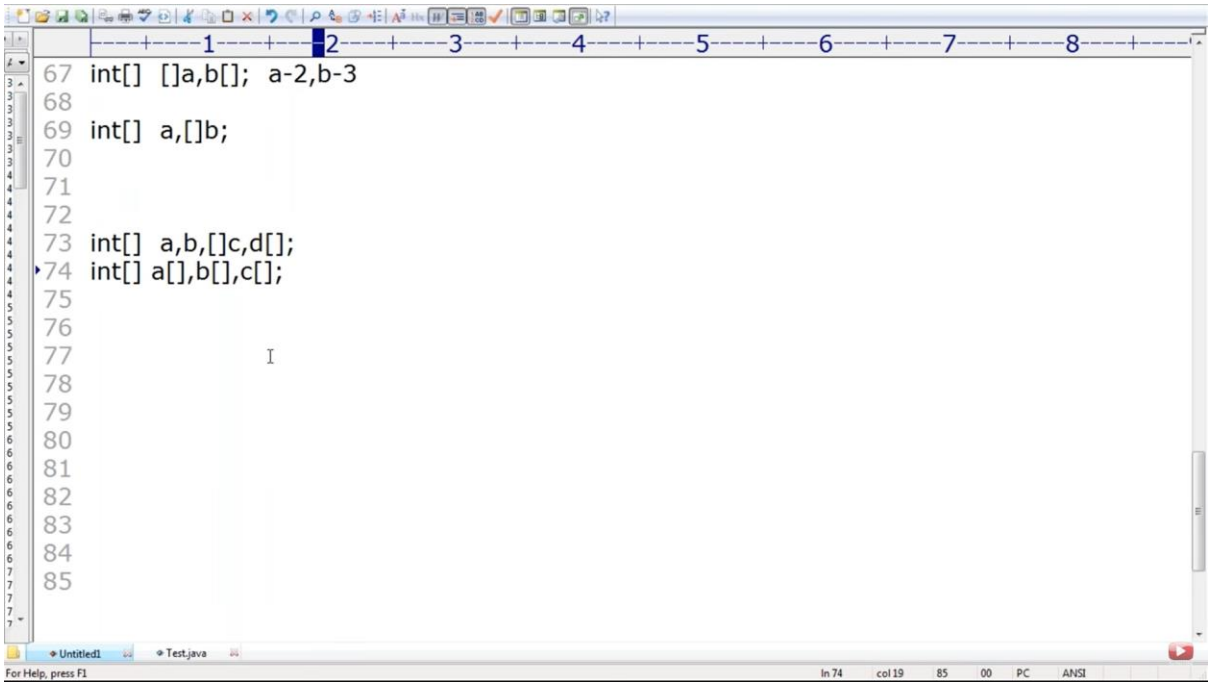
Declaring the dimension is only allowed before the first variable, not before the second variable.

Below will throw an error.

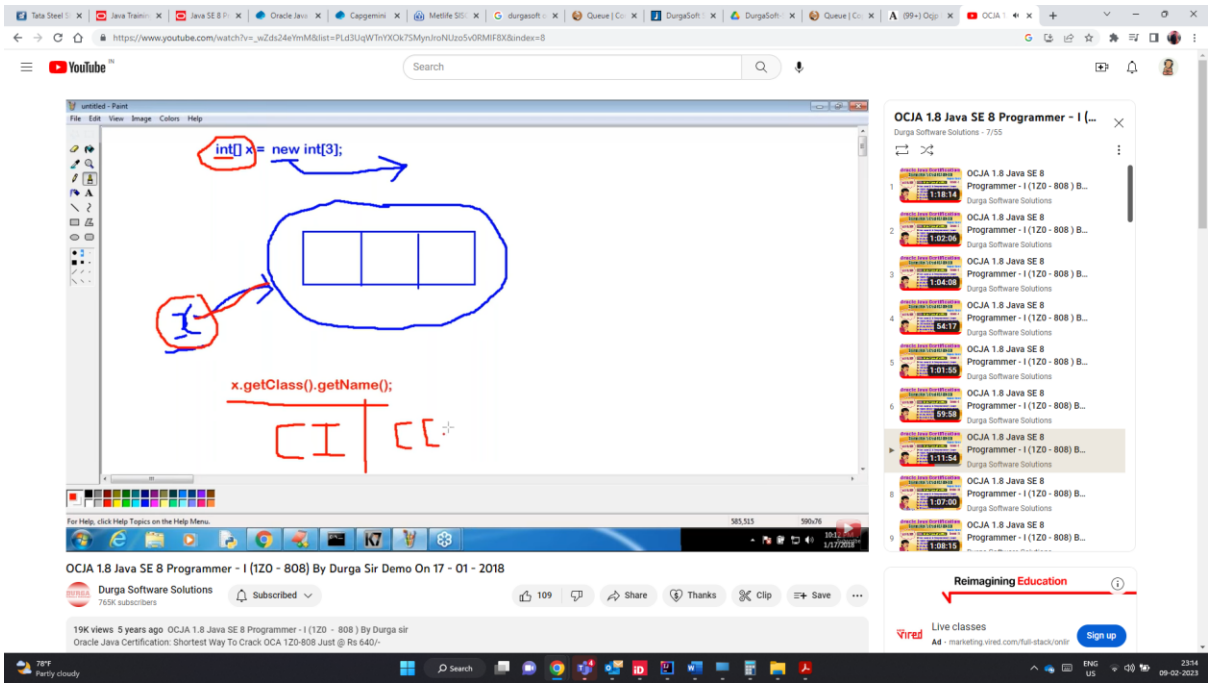




73- Error 74 right

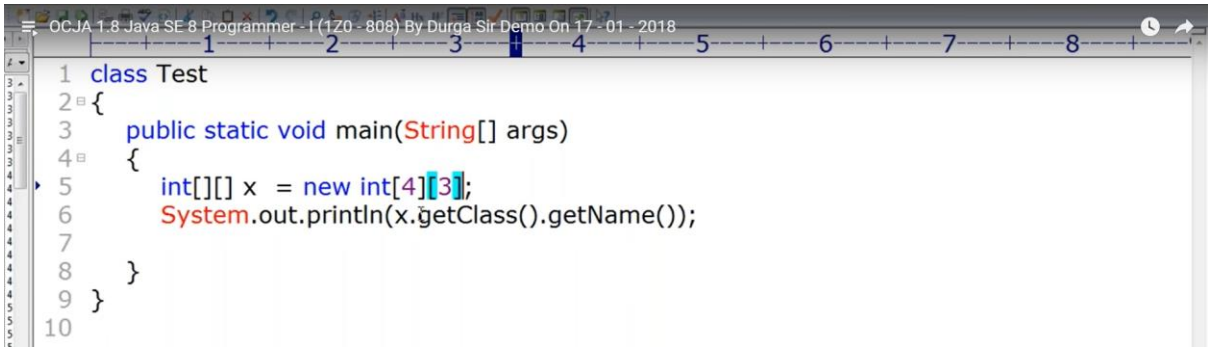


For every array type the class files are available at language level not at API level

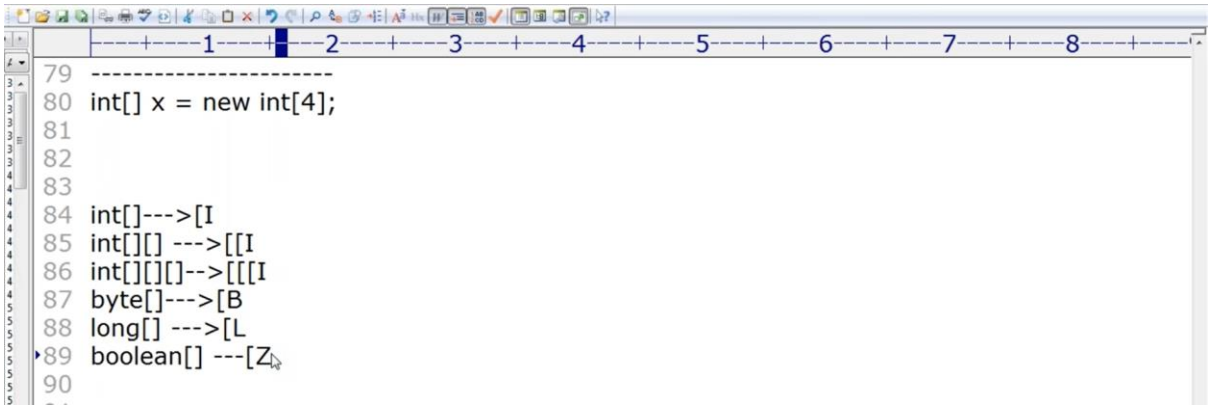


X – is object reference . x.getClass().getName();

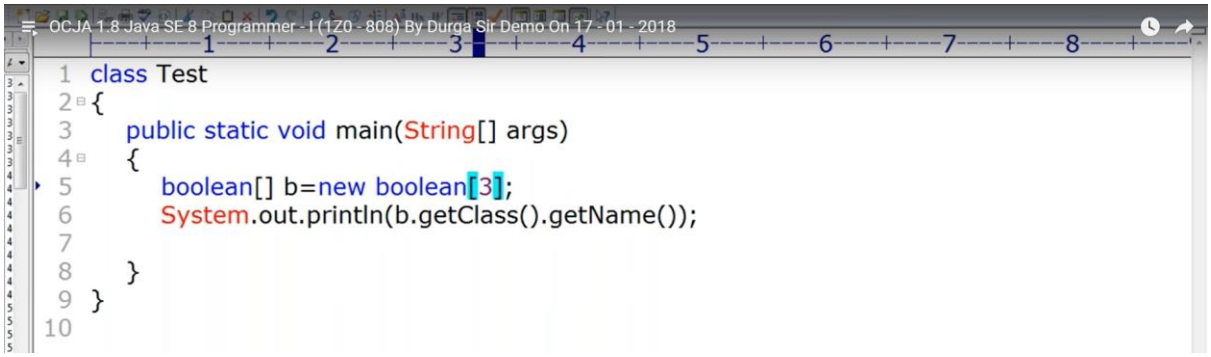
The class name is [I – [ref 1 dimension I rep int; two dimensions [[I



Knowledge purpose



```
79  ----------------------
80  int[] x = new int[4];
81
82
83
84  int[]--->[I
85  int[][] --->[[I
86  int[][][]-->[[[I
87  byte[]--->[B
88  long[] --->[L
89  boolean[] ---[Z
90
```




```
1  class Test
2  {
3      public static void main(String[] args)
4      {
5          boolean[] b=new boolean[3];
6          System.out.println(b.getClass().getName());
7
8      }
9  }
10
```

Dimension is madatory at time or creation. 92 Compile error

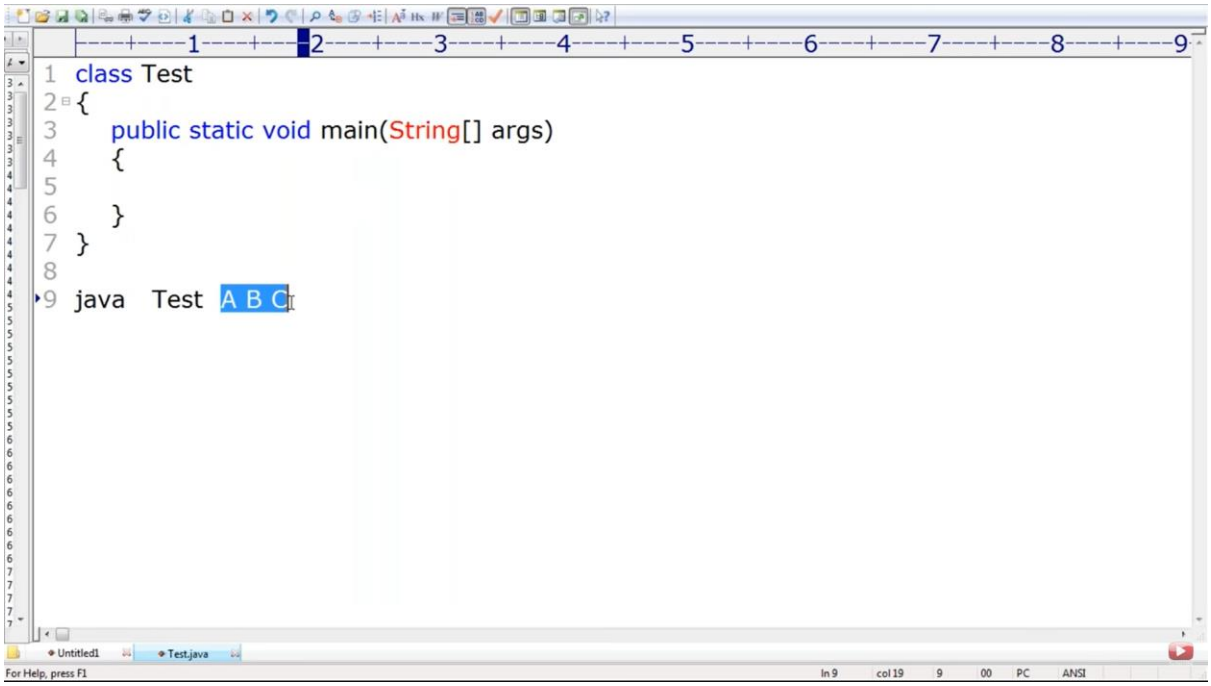
```
92  int[] x = new int[];
93  int[] x = new int[4];
94
95
```

Syntaxically valid but an array of 0 size cant store any value

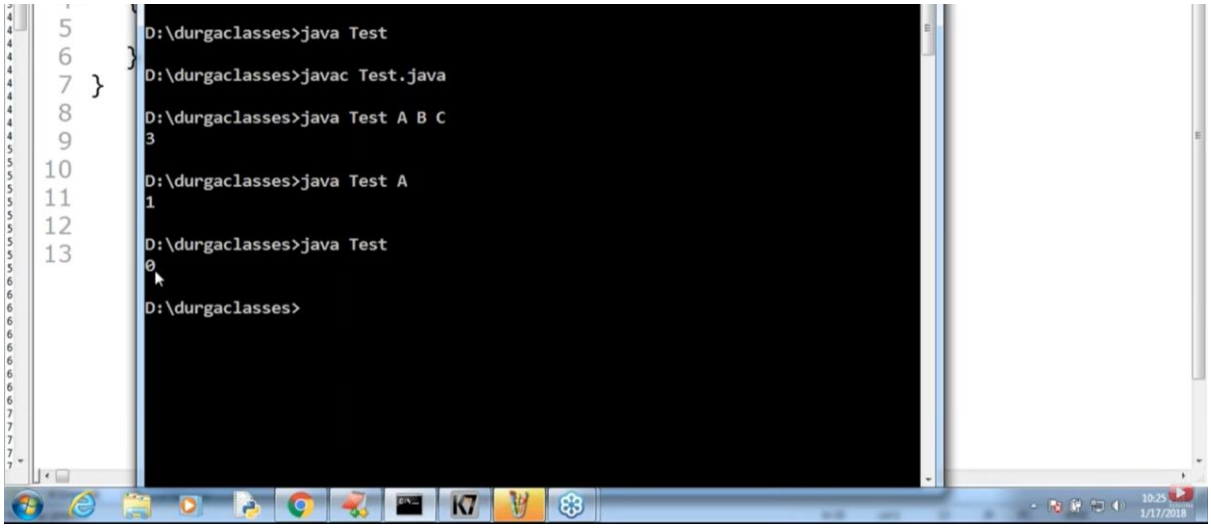
```
95
96
97  int[] x = new int[0];
98
```

Index 0 out of bounds for length 0

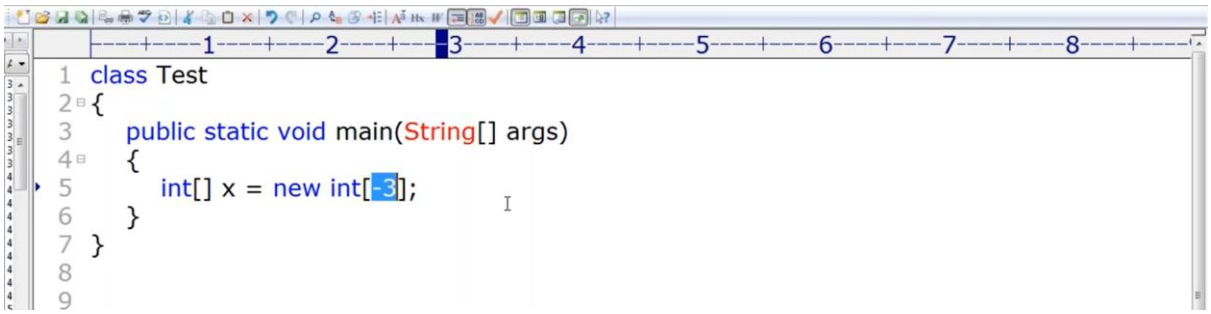
JVM will create a String array with three values A B and C. String args[] = of length 3



Below says an array will be created with size 0



Compiler will always look for type here int and the value is int so no compile time error. JVM are run time allocate memory based on size , Negative array size exception

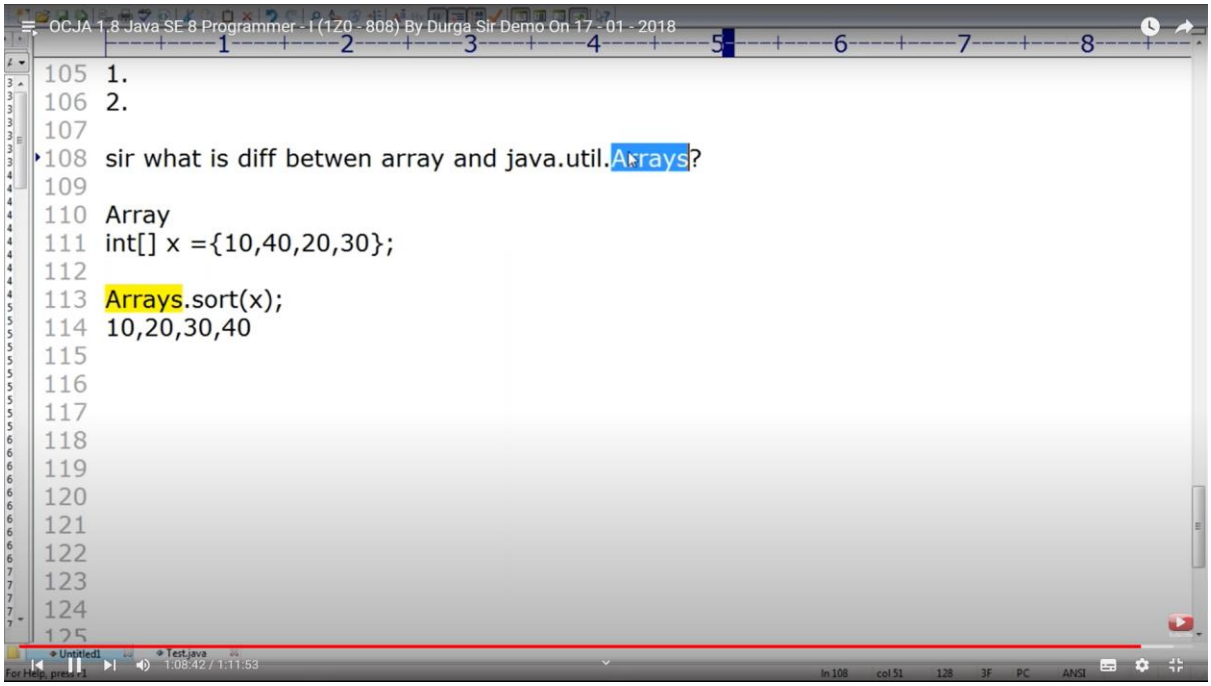


```
class Test
{
    public static void main(String[] args)
    {
        int[] x = new int[-3];
    }
}
```

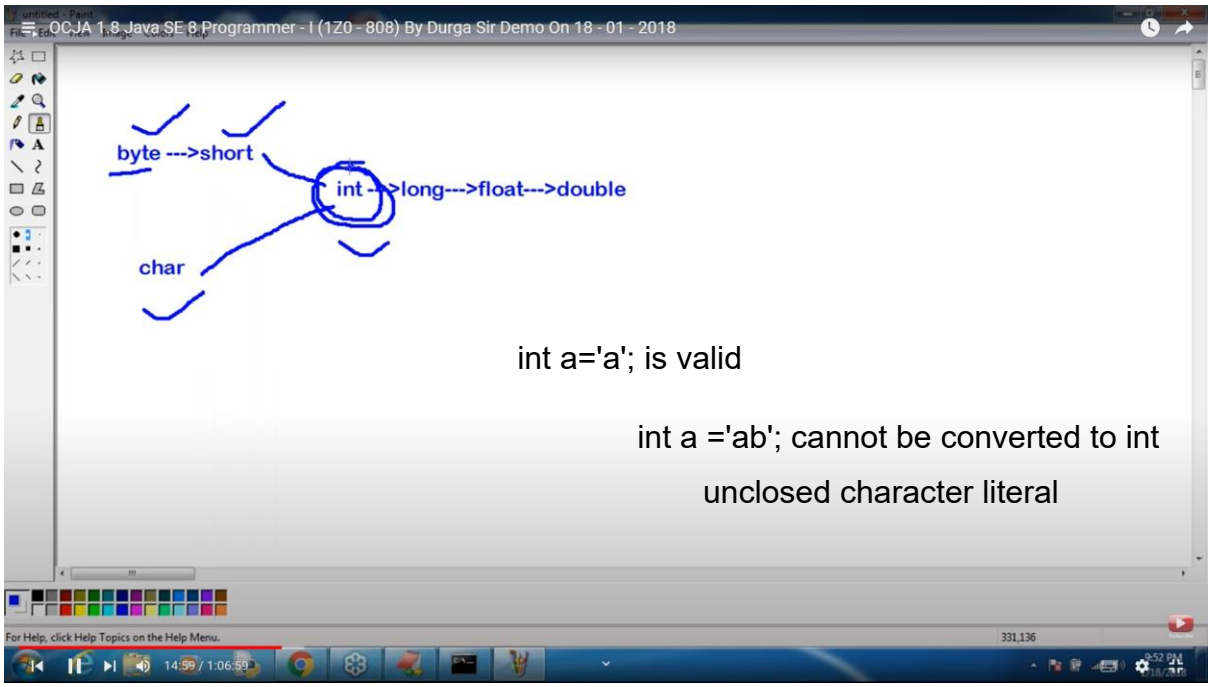
```
D:\durgaclasses>javac Test.java

D:\durgaclasses>java Test
Exception in thread "main" java.lang.NegativeArraySizeException
        at Test.main(Test.java:5)
```

If we want utilities method for array we need java.utl

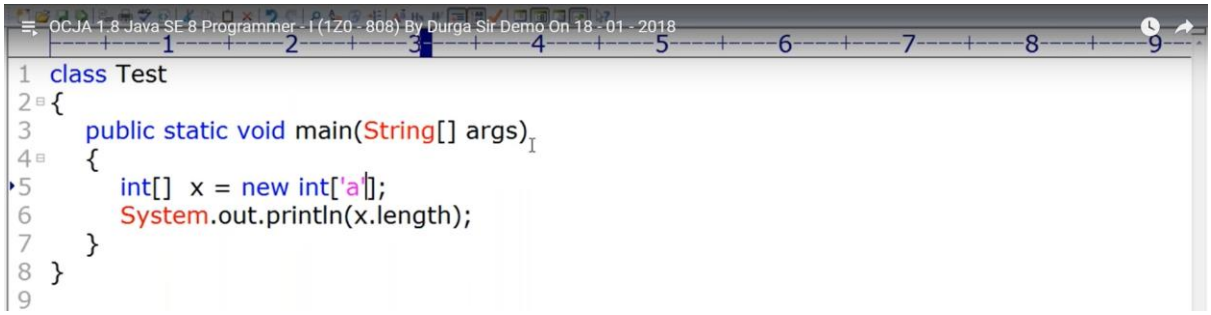
il.arrays



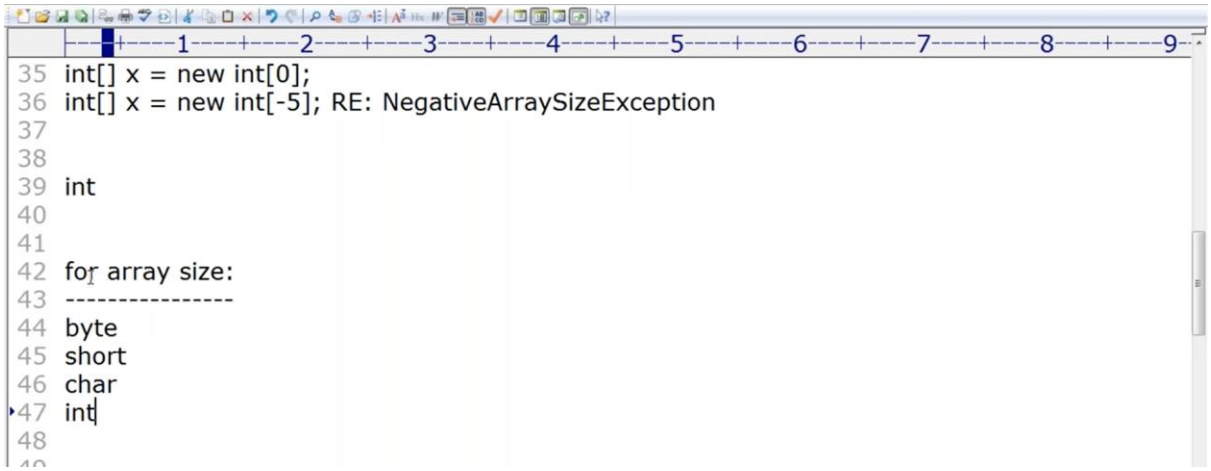
Byte short char are by default promoted to int

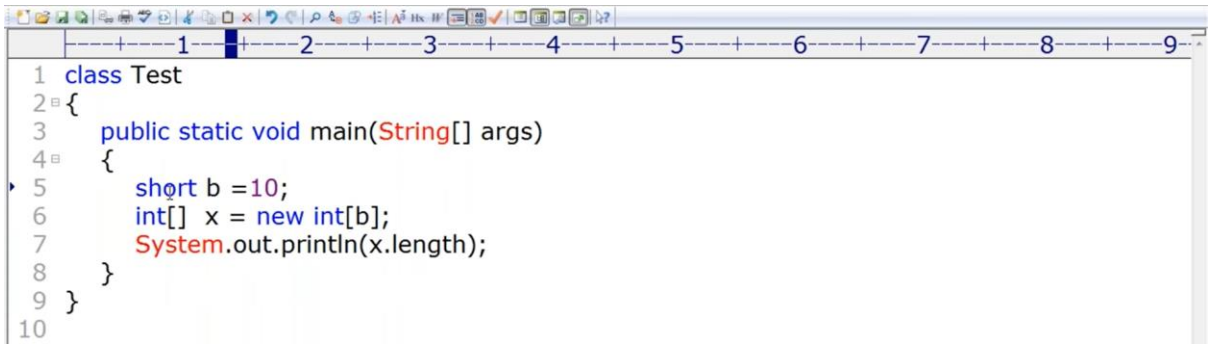


A -97 : size is 97



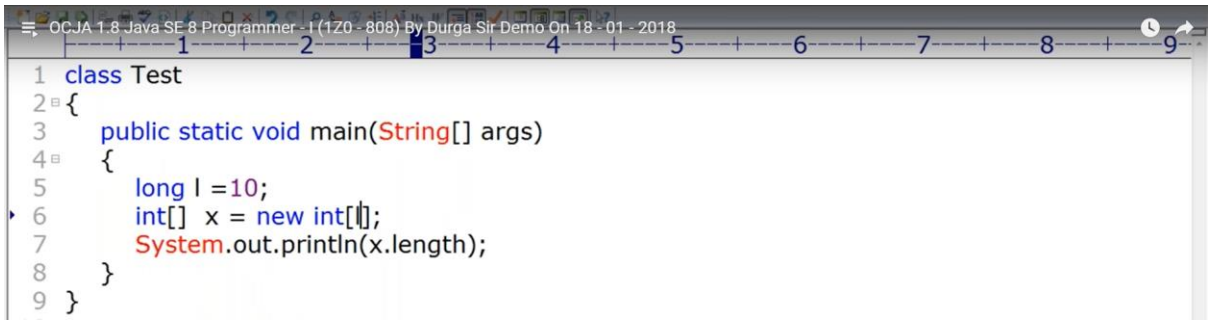
Below are the size alloweds



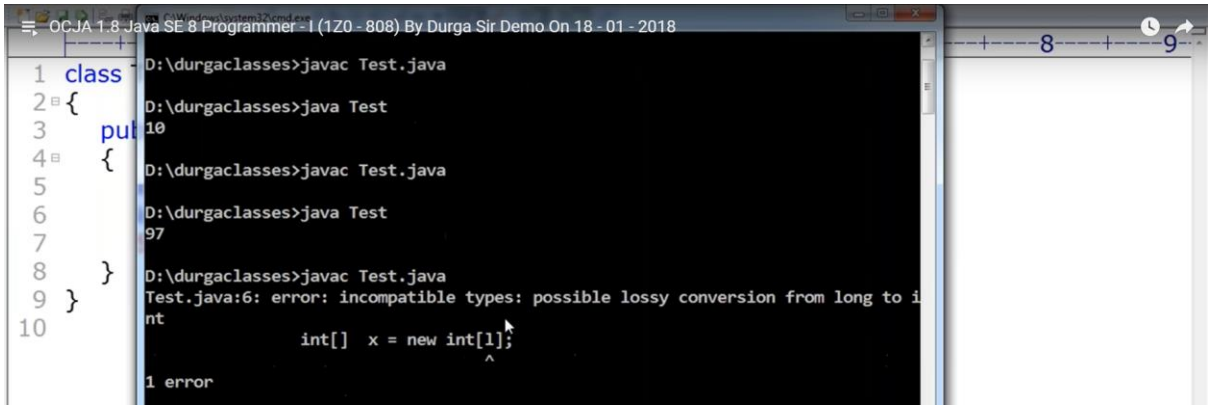


```
class Test
{
    public static void main(String[] args)
    {
        short b =10;
        int[]  x = new int[b];
        System.out.println(x.length);
    }
}
```

Compile error




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



OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 18 - 01 - 2018

```
1  class Test
2  {
3     public static void main(String[] args)
4     {
5        long l =10;
6        int[]  x = new int[10.0f];
7
8
9        System.out.println(x.length);
10    }
11 }
12
```

```
9                                  ^
10    }  1 error
11 }
12
D:\durgaclasses>javac Test.java
Test.java:6: error: incompatible types: possible lossy conversion from float to
int
                int[]  x = new int[10.0f];
                                   ^
```

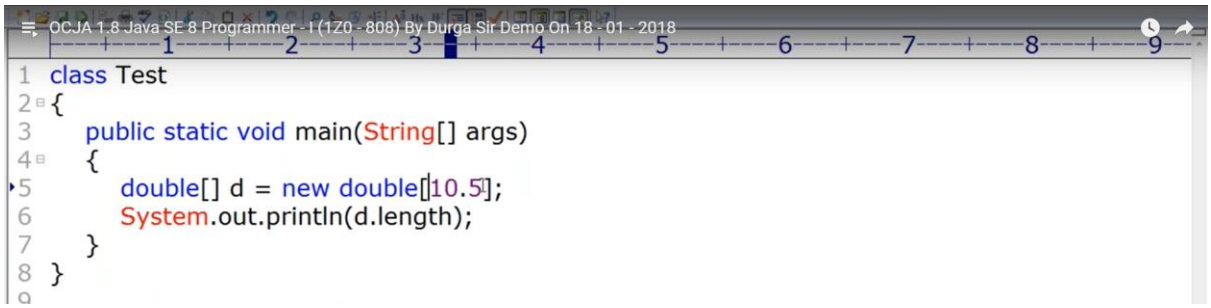
Datatype is double but the size is int

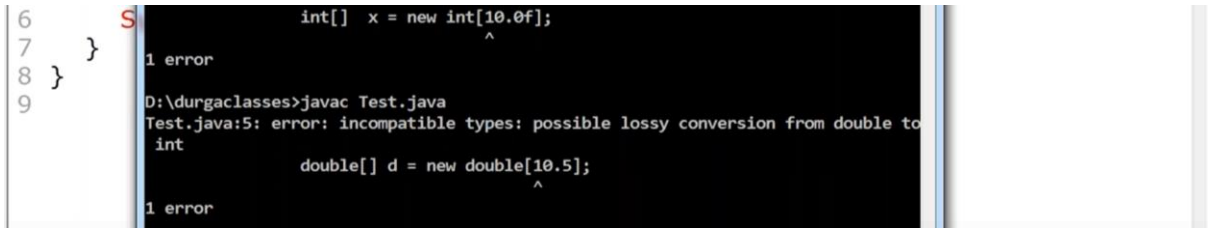
```
48
49
50 double[] d = new double[10];
51
```

```
49
50 double[] d = new double[10];
51
52 array type: double
53 array size : 10
54
```

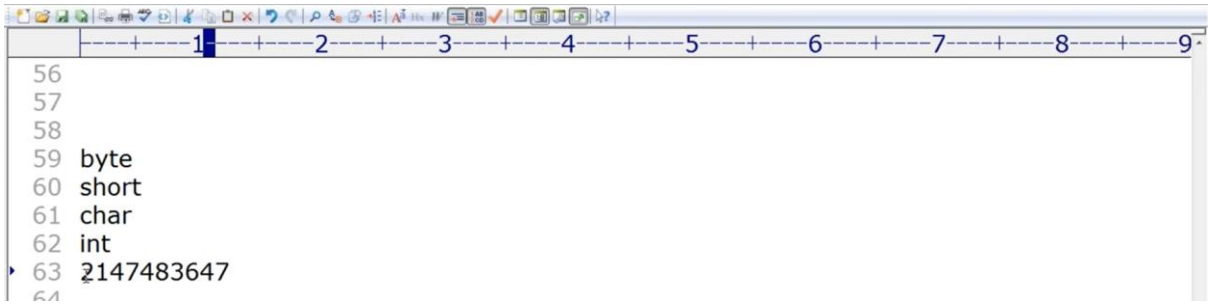
Double casted to int is valid int accepted – Explicit type casting or impliciting type case allowed

```
49
50 double[] d = new double[(int)10.5];
51
52 array type: double
53 array size : 10
54
```

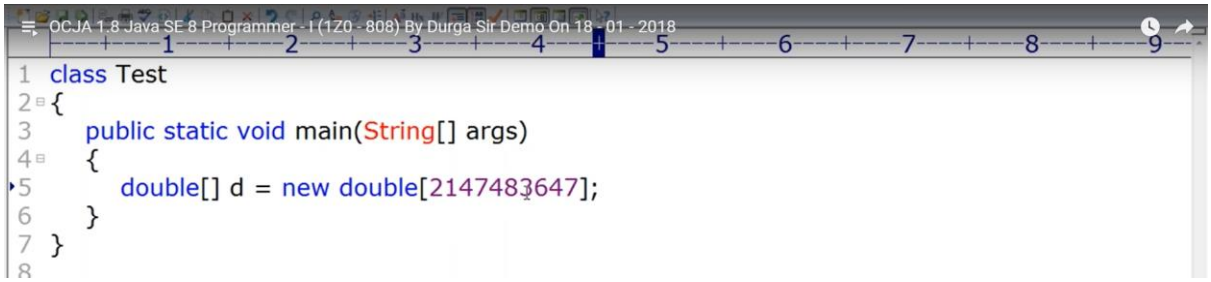




Max allowed array size- max int value



valid



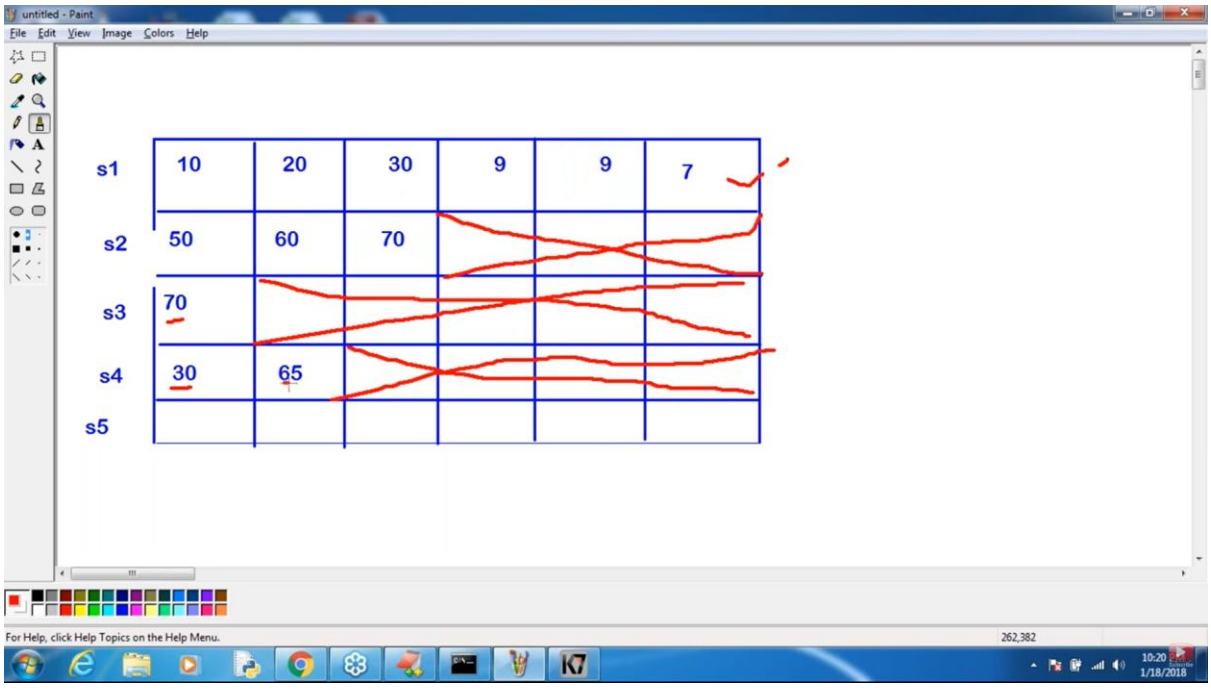
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 18 - 01 - 2018

```
1  class Test
2  {
3      public static void main(String[] args)
4      {
5          double[] d = new double[2147483647];
6          System.out.println("Hello Listen Dont sleep");
7      }
8  }
9
10 // 2147483647*8 bytes: OutOfMemoryError
```

```
7      }
8  }
9
10 // 214
```

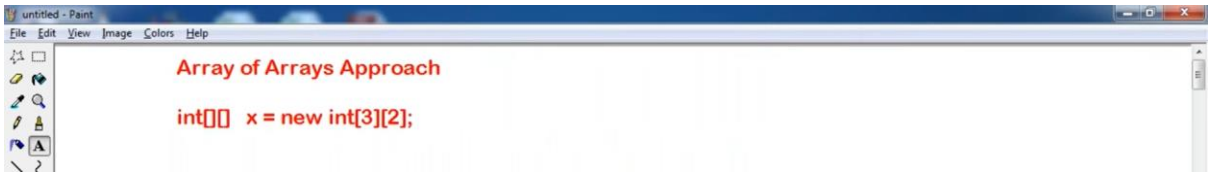
```
1 error

D:\durgaclasses>javac Test.java

D:\durgaclasses>javac Test.java

D:\durgaclasses>java Test
Exception in thread "main" java.lang.OutOfMemoryError: Requested array size exce
eds VM limit
        at Test.main(Test.java:5)
```

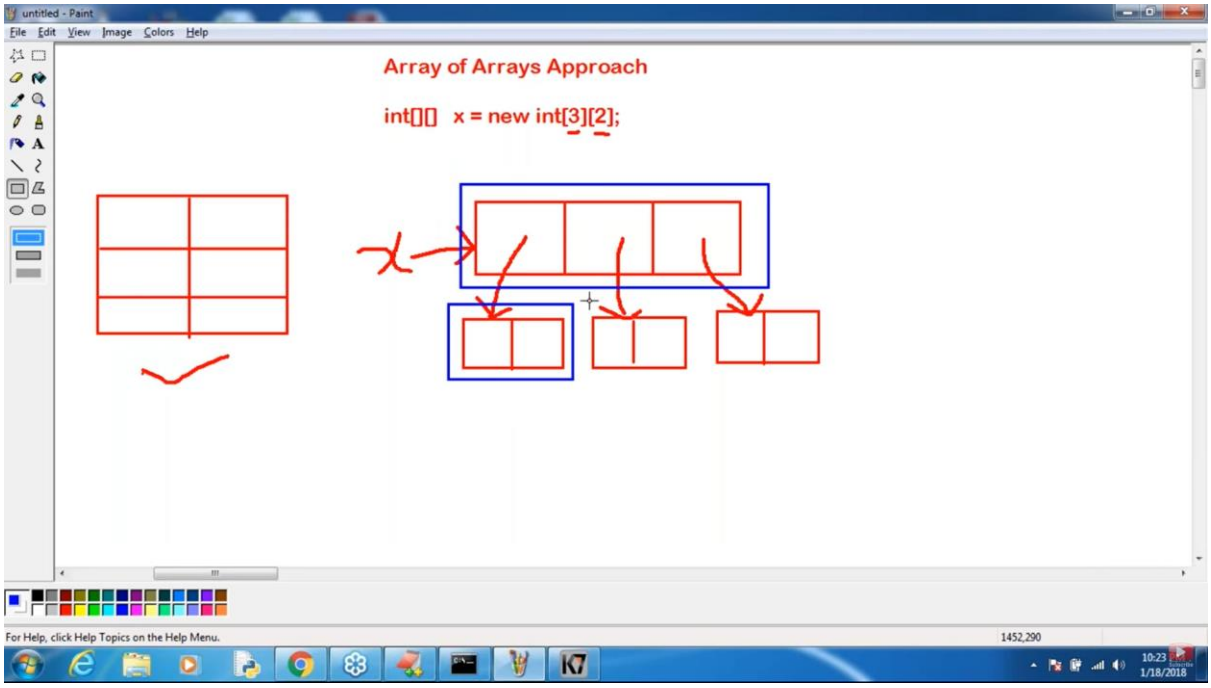
In Jave 2D are not implemented as Matrix Style due to memory wastage

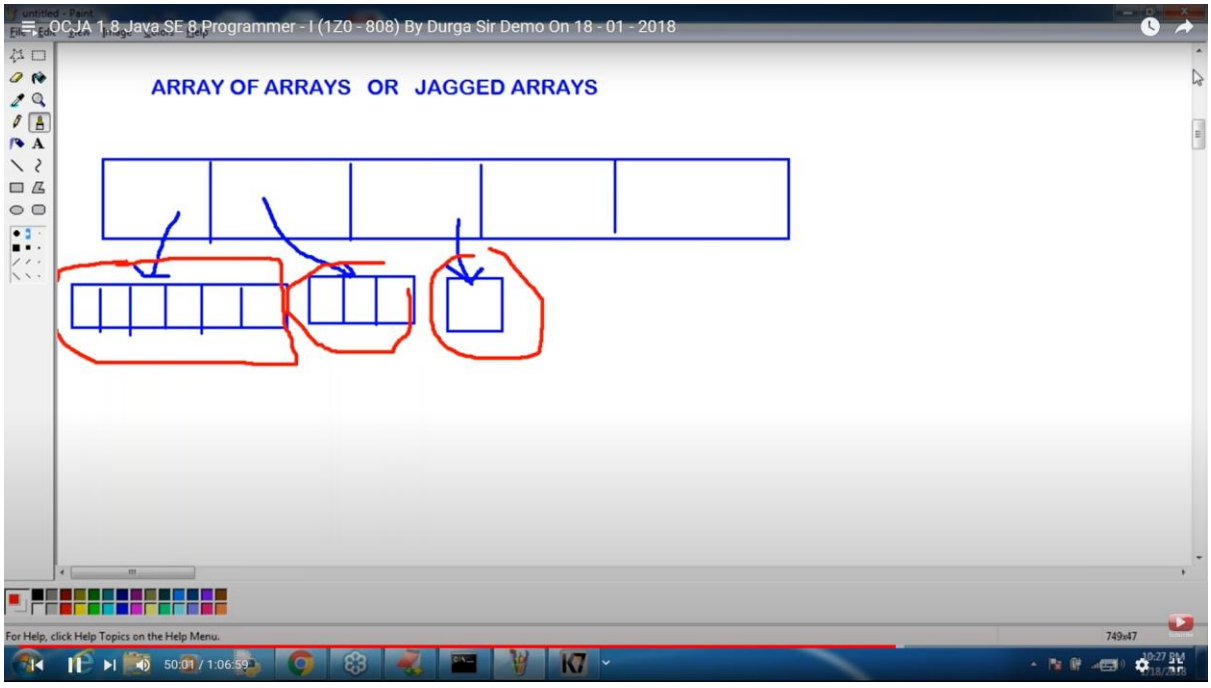


Arrays of Arrays Approach is followed

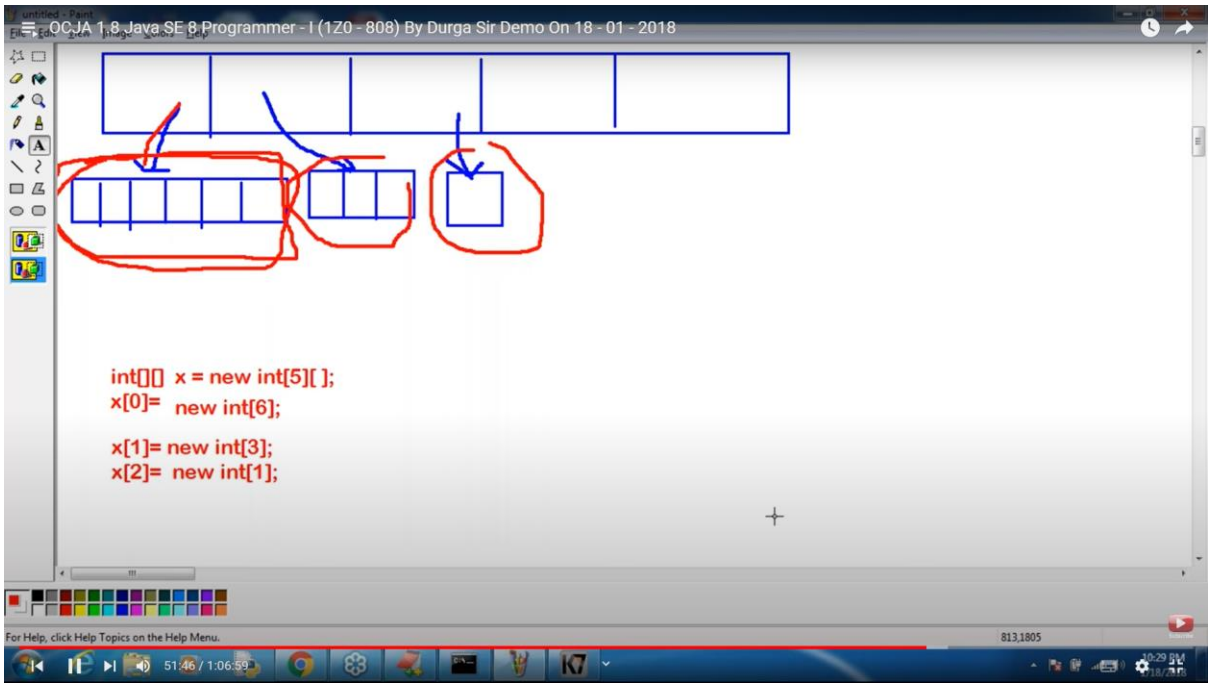
1

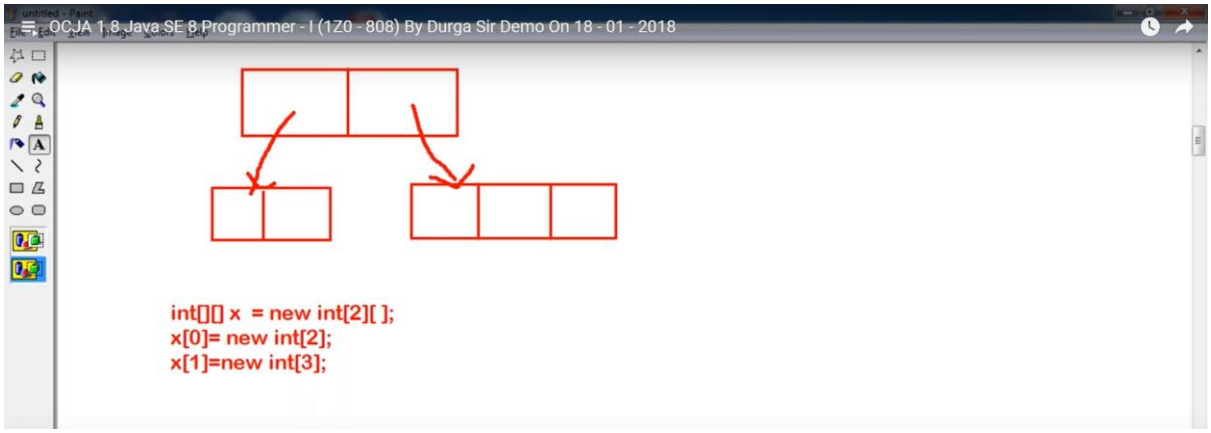


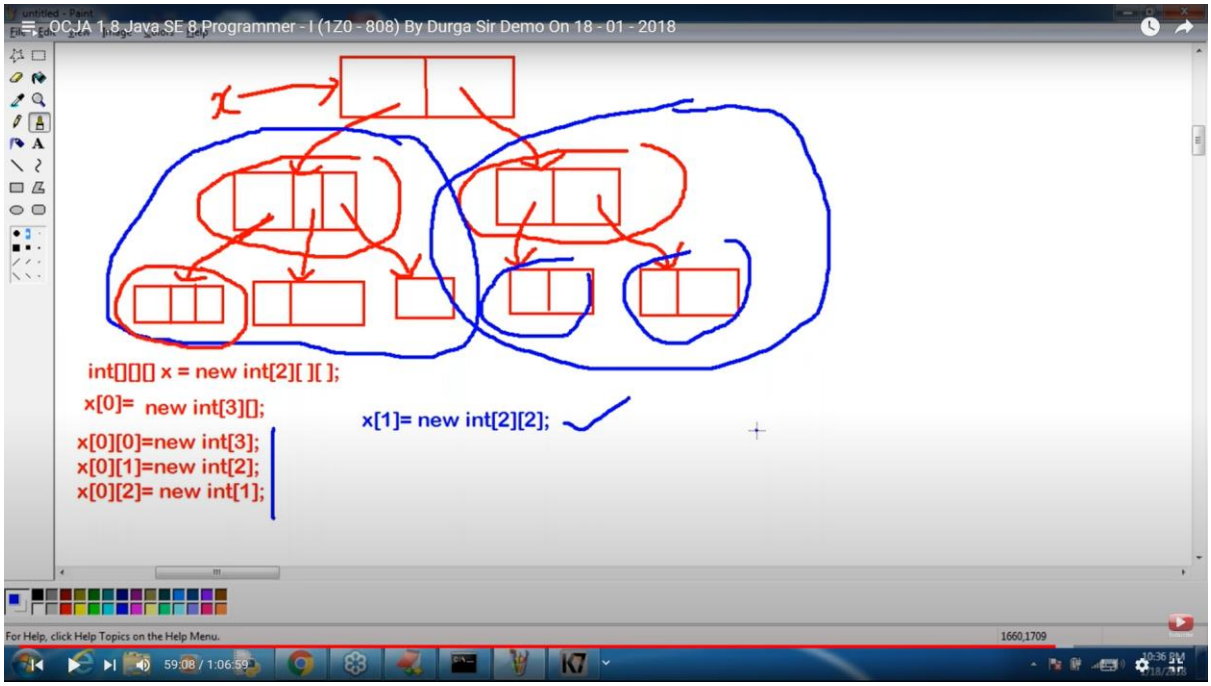




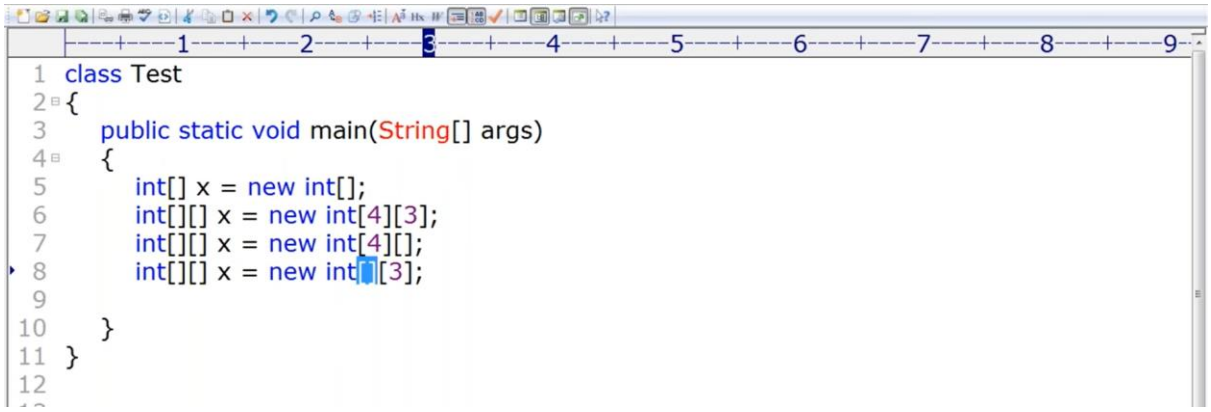
In java its ok if we only specify the base size not all dimension





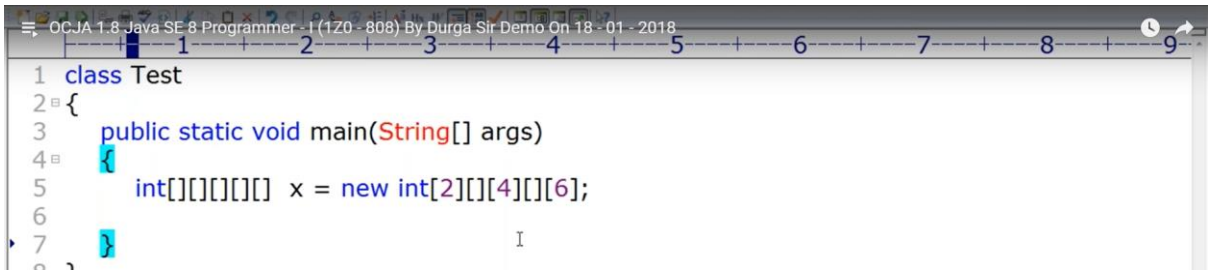


Line 8 error

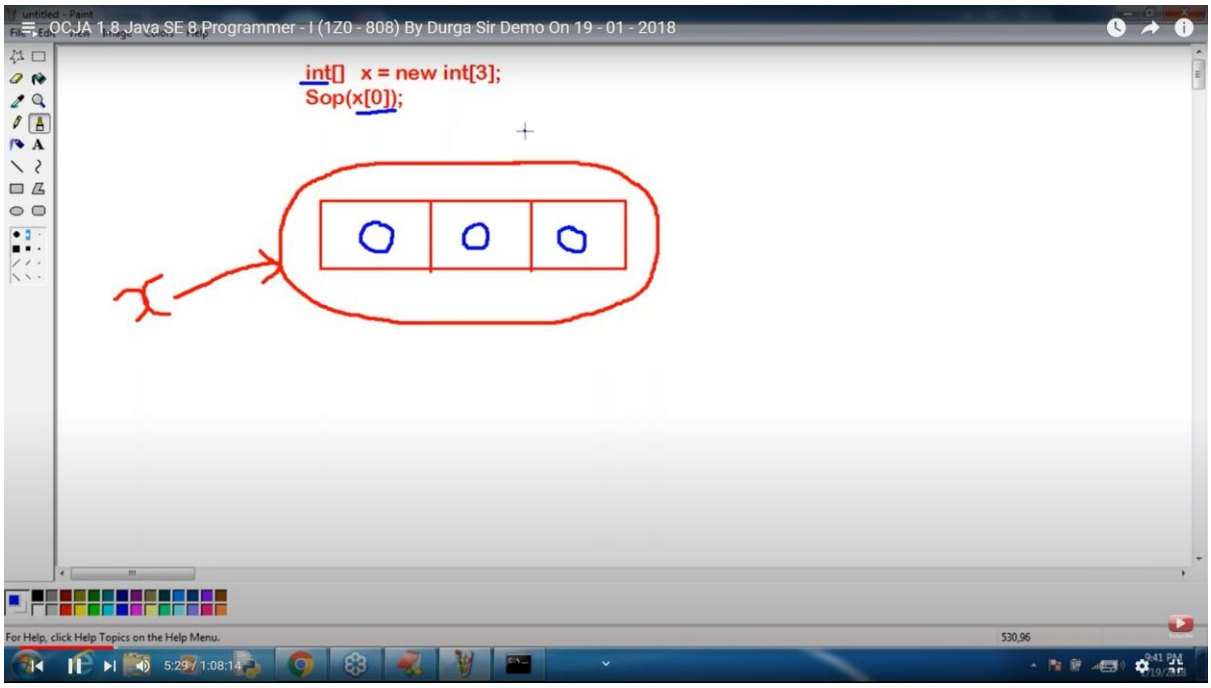


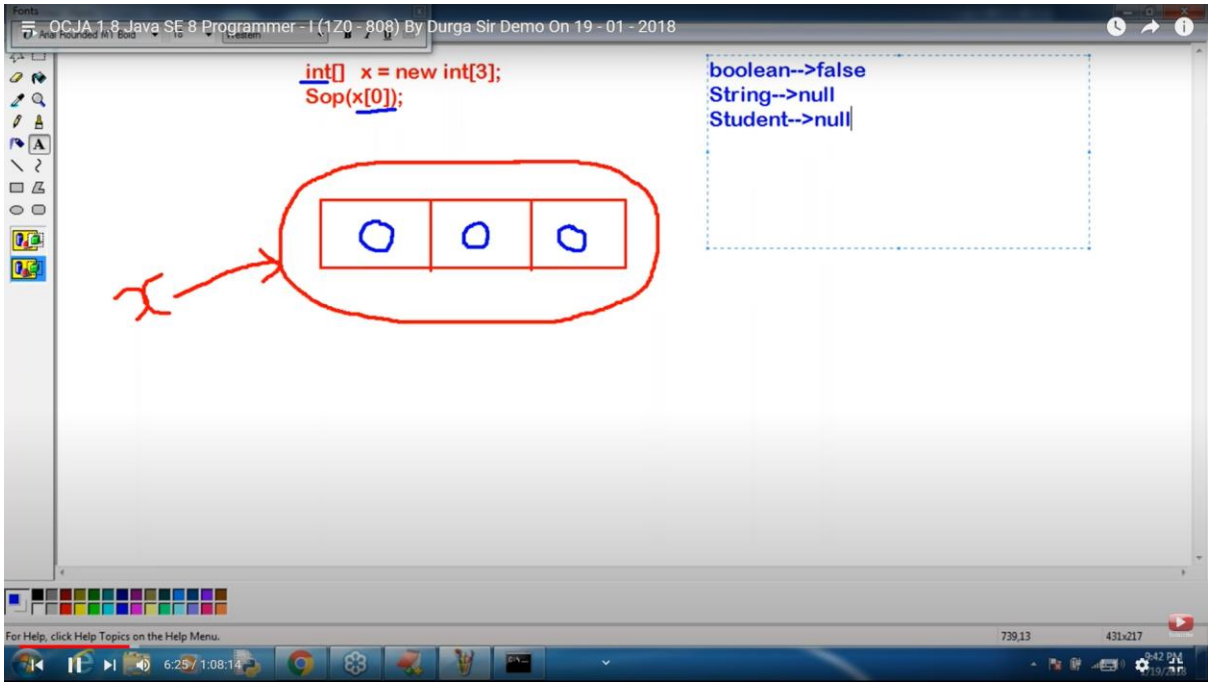
```
1  class Test
2  {
3      public static void main(String[] args)
4      {
5          int[] x = new int[];
6          int[][] x = new int[4][3];
7          int[][] x = new int[4][];
8          int[][] x = new int[][3];
9
10     }
11 }
12
```

Without knowing 3rd D how would you specify 4th and without knowing 6th how would you specify 7

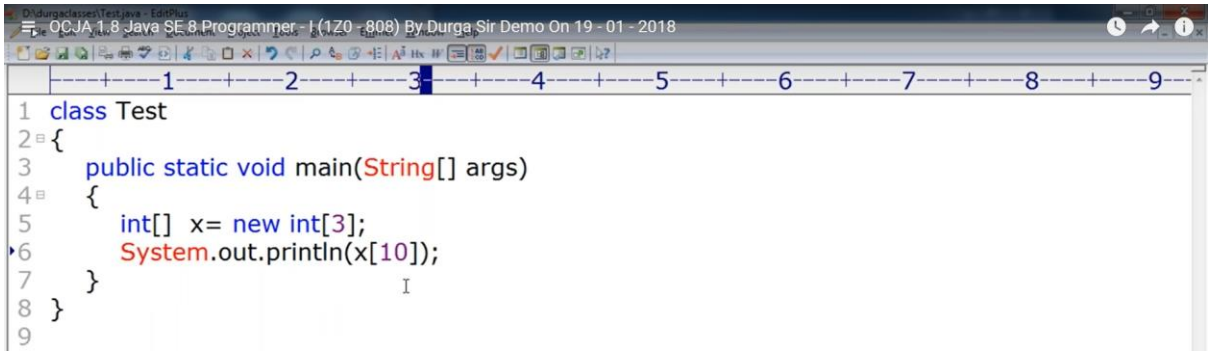


Every array element is of default value.

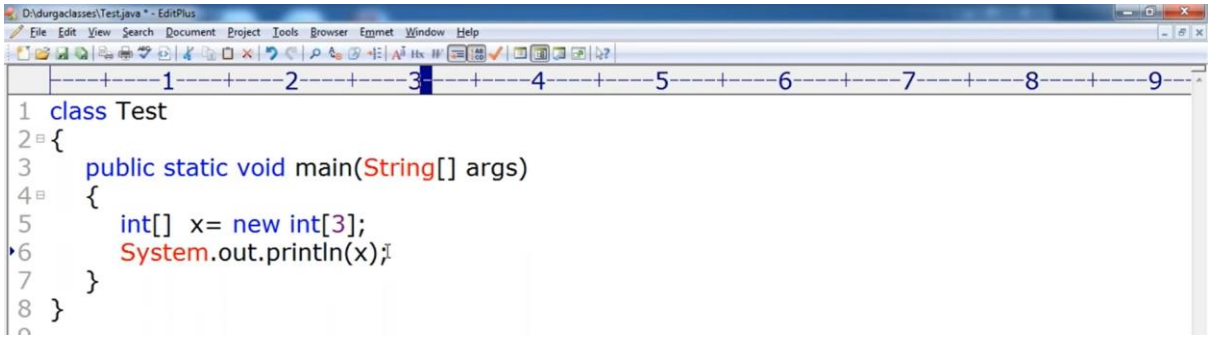




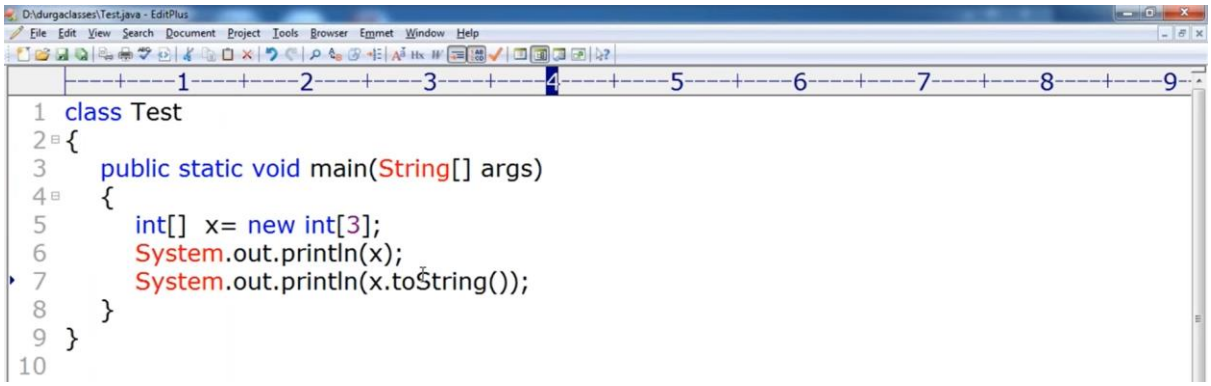
RE – Array Index OOB Exception

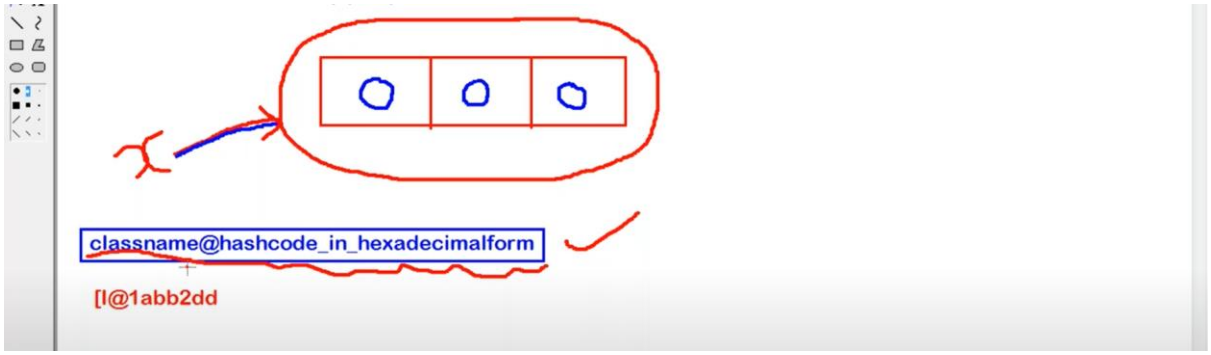


When we print or assign ref variable the toString method will be called



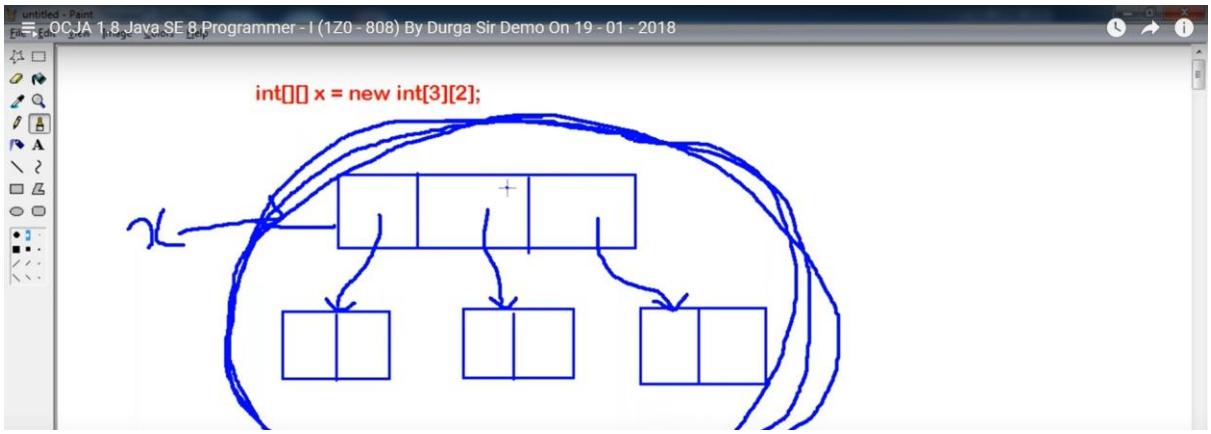
X will be converted to x.toString(). The default implementation of toString method is classname@hascode_in_hexadecimal. The hashcode will change from machine to machine





OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 19 - 01 - 2018

```
1  class Test
2  {
3      public static void main(String[] args)
4      {
5          int[][] x = new int[3][2];
6          System.out.println(x);//[[I@hashcode
7          System.out.println(x[0]);
8          System.out.println(x[0][0]);
9      }
10 }
11
```



OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 19 - 01 - 2018

```
1  class Test
2  {
3      public static void main(String[] args)
4      {
5          int[][] x = new int[3][2];
6          System.out.println(x);//[[I@hashcode
7          System.out.println(x[0]);//[I@hashcode
8          System.out.println(x[0][0]);//0
9      }
10 }
11
```

```
1  Array Declaration
2  Array Creation
3  Array Initialization
4
5  java.lang package==>Object class methods hashCode()
6  int[] x = new int[3];
7  x[0]=100;
8  x[1]=20;
```

```
21
22 int[] x = {10,20,30};
23 char[] ch={'a','e','i','o','u'};
24 String[] s ={"Katrina","Kareena","Sunny","mallika"};
25
```

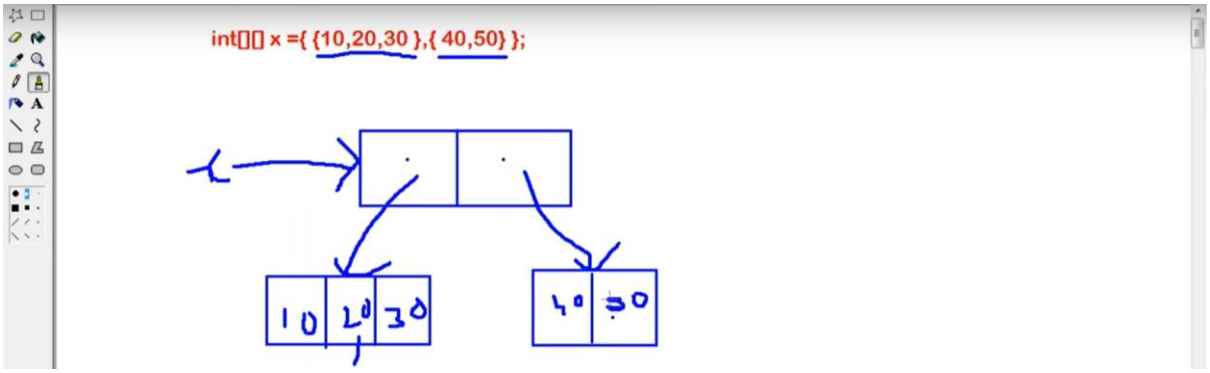
Below is compile time error if you want to use shortcut the initialization and declartion should be in single line.

```
1  class Test
2  {
3      public static void main(String[] args)
4      {
5          int[] x;
6          x={10,20,30};
7          System.out.println(x[0]);
8      }
9  }
```

int x[];
x= new int[] is valid

int x[];
x={10,25,1,5} is invalid

```
27
28 Array of arrays concept
29
30 int[][] x ={ {10,20,30 },{ 40,50} };
31 sop(x[0][1]);//20
32 sop(x[1][1]);//50
33 sop(x[2][0]);//AIOOBE
34
```

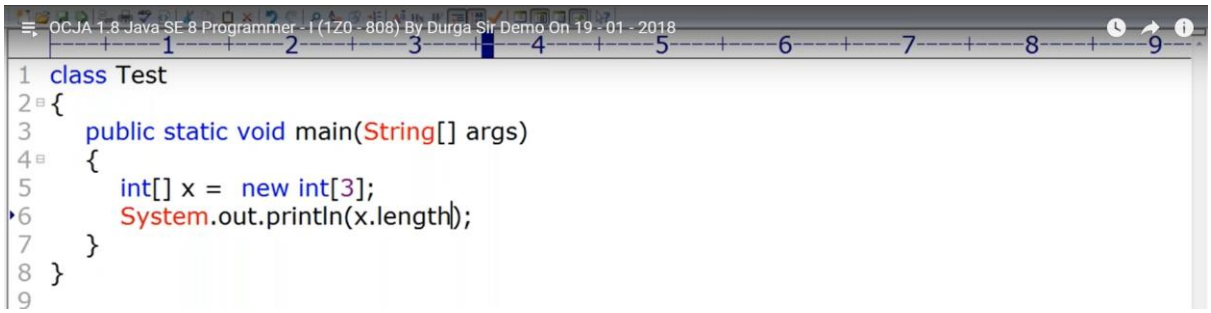


Length method is applicable for array and Length() method is for string length

length() cannot be applied to array – compile time error

```
class Test
{
    public static void main(String[] args)
    {
        int[] x =  new int[3];
        System.out.println(x.length());
    }
}
```

Below is correct



```
class Test
{
    public static void main(String[] args)
    {
        int[] x =  new int[3];
        System.out.println(x.length);
    }
}
```

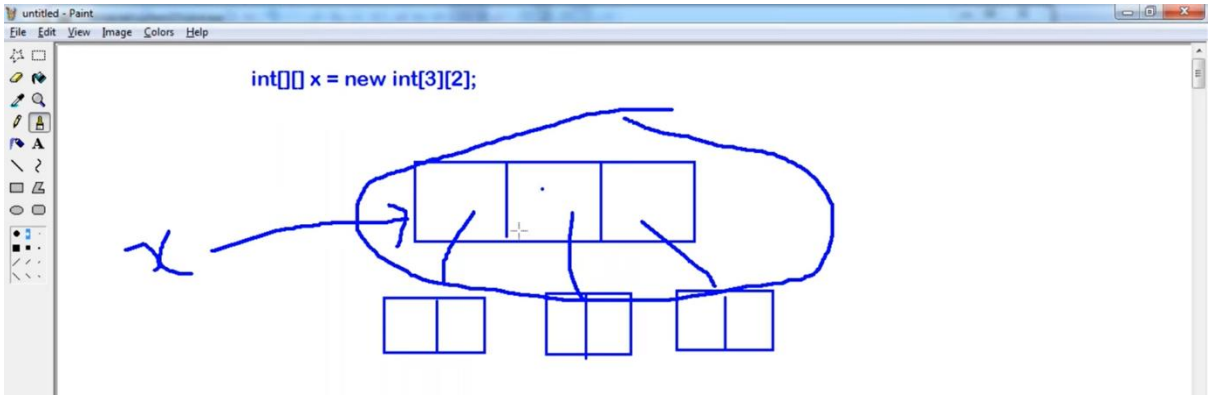
Below is wrong length variable is not applicable for string

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

```
class Test
{
    public static void main(String[] args)
    {
        String[] s ={"A","AA","AAA"};
        System.out.println(s.length);
        //System.out.println(s.length());
        //System.out.println(s[0].length);
        System.out.println(s[0].length());
    }
}
```

```
class Test
{
    public static void main(String[] args)
    {
        int[][] x = new int[3][2];
        System.out.println(x.length);
    }
}
```

Length - 3



X[0].lenth =2

```
class Test
{
    public static void main(String[] args)
    {
        int[][] x = new int[3][2];
        System.out.println(x.length);
        System.out.println(x[0].length);
    }
}
```

```
41 Anonymous Arrays:
42 ---------------------
43 int[] x={10,20,30};
44
45
46 Array just for instant use
47
48 new int[] {10,20,30}
49
50 new int[][] {{10,20},{30,40,50}}
```

OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 19 - 01 - 2018

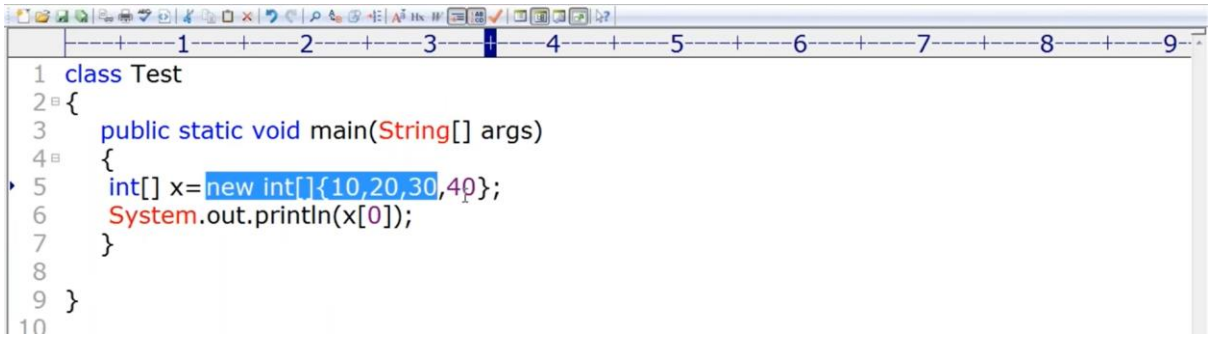
```
1  class Test
2  {
3     public static void main(String[] args)
4     {
5        //10,20,30,40
6        sum(new int[] {10,20,30,40});
7
8     }
9     public static void sum(int[] x)
10    {
11       int total=0;
12       for (int x1:x )
13       {
14          total=total+x1;
15       }
16       System.out.println("The Sum:"+total);
17    }
18 }
19
```

For anonymous array you cant specify size

```
1  class Test
2  {
3     public static void main(String[] args)
4     {
5        //10,20,30,40
6        sum(new int[4] {10,20,30,40});
7
```

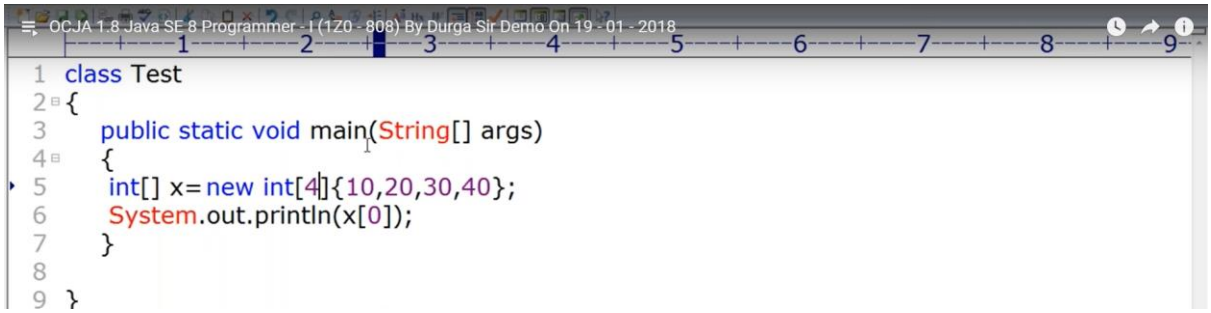
```
C:\Windows\system32\cmd.exe
D:\durgaclasses>javac Test.java
Test.java:6: error: ')' expected
              sum(new int[4] {10,20,30,40});
                           ^
Test.java:6: error: not a statement
              sum(new int[4] {10,20,30,40});
                             ^
```

You can specify size or not .you should either specify size or elements if you specify both. U will get compile time error. Here you are creating an annoymous array and assigning it to x array.

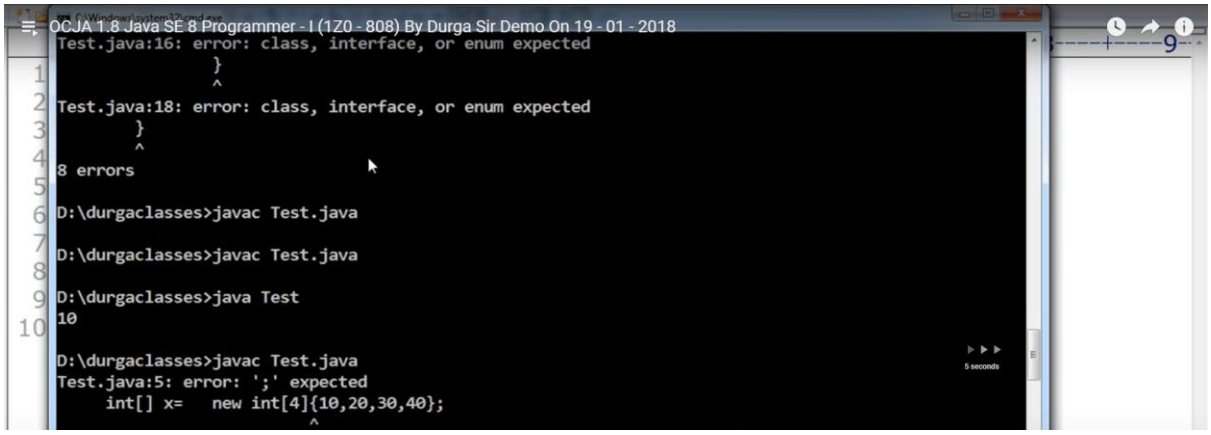


```
class Test
{
    public static void main(String[] args)
    {
     int[] x=new int[]{10,20,30,40};
     System.out.println(x[0]);
    }

}
```

If you specify both size and elements it will throw errror



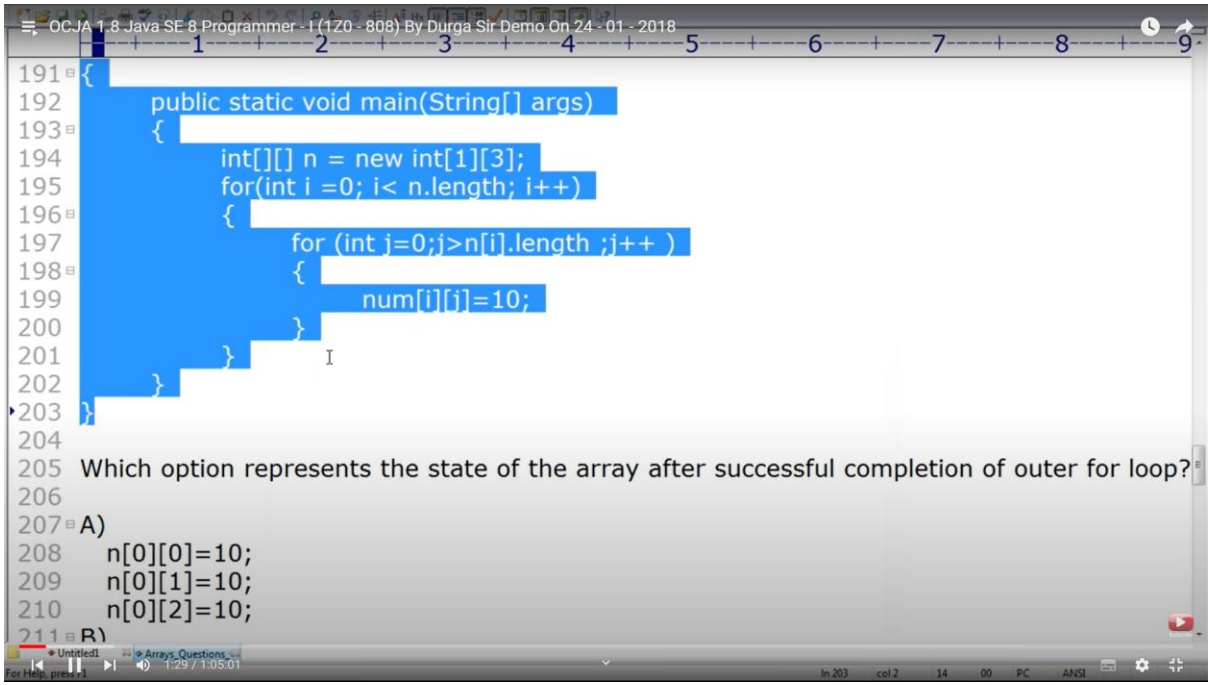

```
class Test
{
    public static void main(String[] args)
    {
     int[] x=new int[4]{10,20,30,40};
     System.out.println(x[0]);
    }

}
```



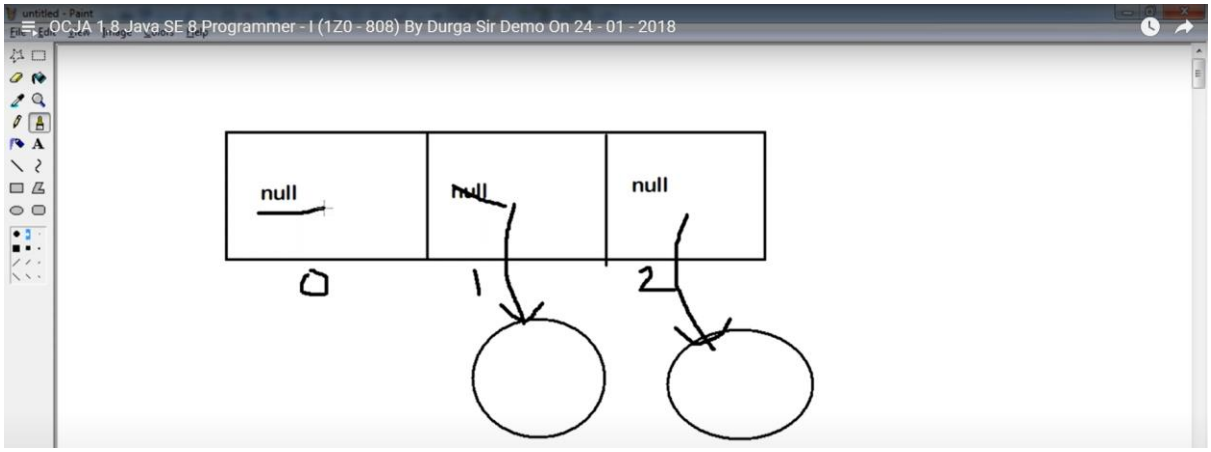

```
Test.java:16: error: class, interface, or enum expected
              }
              ^
Test.java:18: error: class, interface, or enum expected
       }
       ^
8 errors

D:\durgaclasses>javac Test.java

D:\durgaclasses>javac Test.java

D:\durgaclasses>java Test
10

D:\durgaclasses>javac Test.java
Test.java:5: error: ';' expected
    int[] x=   new int[4]{10,20,30,40};
                         ^
```

Here n[0].length = 3

For j=0 j>3 never going to happen. So default value will be saved. Inner loop not gonna get executed .Default value would be dispayed which is 0

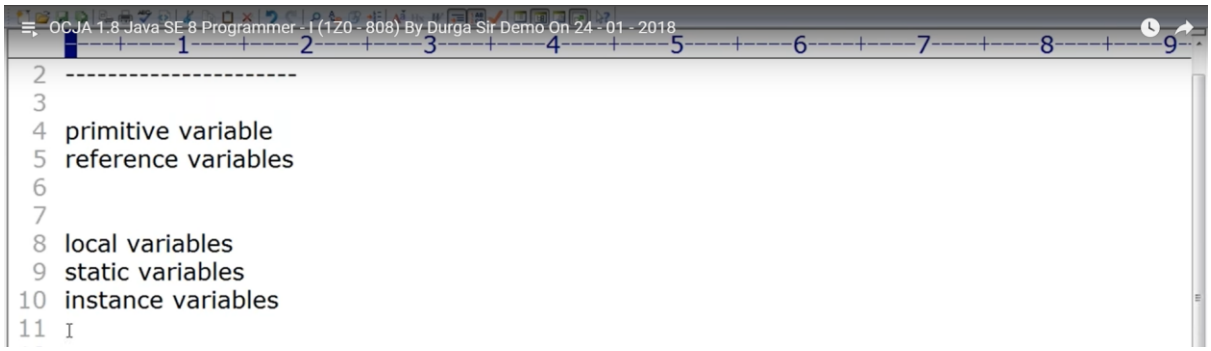


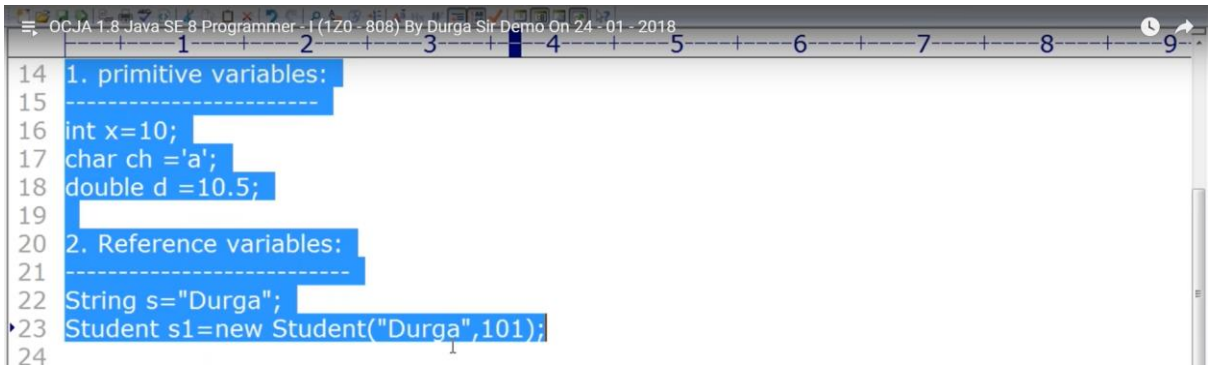
```java
class Student
{
    String name;
    public Student(String name)
    {
        this.name=name;
    }
}
public class Test
{
    public static void main(String[] args)
    {
        Student[] students = new Student[3];
        students[1]= new Student("Durga");
        students[2]= new Student("Ravi");
        for(Student s : students)
        {
            System.out.println(s.name);
        }
    }
```

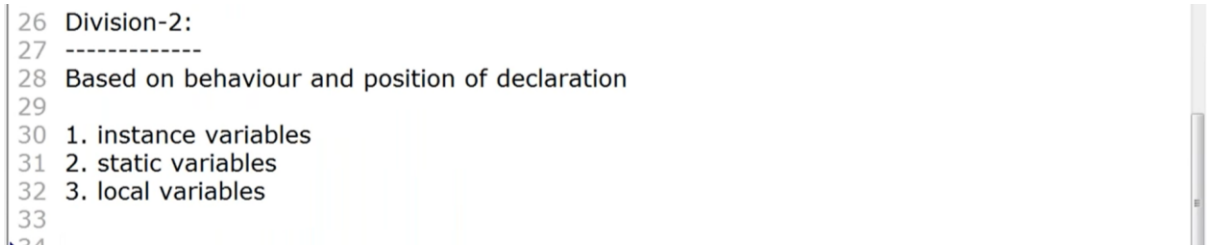


Null – calling name function on null. Null pointer exception. If we call any operation on null its gonna throw null pointer exception

```
D:\durgaclasses>java Test
Exception in thread "main" java.lang.NullPointerException
        at Test.main(Test.java:18)
```


OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 24 - 01 - 2018
----
primitive variable
reference variables
local variables
static variables
instance variables


OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 24 - 01 - 2018
1. primitive variables:
int x=10;
char ch ='a';
double d =10.5;
2. Reference variables:
String s="Durga";
Student s1=new Student("Durga",101);


Division-2:
Based on behaviour and position of declaration
1. instance variables
2. static variables
3. local variables

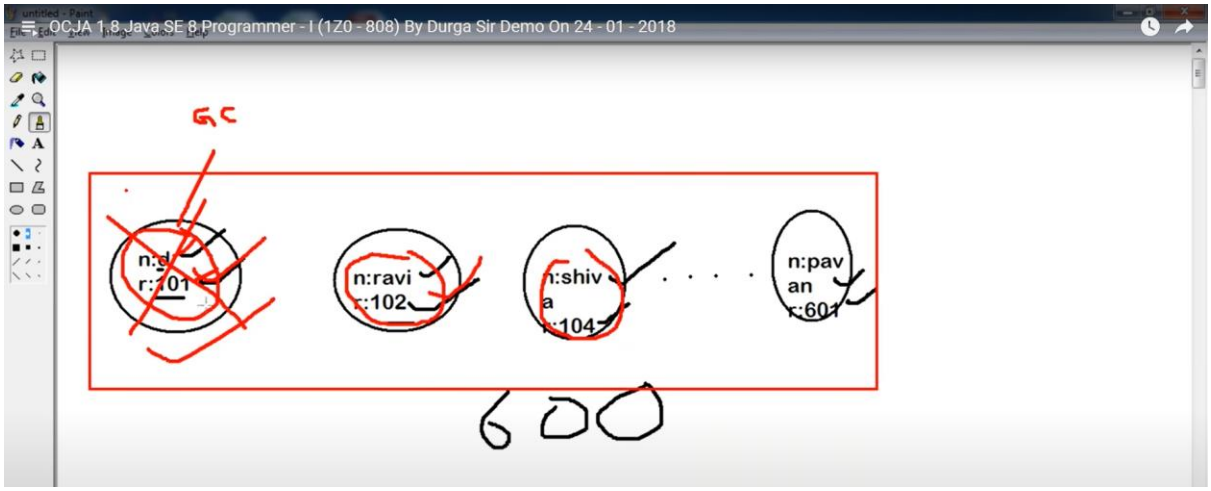
If the value of the variable is going to change for instance to instance its called instance variable.

Here name and roll no changes for every instance of Student. Instance variables should be declared inside the class not inside methods or constructors or blocks

Instance variables are created at the time of object creation. When the object is destroyed then instance variable with be destroyed

```
class Student
{
    String name;
    int rollno;
}

instance variables or object level variables...


Student s1 = new Student("Durga",101);
Student s2 = new Student("Ravi",102);
Student s3 = new Student("Shiva",103);
....600 students
```

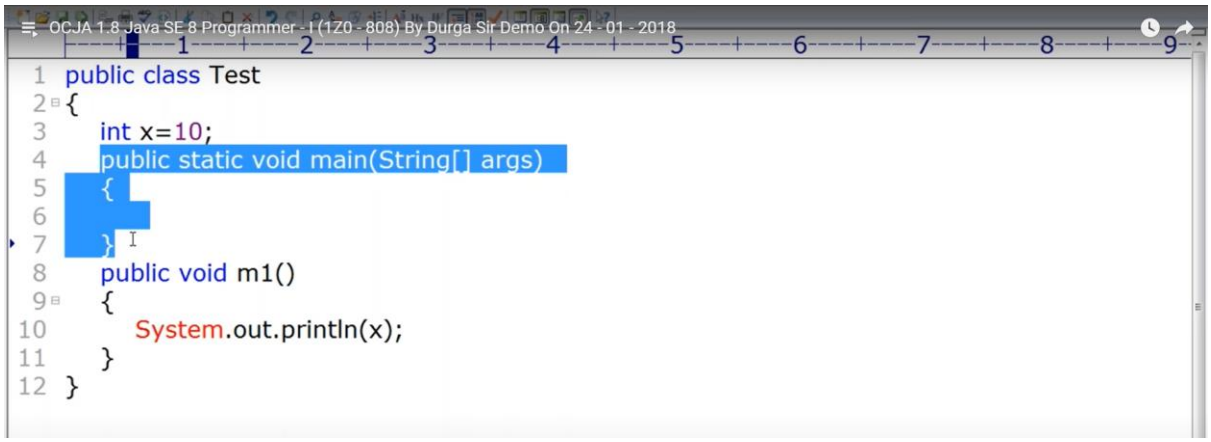
For every object separate copy will be created



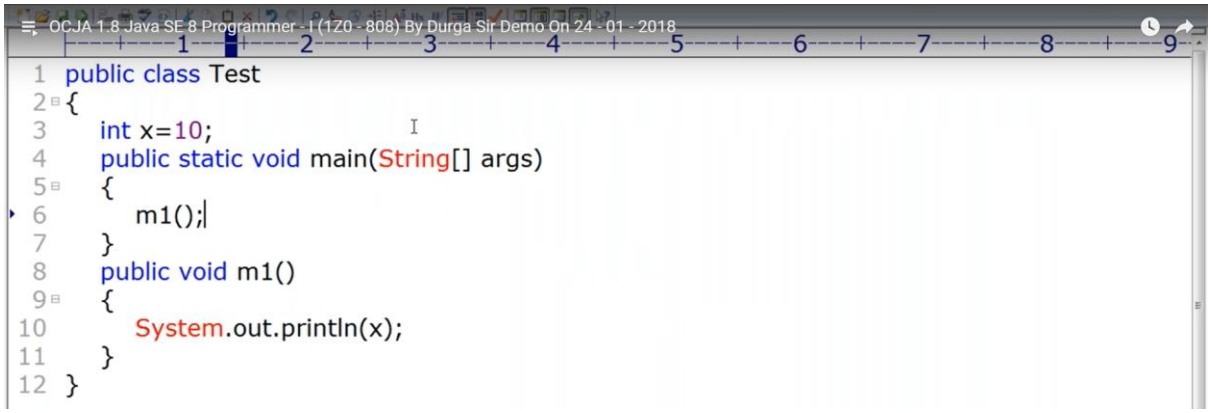
From static method we would not be able to access instance variable. The instance variable is part of object not class. Without object we cant access it from static method. Non static variable cannot be accessed from a static context.

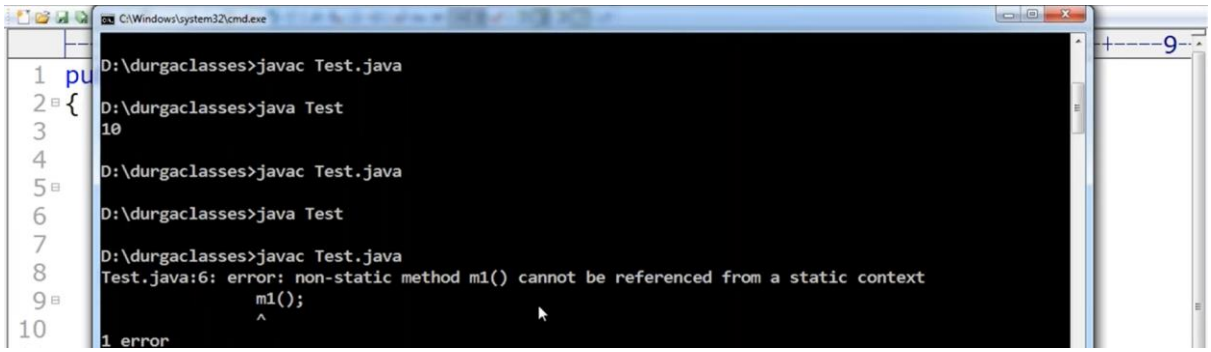
```
public class Test
{
    int x=10;
    public static void main(String[] args)
    {
        System.out.println(x);
    }
}
```

Instance variable can be accessed from instance area. This is right but main method will not provide any output

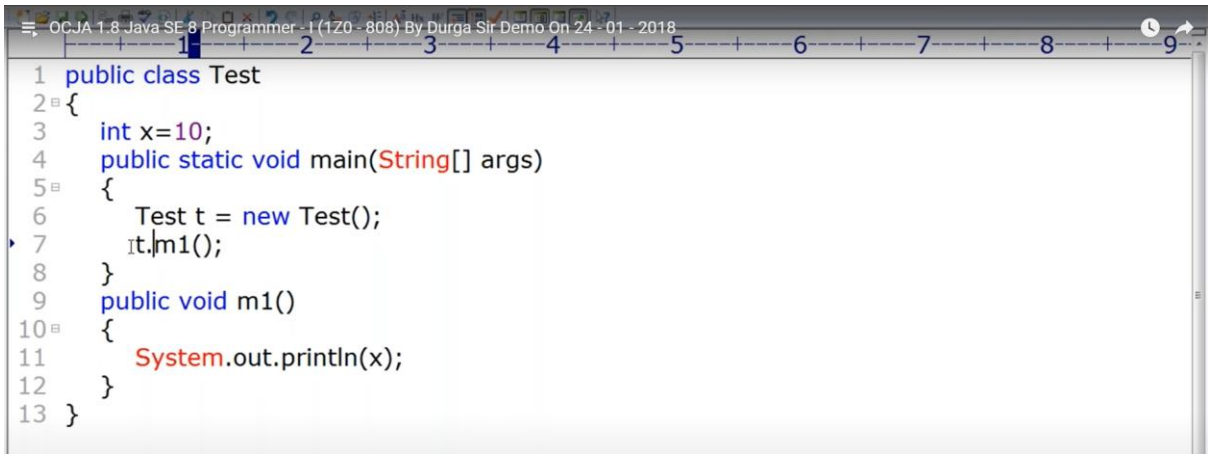


Static method cannot access instance method or variable directly.

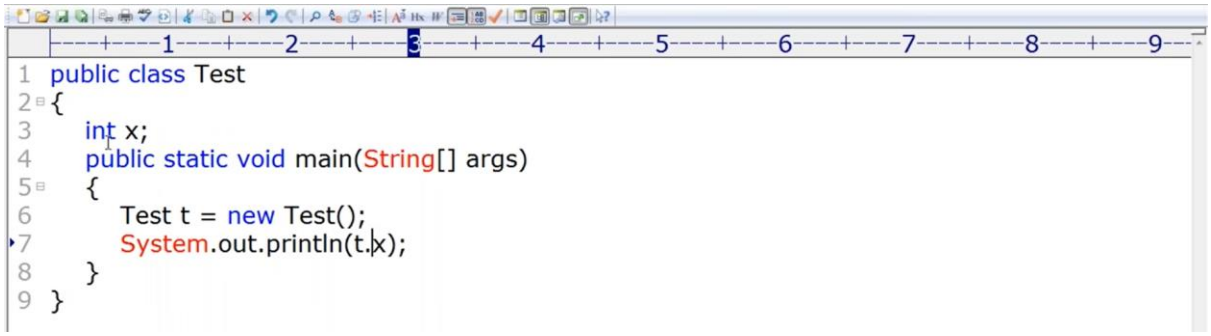


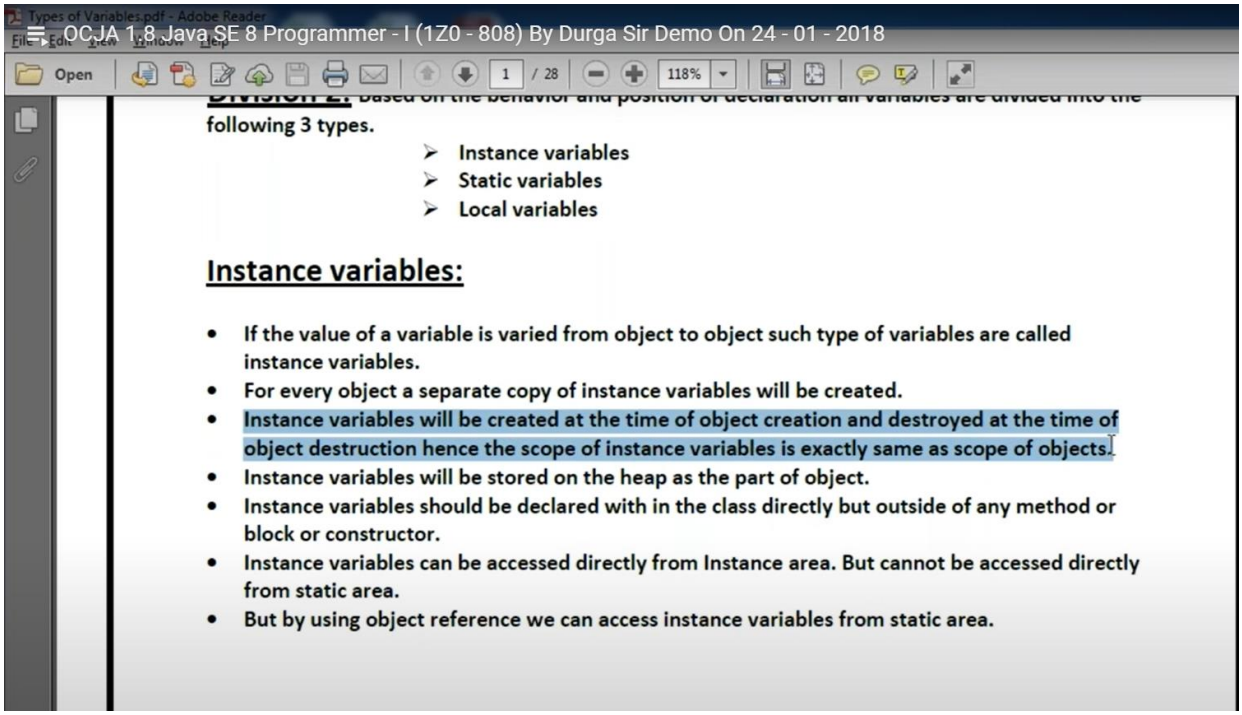


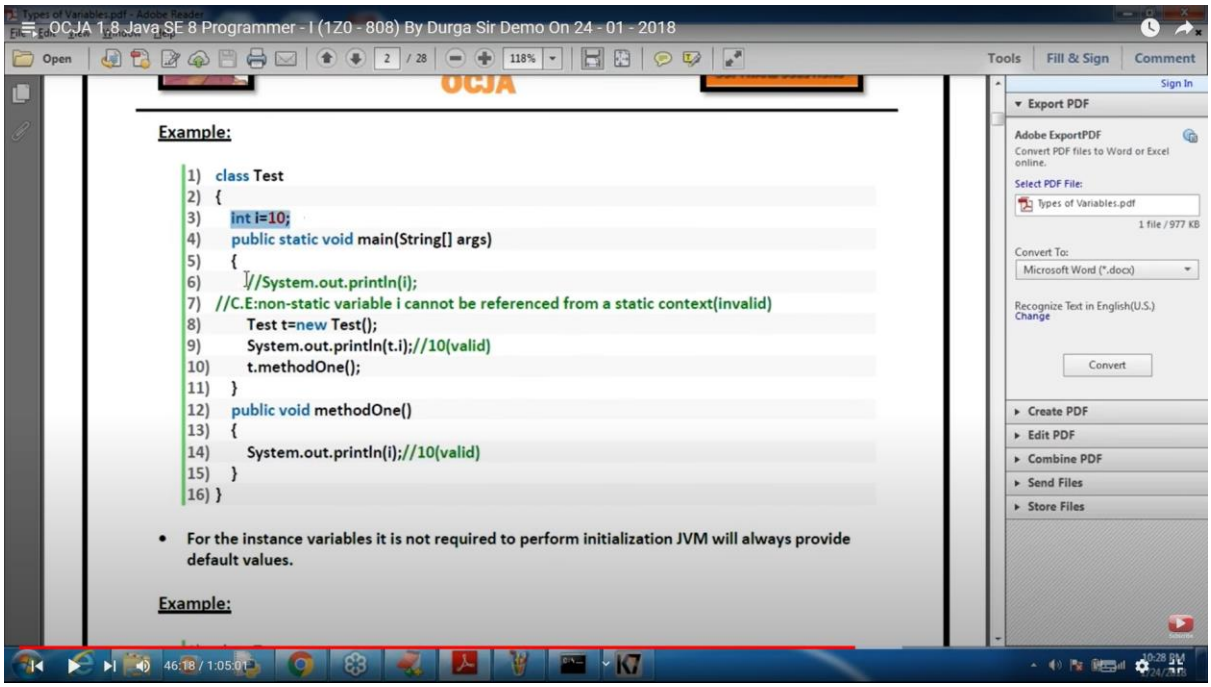
To access it you need create an object and access it. We can access instance variable from static area using object ref

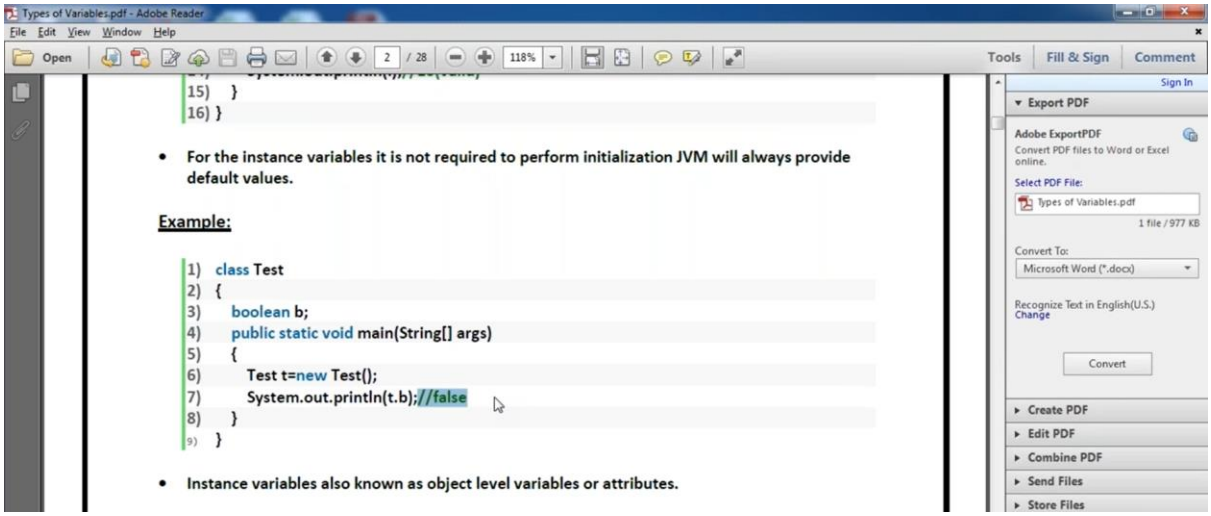


For instance variable we don't need to initialize. JVM will provide the default value explicitly. If we don't want default the we can specify default variable

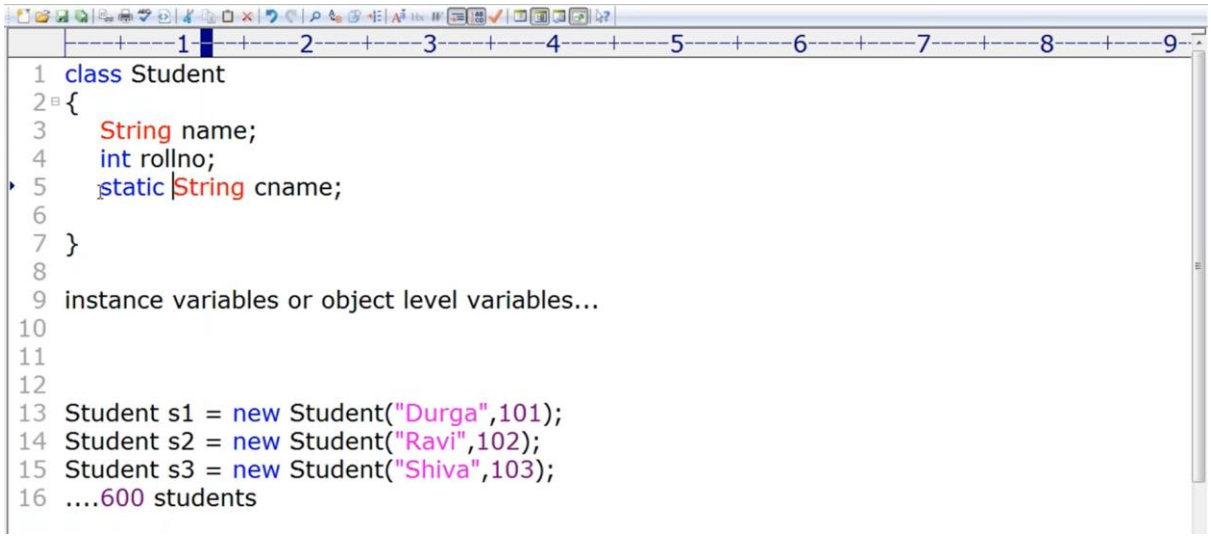



OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 24 - 01 - 2018
Types of Variables.pdf - Adobe Reader
Open
1 / 28
118%
following 3 types.
Instance variables
Static variables
Local variables
Instance variables:
If the value of a variable is varied from object to object such type of variables are called instance variables.
For every object a separate copy of instance variables will be created.
Instance variables will be created at the time of object creation and destroyed at the time of object destruction hence the scope of instance variables is exactly same as scope of objects.
Instance variables will be stored on the heap as the part of object.
Instance variables should be declared with in the class directly but outside of any method or block or constructor.
Instance variables can be accessed directly from Instance area. But cannot be accessed directly from static area.
But by using object reference we can access instance variables from static area.


OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 24 - 01 - 2018
Open
2 / 28
118%
Tools
Fill & Sign
Comment
OCJA
Example:
1) class Test
2) {
3) int i=10;
4) public static void main(String[] args)
5) {
6) //System.out.println(i);
7) //C.E:non-static variable i cannot be referenced from a static context(invalid)
8) Test t=new Test();
9) System.out.println(t.i);//10(valid)
10) t.methodOne();
11) }
12) public void methodOne()
13) {
14) System.out.println(i);//10(valid)
15) }
16) }
For the instance variables it is not required to perform initialization JVM will always provide default values.
Example:
Export PDF
Adobe ExportPDF
Convert PDF files to Word or Excel online.
Select PDF File:
Types of Variables.pdf
1 file / 977 KB
Convert To:
Microsoft Word (*.docx)
Recognize Text in English(U.S.)
Change
Convert
Create PDF
Edit PDF
Combine PDF
Send Files
Store Files
46:18 / 1:05:01



If its static these are class level variable (information). For all 600 object only 1 copy of cname will be created at class level and can be accessed by all the objects. To save memory.. if the value does not change for object to object then that variable can be defined as static



Object level variables – Instance Variable

Class level variables – Static

For all the object if the variable value is unique – instance variable

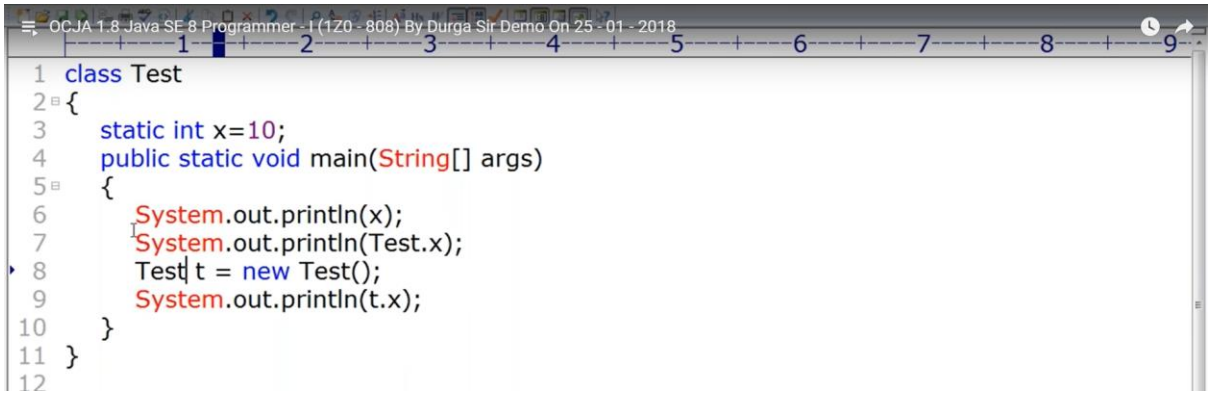
Static variable are created when the class is loading. And destroyed at time of class unloading

When ever we execute the class file.

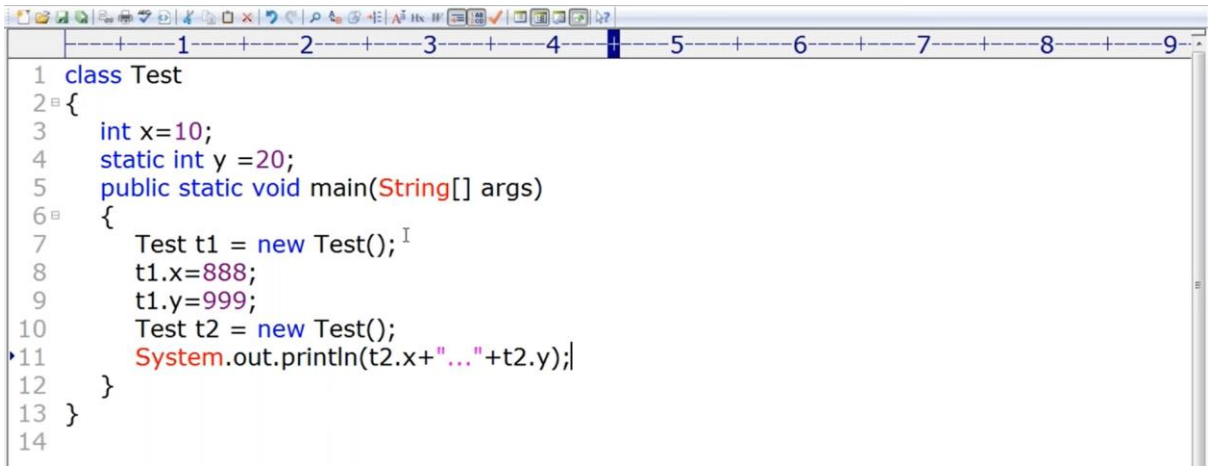
Staic variable will be created when the Test.class is loaded and destroyed when Test.class is unloaded

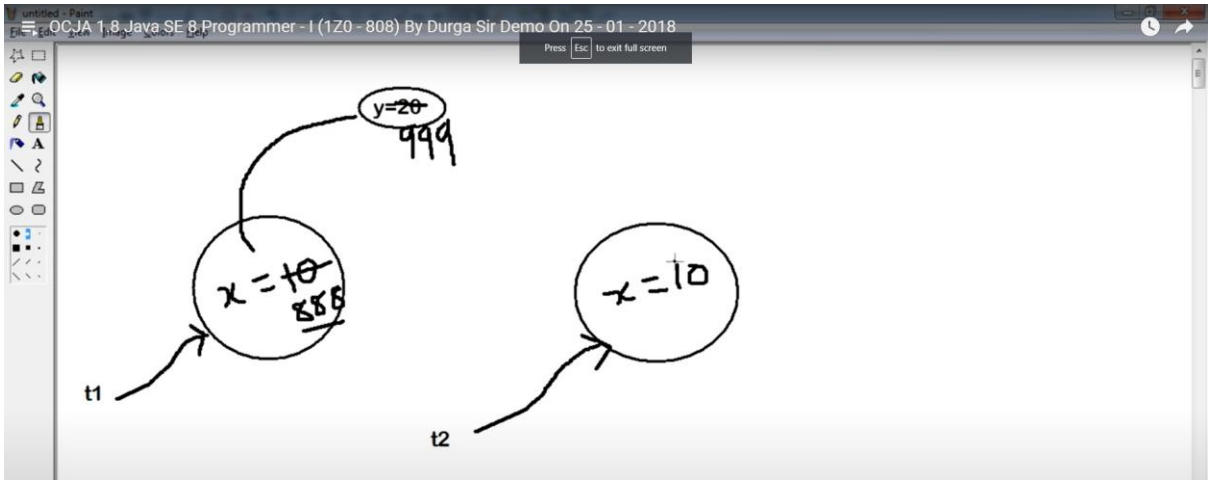
```
D:\durgaclasses>java Test

1. JVM will be started
2. Create and Start Main Thread by JVM
3. Main Thread will search for Test.class==>(NoClassDefFoundError)
4. Main Thread will load Test.class
5. Execute main method
6. unload Test.class file
7. Terminate Main Thread
8. JVM will be shutdown
```

Static variable can be accessed via object ref but not advisable. Line 9



10 …999. For all object ref a sep copy of instance varaiable will be there a common copy of static variable will be there. Any change to static variable will be ref across all objects ref





20 and null pointer exception. Static needs no object creation

```
class Test
{
    int x=10;
    static int y =20;
    public static void main(String[] args)
    {
        Test t = null;
        System.out.println(t.y);
        System.out.println(t.x);

    }
}
```

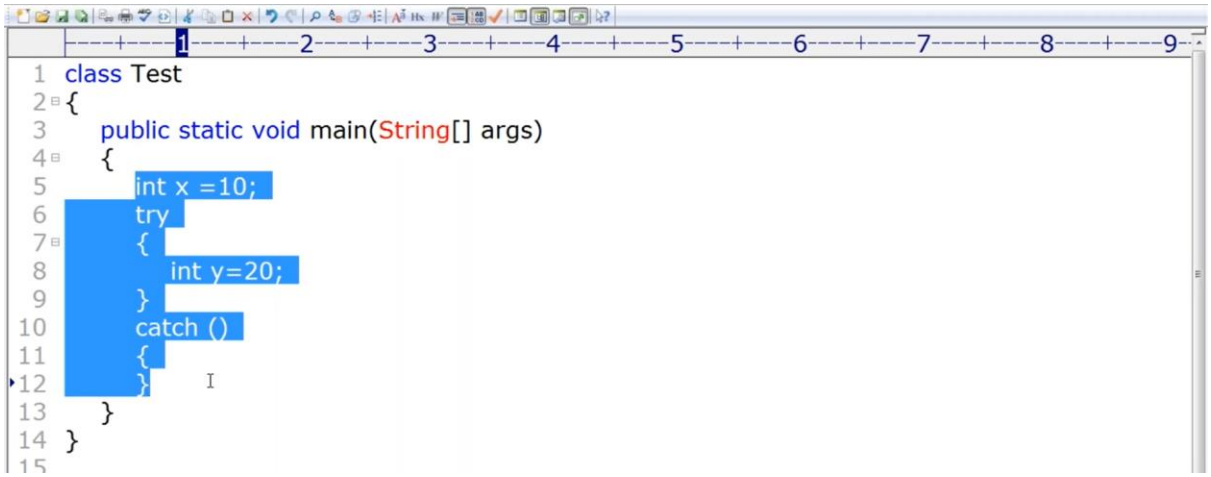
```
D:\durgaclasses>javac Test.java

D:\durgaclasses>java Test
20
Exception in thread "main" java.lang.NullPointerException
        at Test.main(Test.java:9)
```

```
----------------------
3. local variables:
----------------------
a variable inside method|block|constructor local variables
```

Local varibale will be created when the method block or const is executed and will be destroyed upn completion.

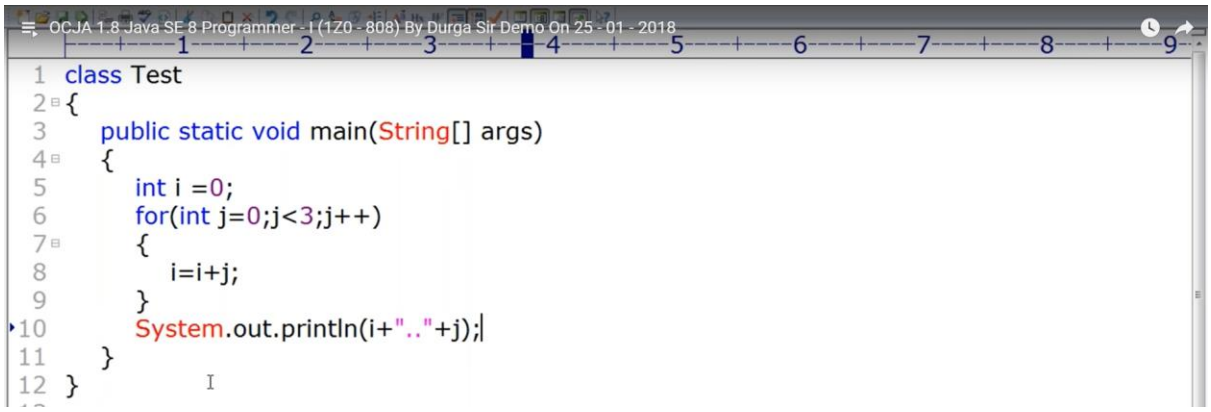
```
class Test
{
    public static void main(String[] args)
    {
        int x=10;
    }
}
```

x-Scope within main and y- scope within try block

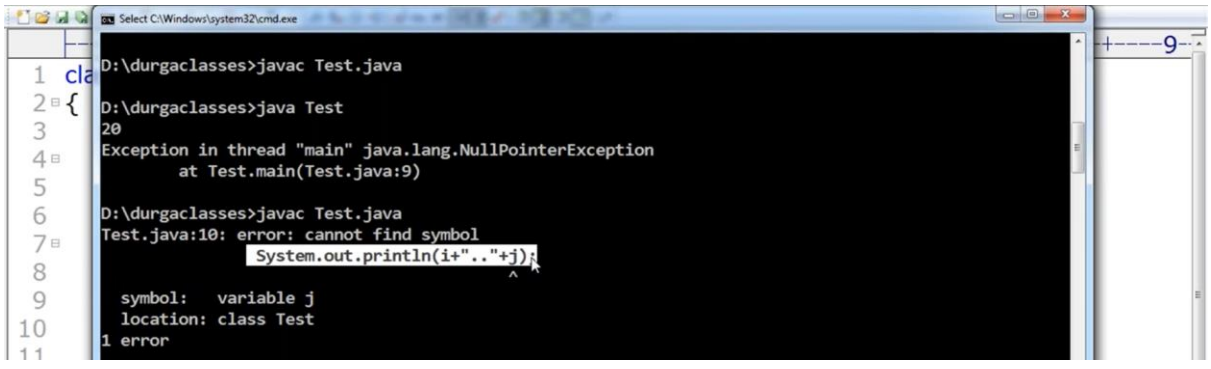


```
class Test
{
    public static void main(String[] args)
    {
        int x =10;
        try
        {
            int y=20;
        }
        catch ()
        {
        }
    }
}
```

Compile time error- J is accessed outside for loop




```
class Test
{
    public static void main(String[] args)
    {
        int i =0;
        for(int j=0;j<3;j++)
        {
            i=i+j;
        }
        System.out.println(i+".."+j);
    }
}
```



2 compilation error x- is local to try and cant be accessed from catch and main

```
1  class Test
2  {
3      public static void main(String[] args)
4      {
5          try
6          {
7              int x =Integer.parseInt("ten");
8          }
9          catch (NumberFormatException e)
10         {
11             x=10;
12         }
13         System.out.println(x);
14     }
15 }
```

For local variable JVM will not provide temp value , This is very local and temp.

OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 25 - 01 - 2018

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

Before using local variable we nee to initialize , if we are not using we don't hv to initialize

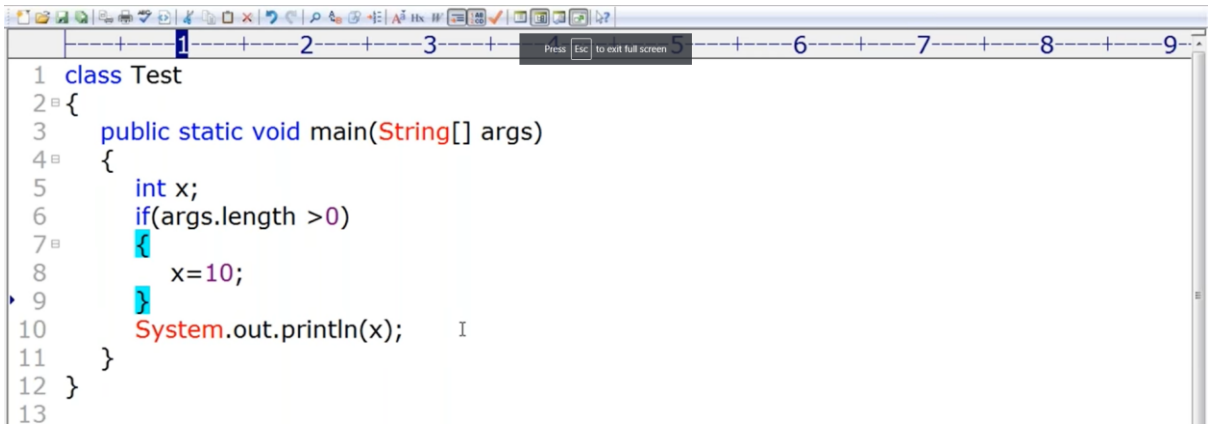
```
D:\durgaclasses>java Test
Hello dont sleep
```

Below compile time local variable should be initialized before using

OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 25 - 01 - 2018

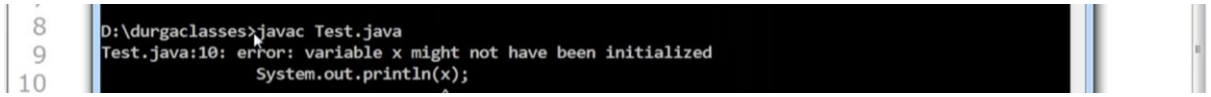
```
1  class Test
2  {
3      public static void main(String[] args)
4      {
5          int x;
6          System.out.println(x);
7      }
8  }
```

The if will be executed only when there are command line args . Compile error. It cant assume that run time args will be passed



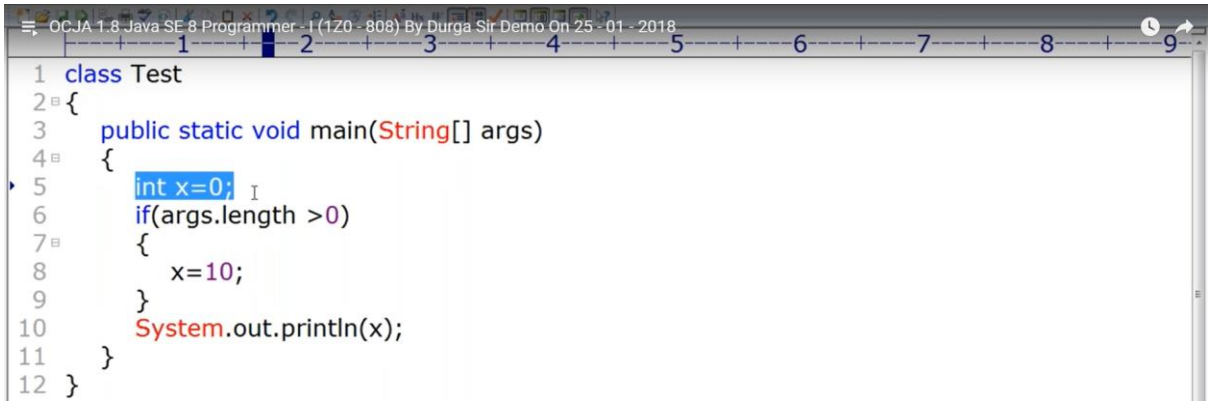
```
1  class Test
2  {
3     public static void main(String[] args)
4     {
5        int x;
6        if(args.length >0)
7        {
8           x=10;
9        }
10       System.out.println(x);
11    }
12 }
13
```

Not recommended to initialize local variable inside logical blocks. Good to initialize beginning with default value.



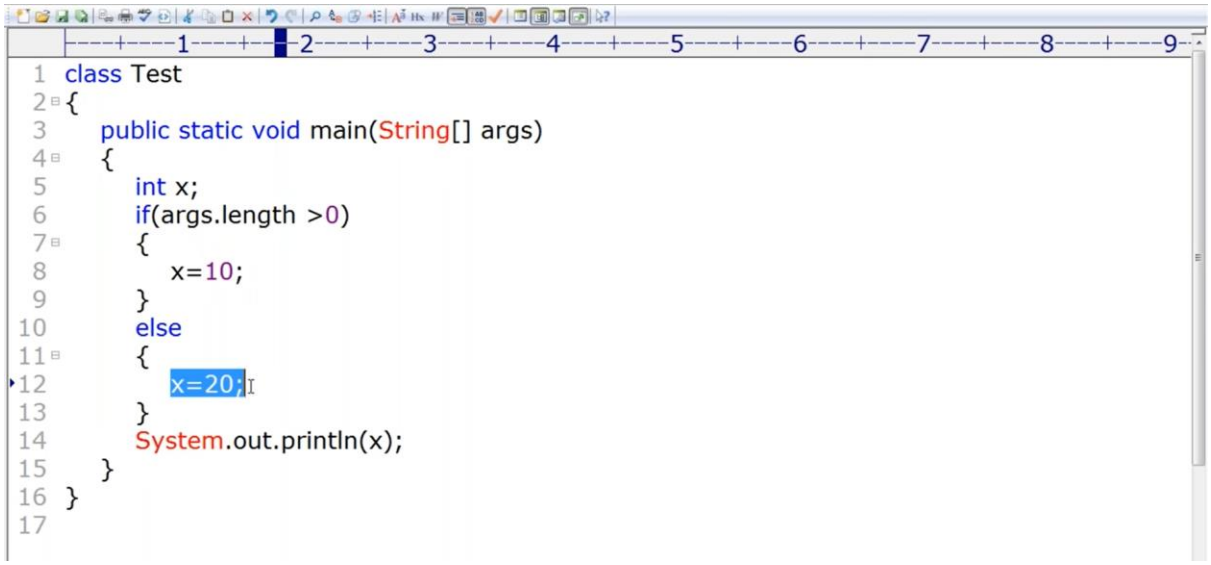
```
D:\durgaclasses>javac Test.java
Test.java:10: error: variable x might not have been initialized
                System.out.println(x);
```

Below pass, default value is passeds



```
1  class Test
2  {
3     public static void main(String[] args)
4     {
5        int x=0;
6        if(args.length >0)
7        {
8           x=10;
9        }
10       System.out.println(x);
11    }
12 }
```

X value is always initialized . so compiler knows that .

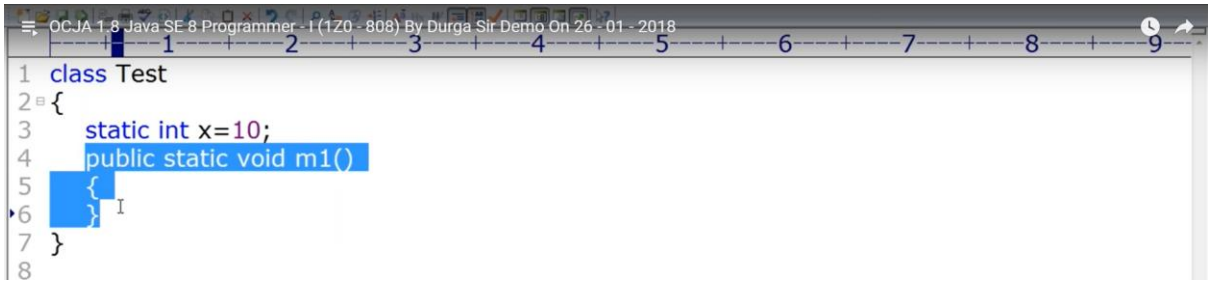


```
1  class Test
2  {
3      public static void main(String[] args)
4      {
5          int x;
6          if(args.length >0)
7          {
8              x=10;
9          }
10         else
11         {
12             x=20;
13         }
14         System.out.println(x);
15     }
16 }
17
```

Local variables before using should always be initialized

A top-level class cannot be defined as static but inner level class can be defined as static. Variables and methods can be declared as static.

The variable and method are class level no object instance is needed




```
1  class Test
2  {
3      static int x=10;
4      public static void m1()
5      {
6      }
7  }
8
```

Any person is added to access this class and its members

public class Test{

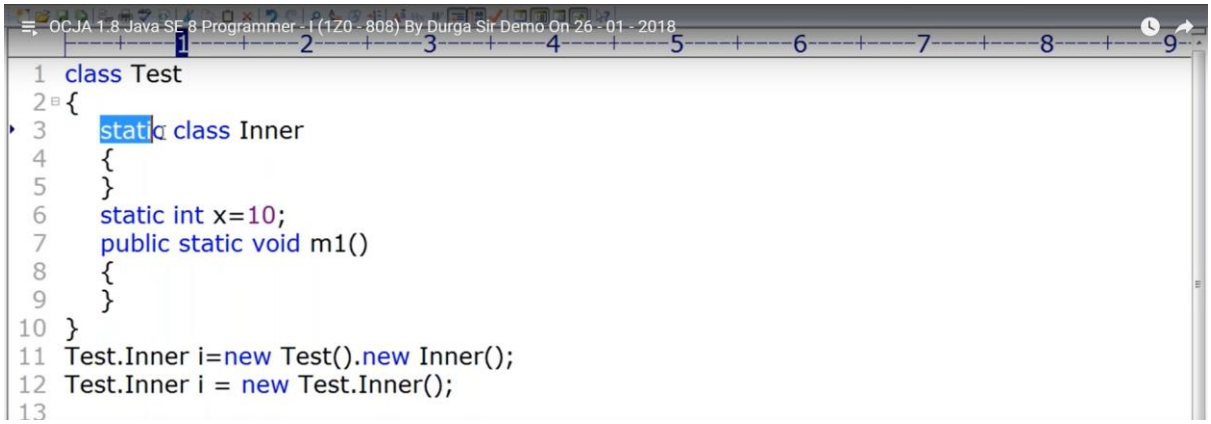
final class Test{ - cant create child class

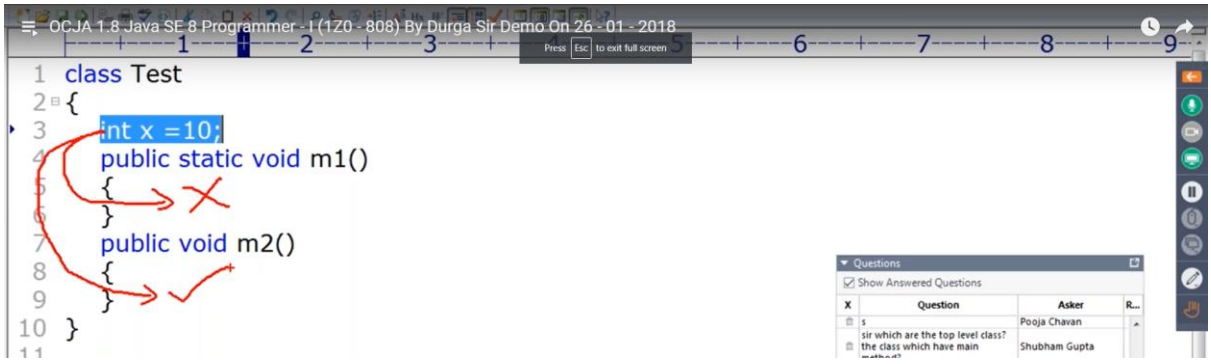
static class Test{ - Static means only 1 copy is maintained – when we define class static it does not provide any value or meaning

the below is an inner class. If want to create an object of inner class we need to create an object of outer class.

\

If we define the inner class as static we can create an object of inner class without creating an object of outerclass



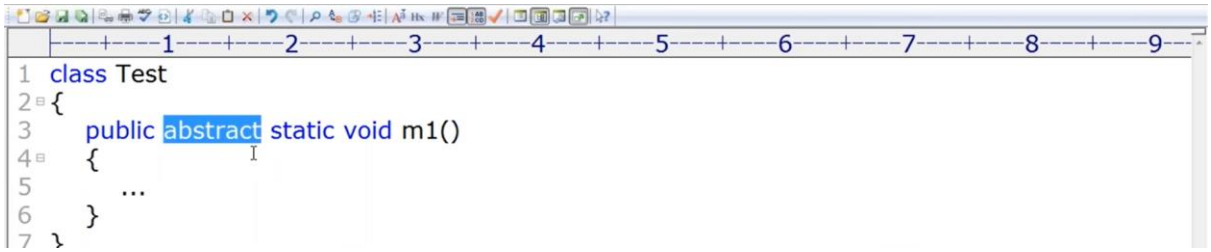


Static method should always have an implementation cant be emtpty or without body

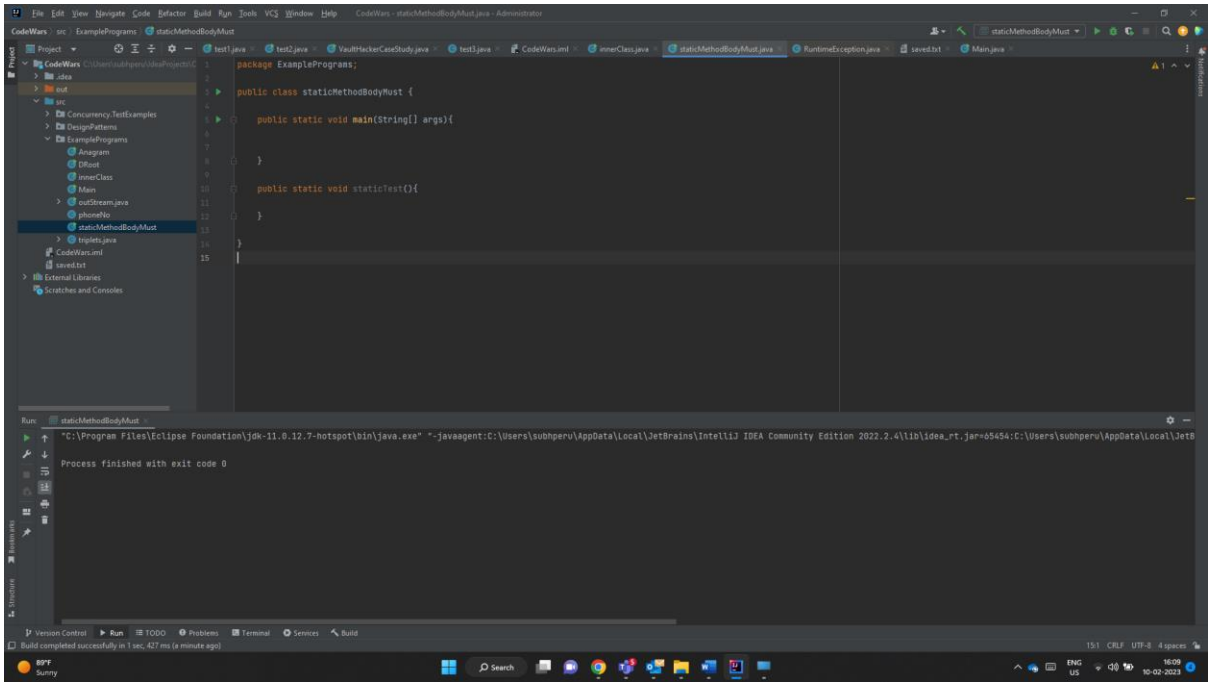
Abstract method cannot have an implementation, static method should have an implementation

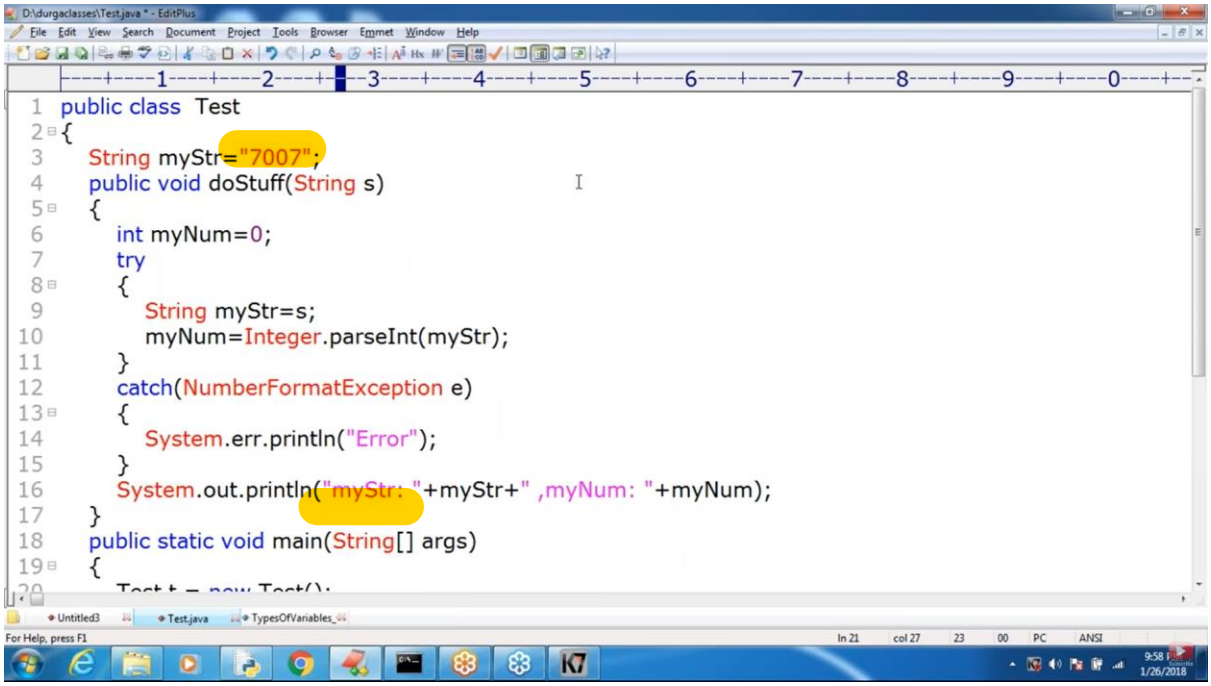
So abstract method can be static. Illegal combination of modifiers.

.

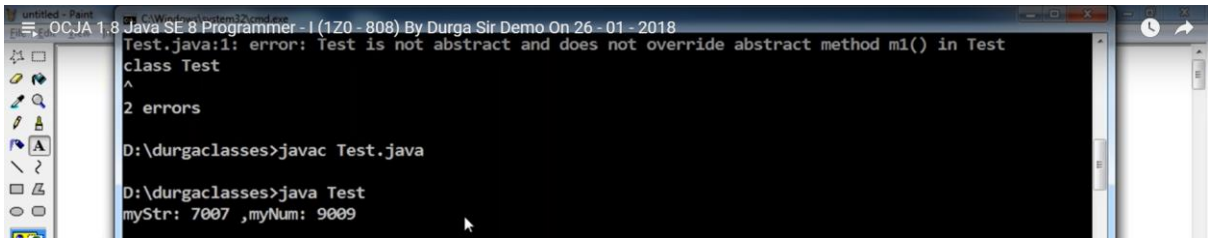


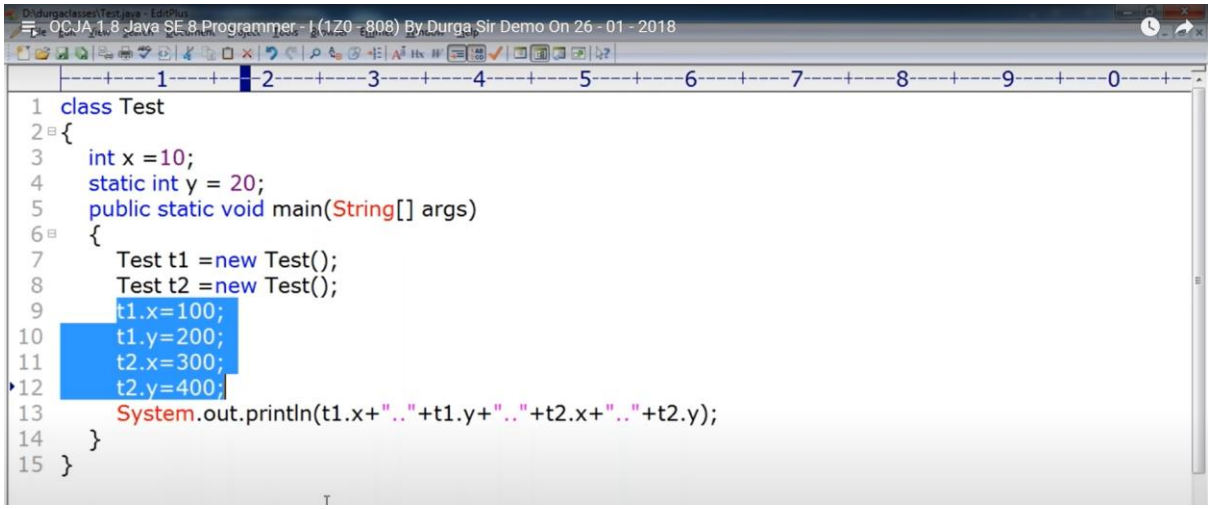
The below works static method the implementation is empty





```
1  public class  Test
2  {
3     String myStr="7007";
4     public void doStuff(String s)
5     {
6        int myNum=0;
7        try
8        {
9           String myStr=s;
10          myNum=Integer.parseInt(myStr);
11       }
12       catch(NumberFormatException e)
13       {
14          System.err.println("Error");
15       }
16       System.out.println("myStr: "+myStr+" ,myNum: "+myNum);
17    }
18    public static void main(String[] args)
19    {
```



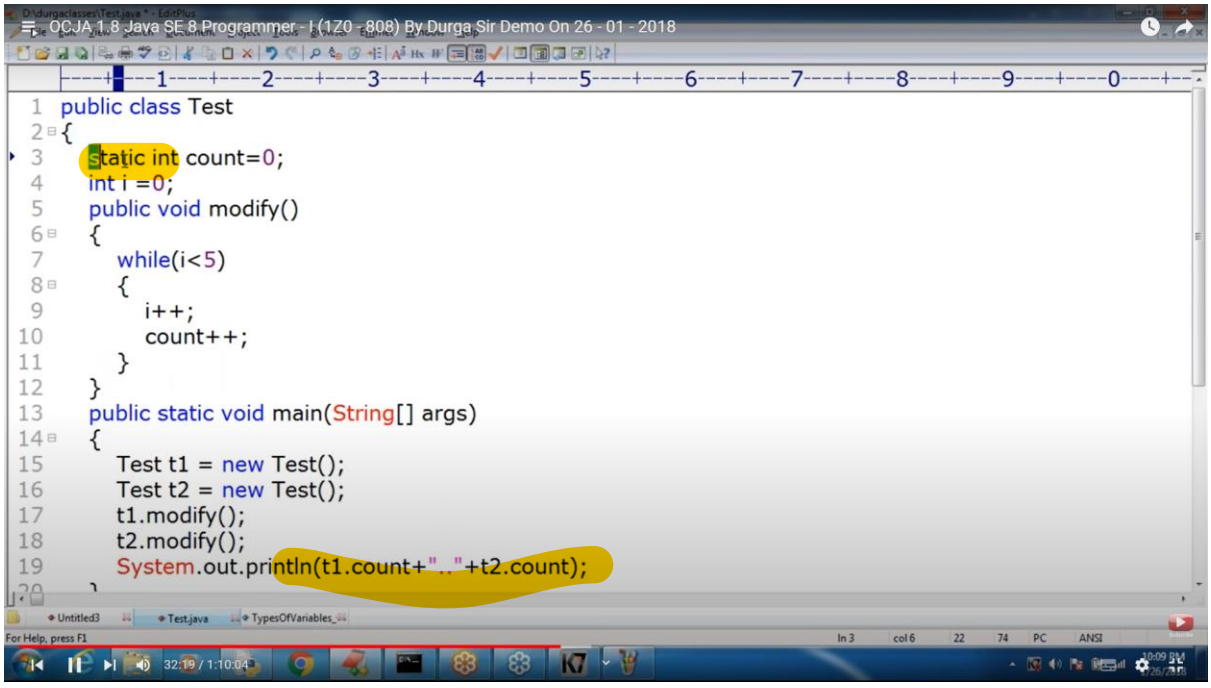



```
1  class Test
2  {
3     int x =10;
4     static int y = 20;
5     public static void main(String[] args)
6     {
7        Test t1 =new Test();
8        Test t2 =new Test();
9        t1.x=100;
10       t1.y=200;
11       t2.x=300;
12       t2.y=400;
13       System.out.println(t1.x+".."+t1.y+".."+t2.x+".."+t2.y);
14    }
15 }
```

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

D:\durgaclasses>java Test
100..400..300..400
```

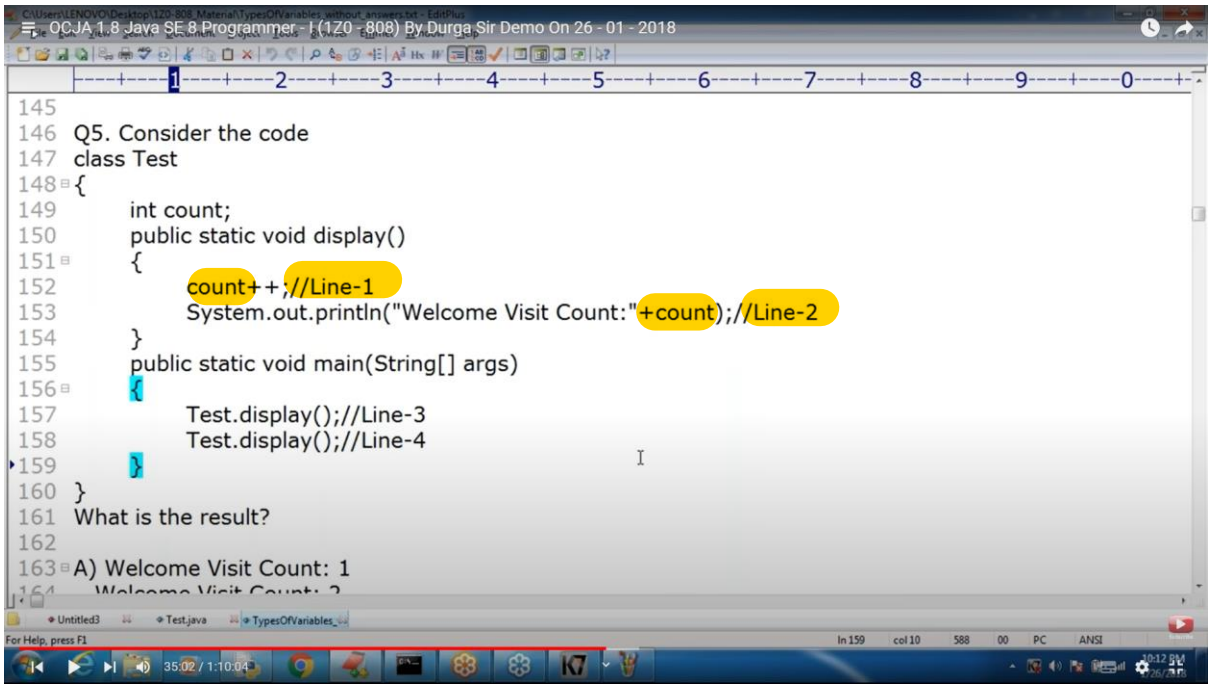
5 4 5 6

```
1  public class Test
2  {
3     static int x;
4     int y;
5     public static void main(String[] args)
6     {
7        Test t1 = new Test();
8        Test t2 = new Test();
9        t1.x=3;
10       t1.y=4;
11       t2.x=5;
12       t2.y=6;
13       System.out.println(t1.x+":"+t1.y+":"+t2.x+":"+t2.y);
14    }
15 }
```

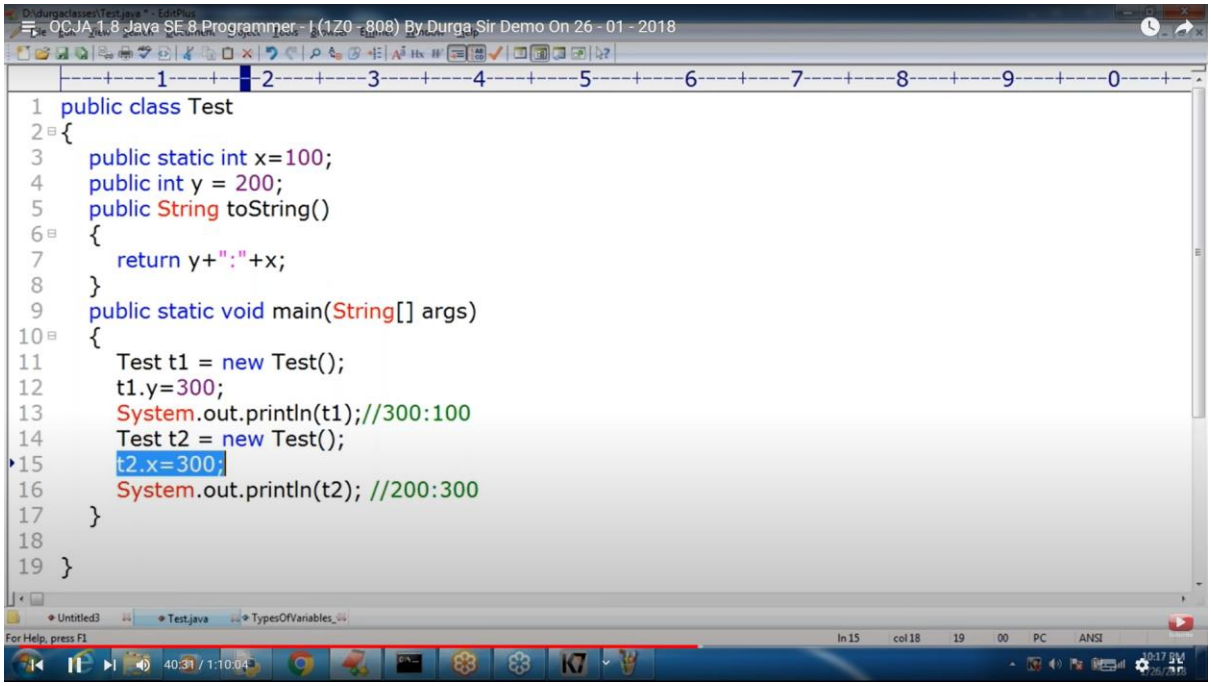


```
138
139 What is the result?
140 A) 10..10
```

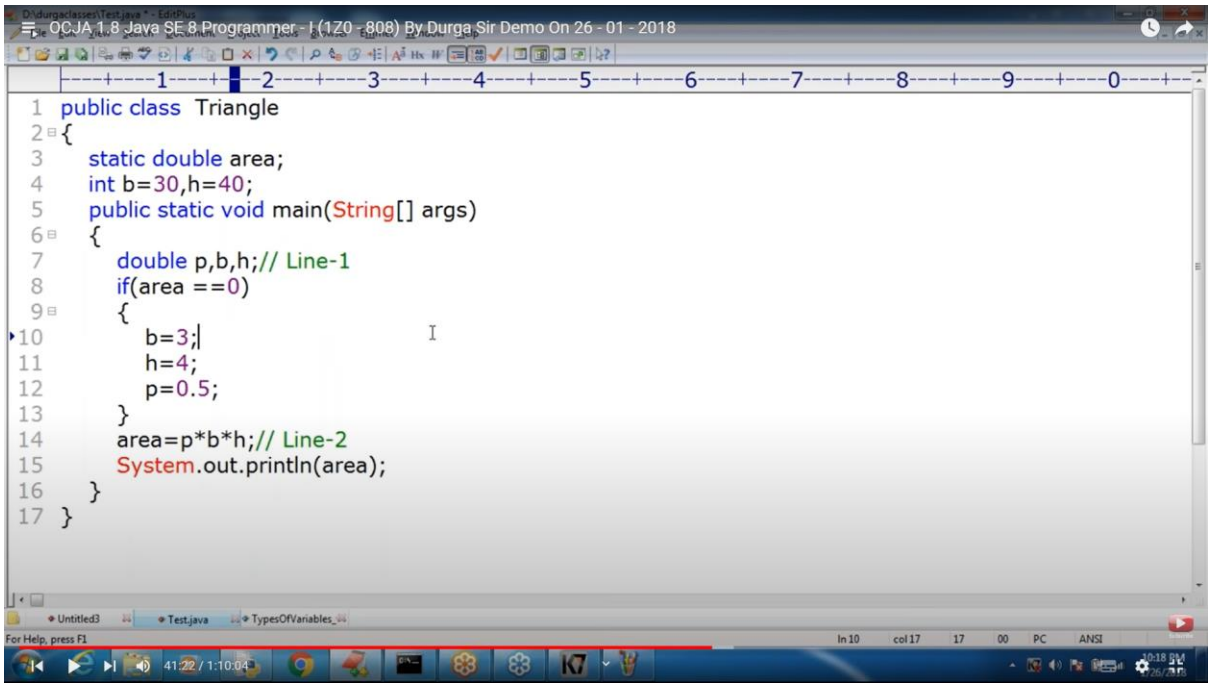
Instance variable cant be accessed in static method. Count

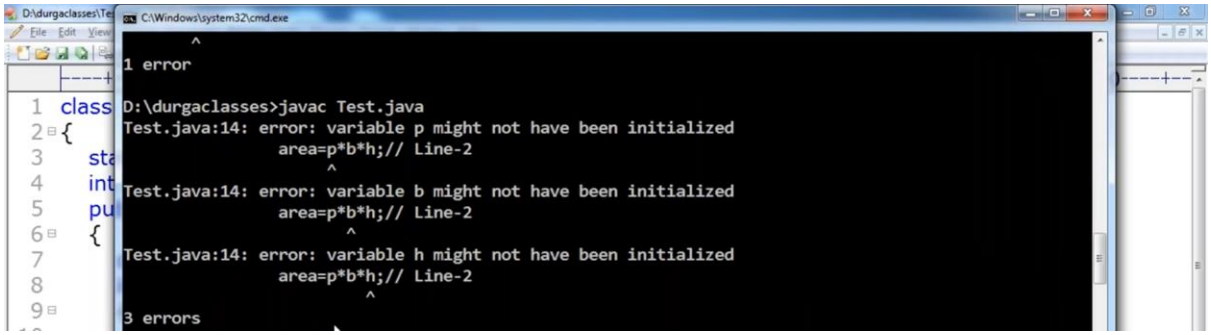


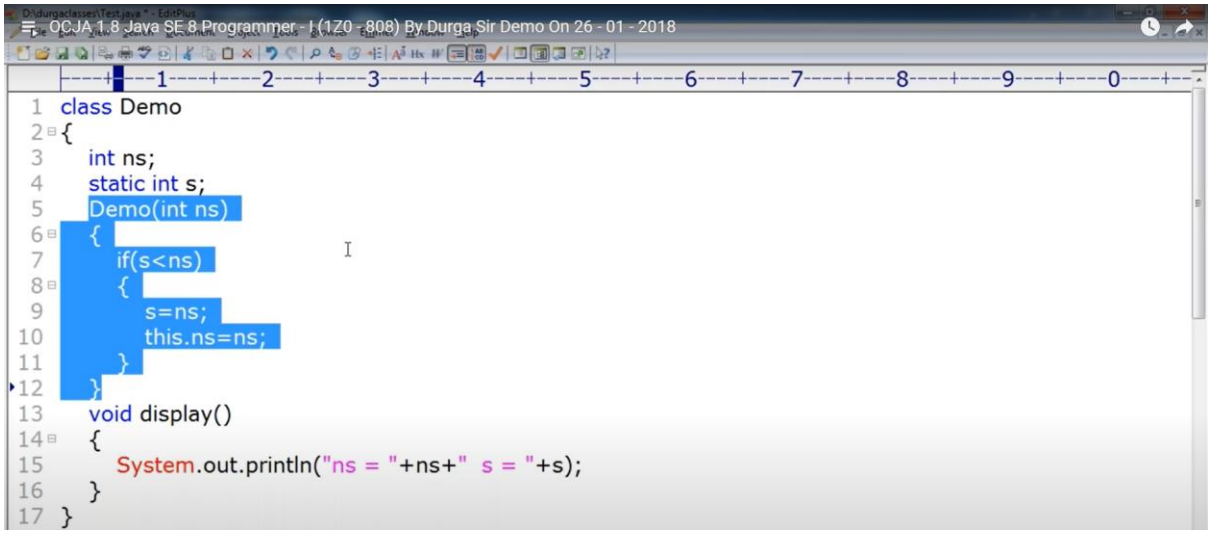

```
161 What is the result?
162
163 A) Welcome Visit Count: 1
164    Welcome Visit Count: 2
165 A) Welcome Visit Count: 1
166    Welcome Visit Count: 1
167 C) Compilation Fails at Line-1 and Line-2
168 D) Compilation Fails at Line-3 and Line-4
```

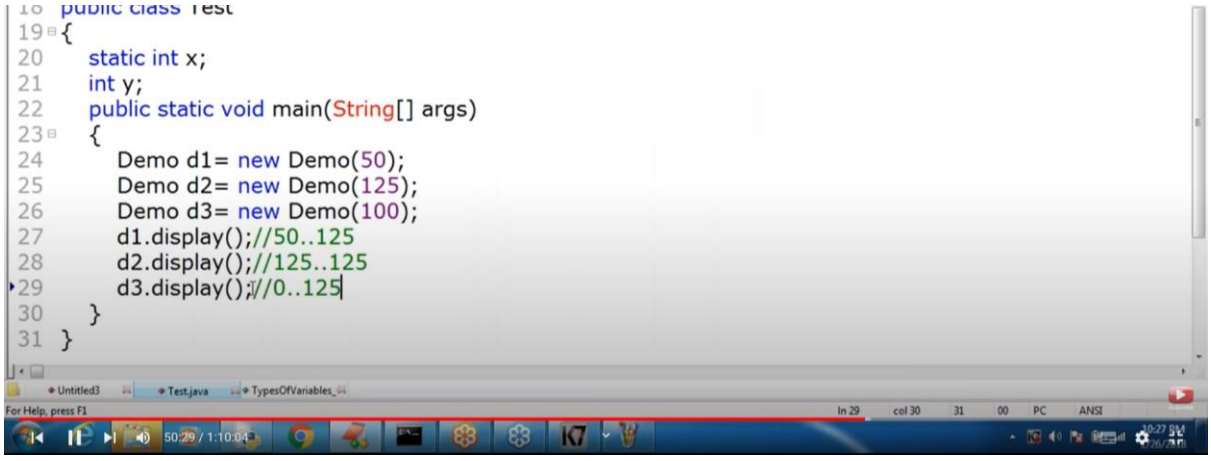


Local variables should always be initialized before use. The if block can or cannot be executed so compiler throws error at line 2.

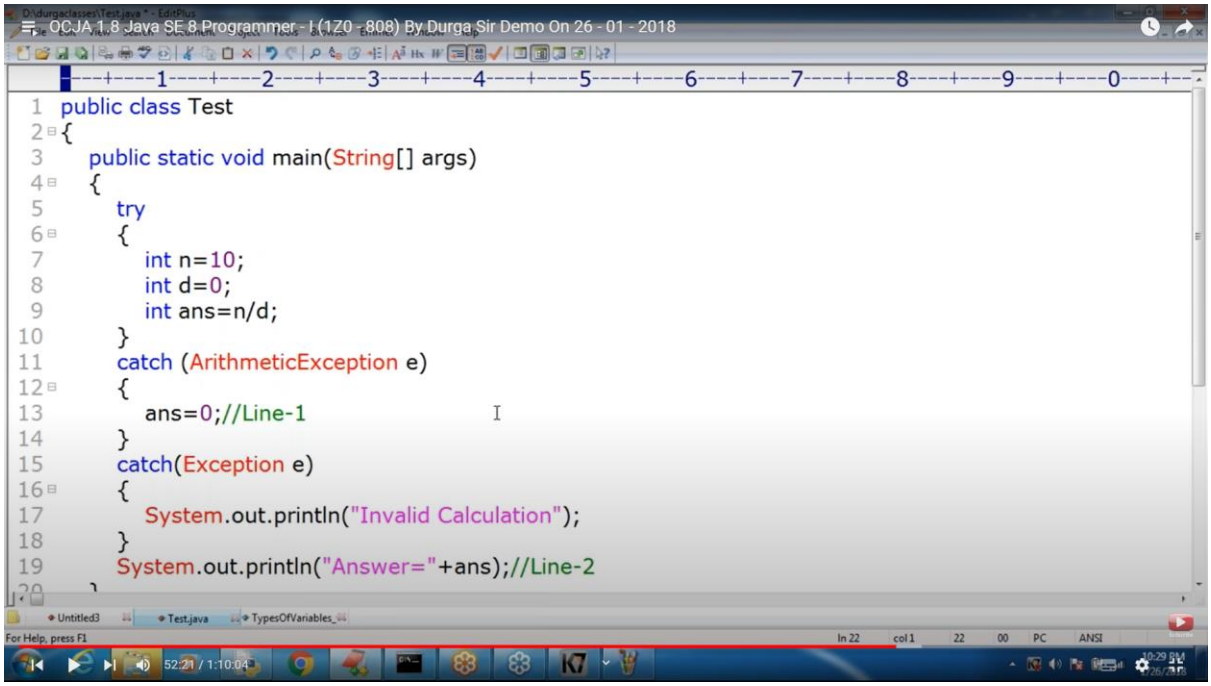


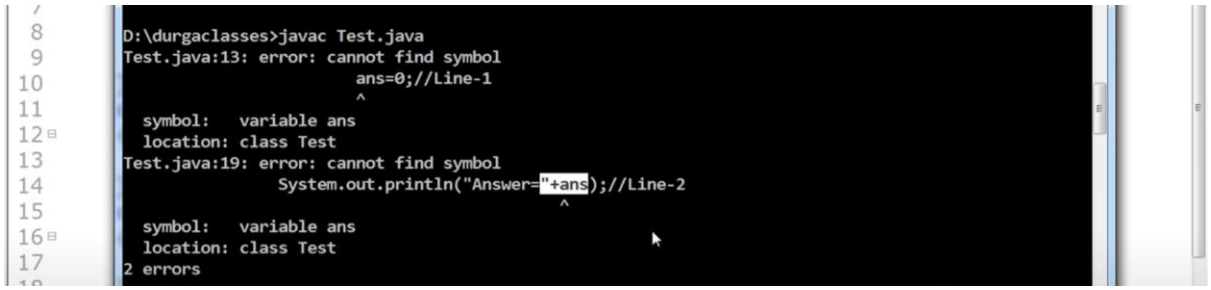

C:\Windows\system32\cmd.exe
^
1 error
D:\durgaclasses>javac Test.java
Test.java:14: error: variable p might not have been initialized
area=p*b*h;// Line-2
^
Test.java:14: error: variable b might not have been initialized
area=p*b*h;// Line-2
^
Test.java:14: error: variable h might not have been initialized
area=p*b*h;// Line-2
^
3 errors


OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) By Durga Sir Demo On 26 - 01 - 2018
class Demo
{
int ns;
static int s;
Demo(int ns)
{
if(s<ns)
{
s=ns;
this.ns=ns;
}
}
void display()
{
System.out.println("ns = "+ns+" s = "+s);
}
}

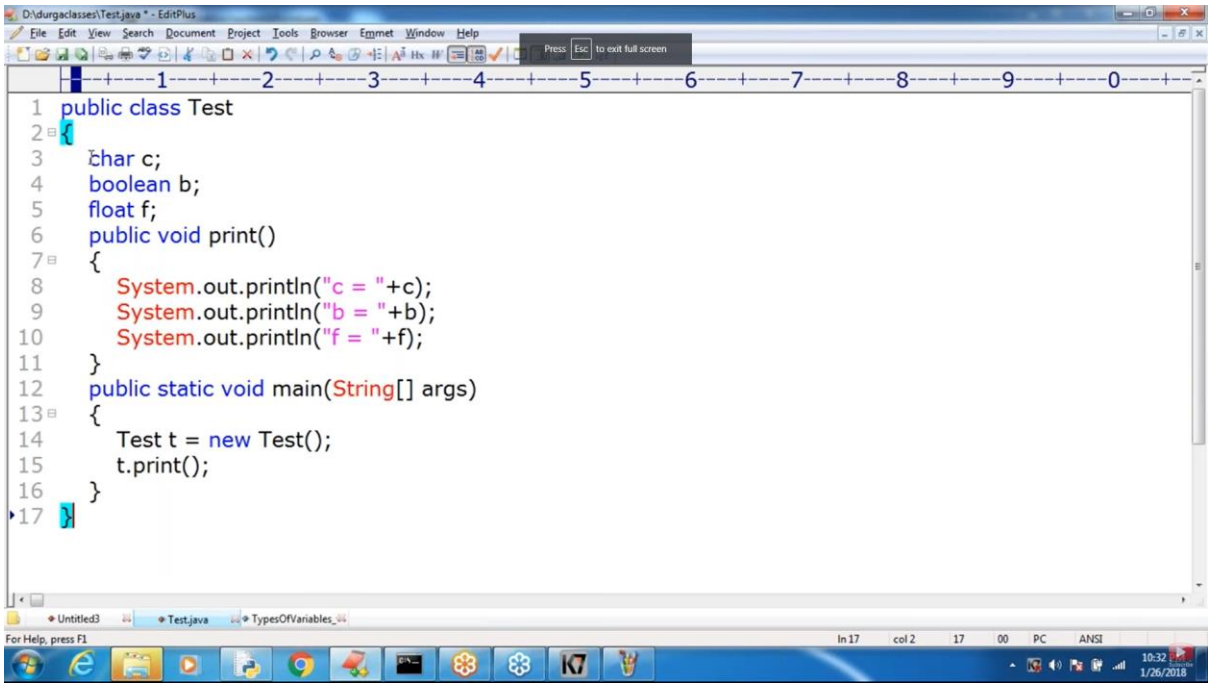

public class Test
{
static int x;
int y;
public static void main(String[] args)
{
Demo d1= new Demo(50);
Demo d2= new Demo(125);
Demo d3= new Demo(100);
d1.display();//50..125
d2.display();//125..125
d3.display();//0..125
}
}
Untitled3
Test.java
TypesOfVariables_
For Help, press F1
ln 29
col 30
31
00
PC
ANSI
50:29 / 1:10:04

Compile error in line1 and Line2





Space is the default value for char " "

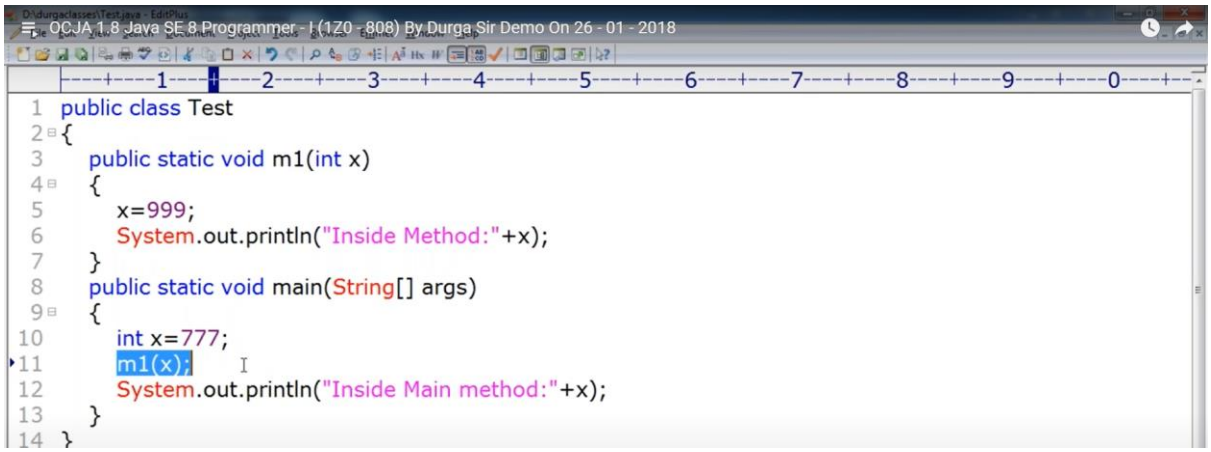


```
public class Test
{
   char c;
   boolean b;
   float f;
   public void print()
   {
      System.out.println("c = "+c);
      System.out.println("b = "+b);
      System.out.println("f = "+f);
   }
   public static void main(String[] args)
   {
      Test t = new Test();
      t.print();
   }
}
```

default value of char is \u0000 its like Square like 0 with a diagonal slash not blank . check ide

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

D:\durgaclasses>java Test
c =
b = false
f = 0.0
```

999 and 777




```
public class Test
{
   public static void m1(int x)
   {
      x=999;
      System.out.println("Inside Method:"+x);
   }
   public static void main(String[] args)
   {
      int x=777;
      m1(x);
      System.out.println("Inside Main method:"+x);
   }
}
```

999 999

If we are passing primitive variable reflection would not happen whereas if we pass object ref reflection would happen