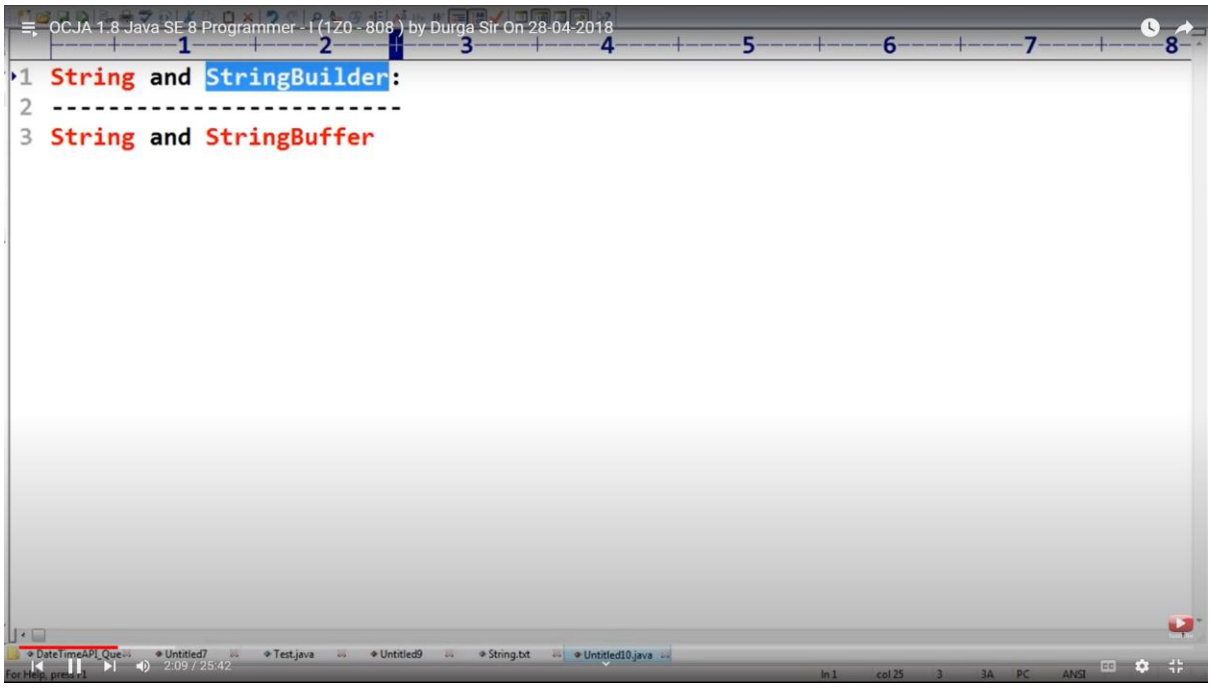


String builder is always non-Synchronized

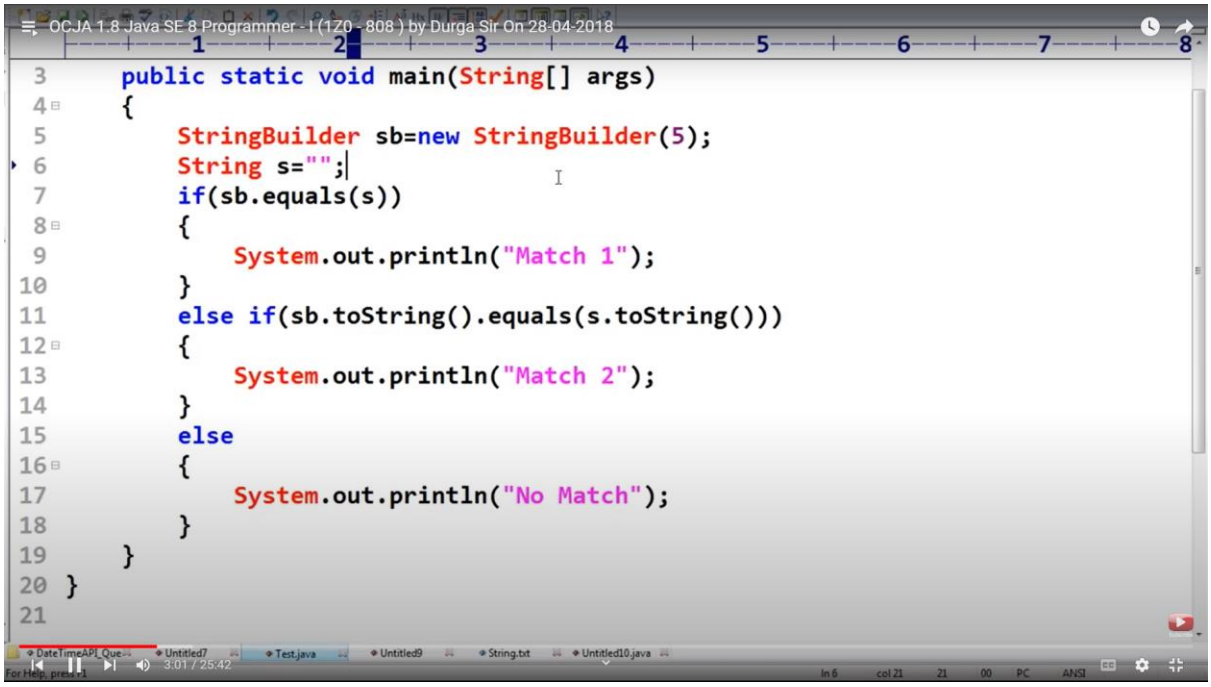
StringBuffer Synchronized

Sb.equals(s). String Builder and buffer is not overridden and calls object class equals method. Ref check since the objects are different its false

Line11- Sb at line 5 is empty . toString will create a empty string. Sb.toString will return String.

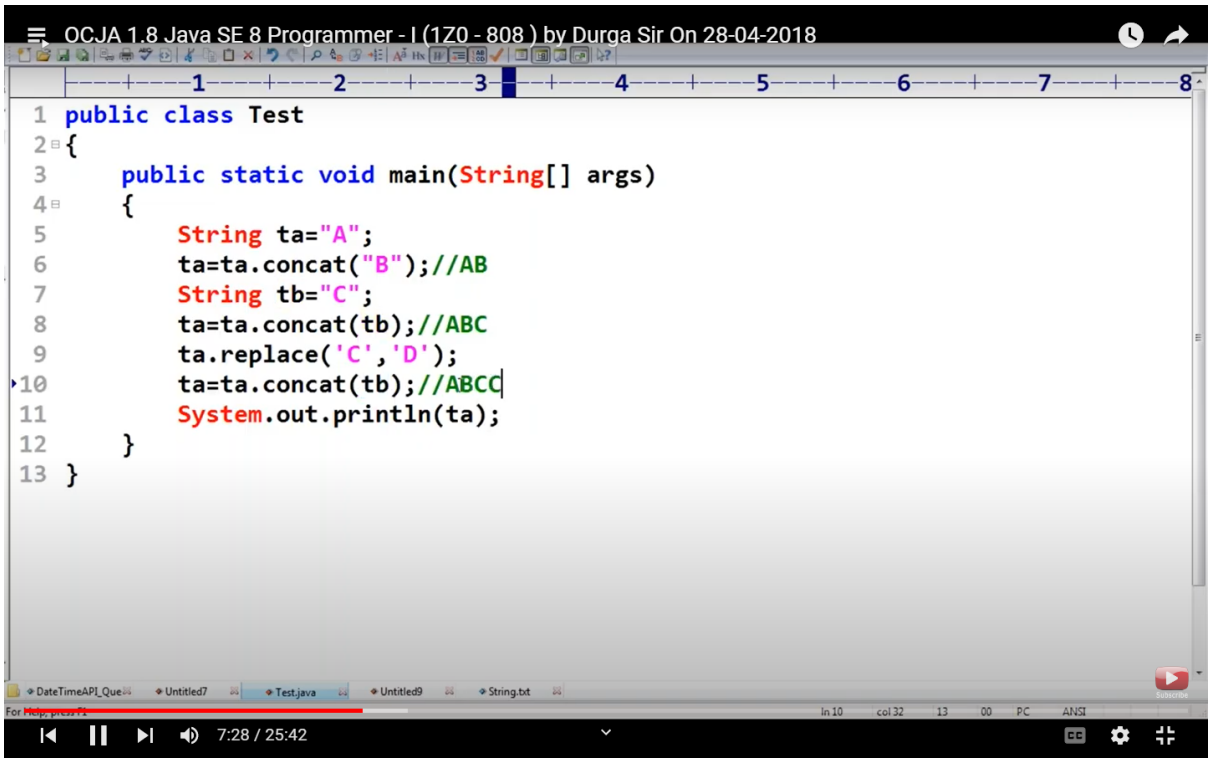
EmptyString.equals(s.toString) -> will call equals on String class which is overridden to perform content comparison. Empty String.equals(emptyString) -> returns true

Match 2 is the answer

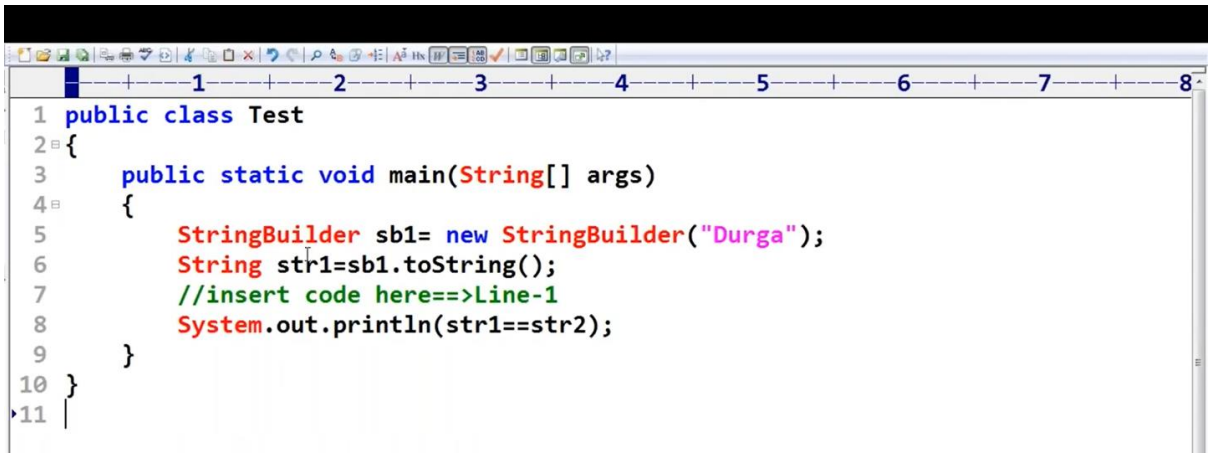


ABCC

Line 9 replace is not assigned to ta-> its just create a new object and available for GC



== always meant for ref comparison so str1 and str2 should point to same object



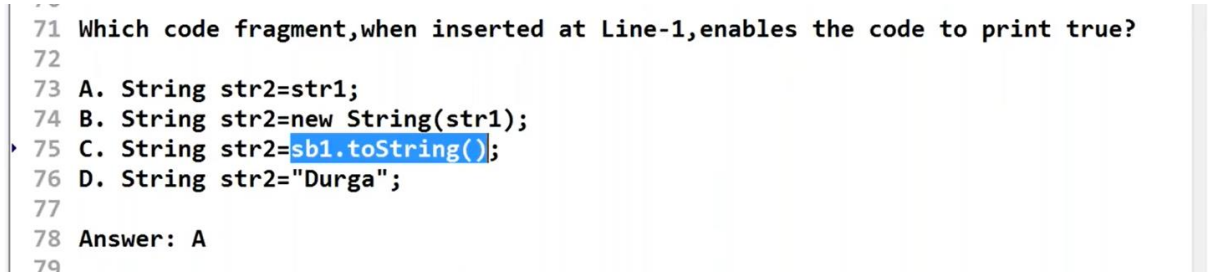
```
1  public class Test
2  {
3      public static void main(String[] args)
4      {
5          StringBuilder sb1= new StringBuilder("Durga");
6          String str1=sb1.toString();
7          //insert code here==>Line-1
8          System.out.println(str1==str2);
9      }
10 }
11
```

B- creates a new object

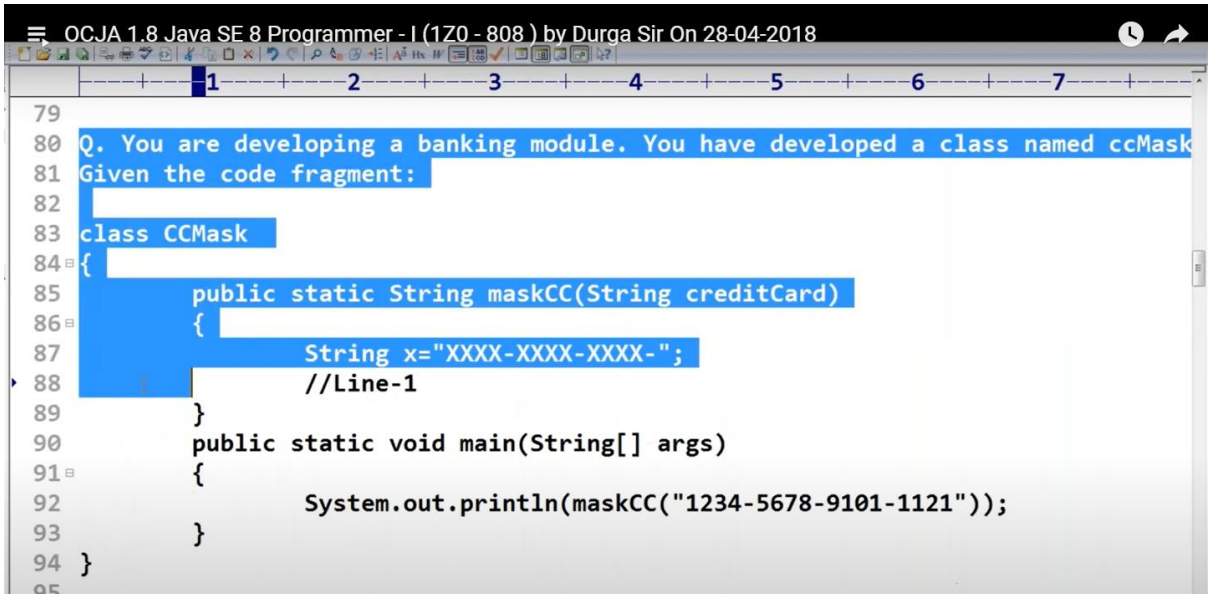
C- when you call a method on sb1 a new object will be created , same as B

D- will create a new object in SCP area. String constant pool

A is the only possible option



```
71 Which code fragment,when inserted at Line-1,enables the code to print true?
72
73 A. String str2=str1;
74 B. String str2=new String(str1);
75 C. String str2=sb1.toString();
76 D. String str2="Durga";
77
78 Answer: A
79
```



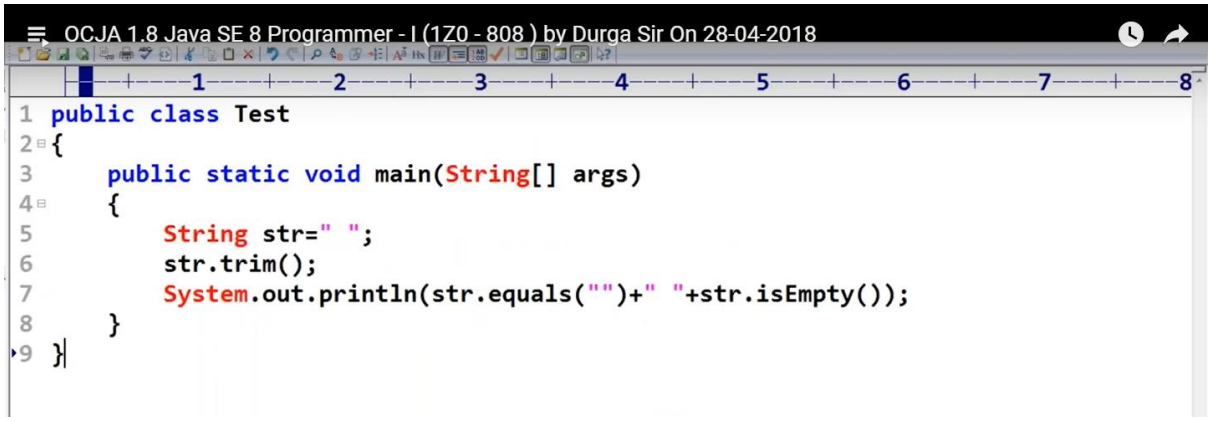
Return the xxxx and last four digits od the credit card string 15-19 position

```
You must ensure that maskCC method returns a String that hides all digits of th

Which two code fragments should you use at line 1, independently to achieve the

A.
StringBuilder sb=new StringBuilder(creditCard);
sb.substring(15,19);
return x+sb;

B.
return x+creditCard.substring(15,19);
```

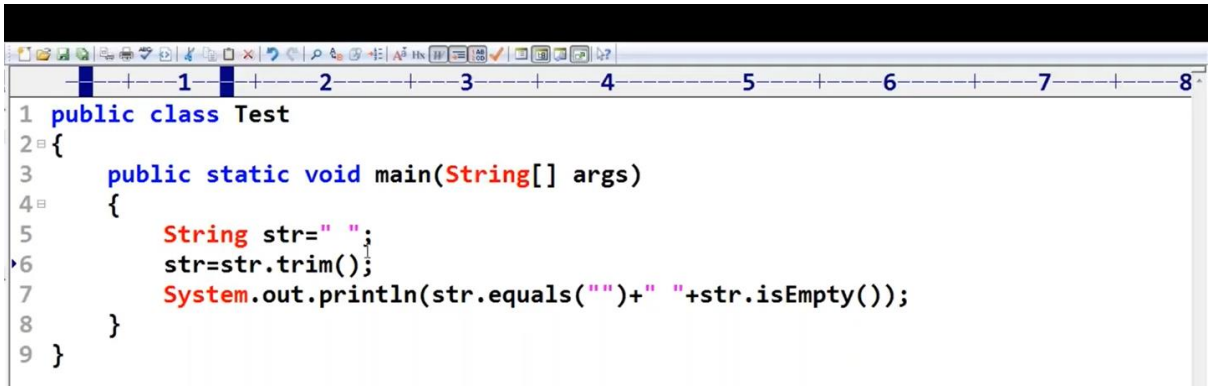
```
C.
StringBuilder sb=new StringBuilder(x);
sb.append(creditCard,15,19);
return sb.toString();

D.
StringBuilder sb=new StringBuilder(creditCard);
StringBuilder s=sb.insert(0,x);
return s.toString();

Answer: B,C
```

A- Line102 is not assigned to SB
B- Is right
C- Is right
D- Insert will shift the sb characters to right and will not produce the right result

6 has not been assigned to str



False False

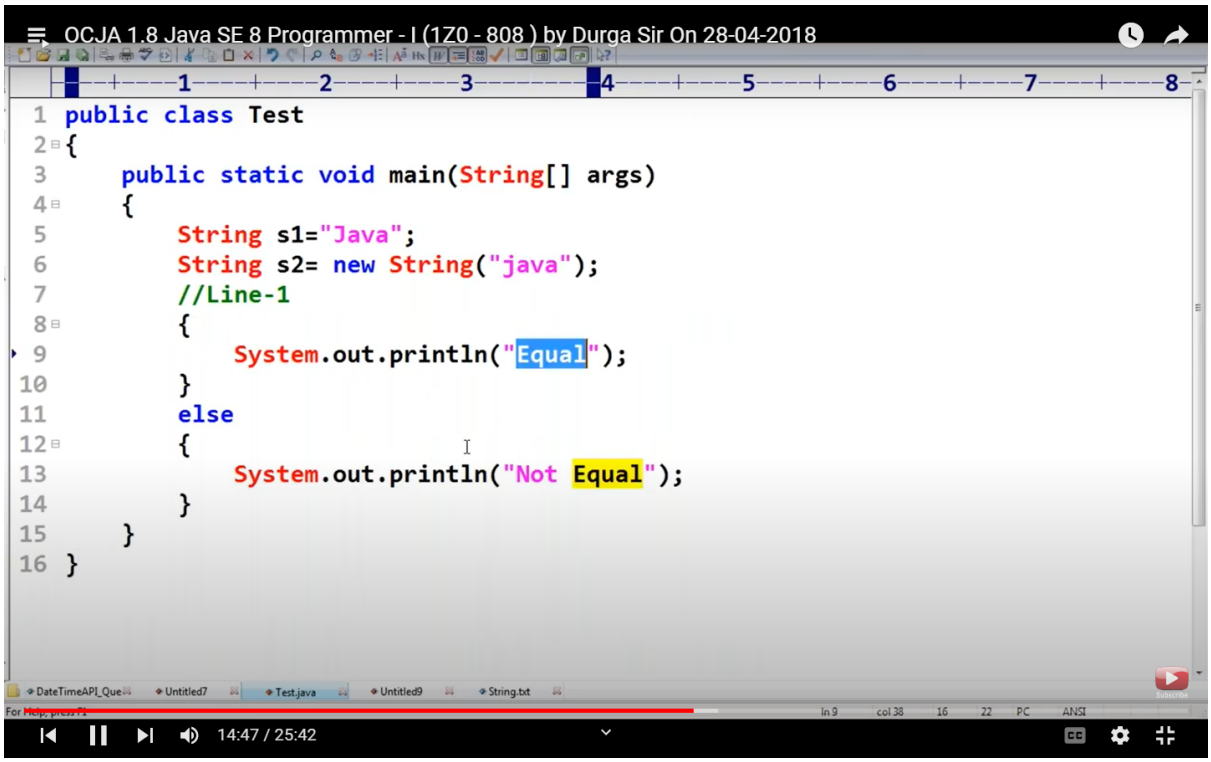


True True

“” and .isEmpty() -> the same

Need to Print Equal, what code should be inserted

If(s1.equalsIgnoreCase(s2)

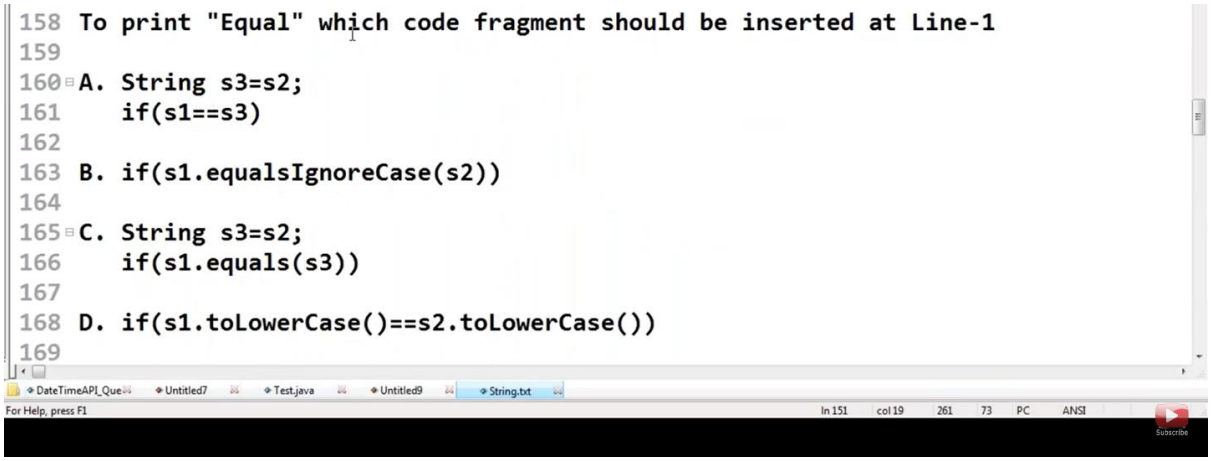


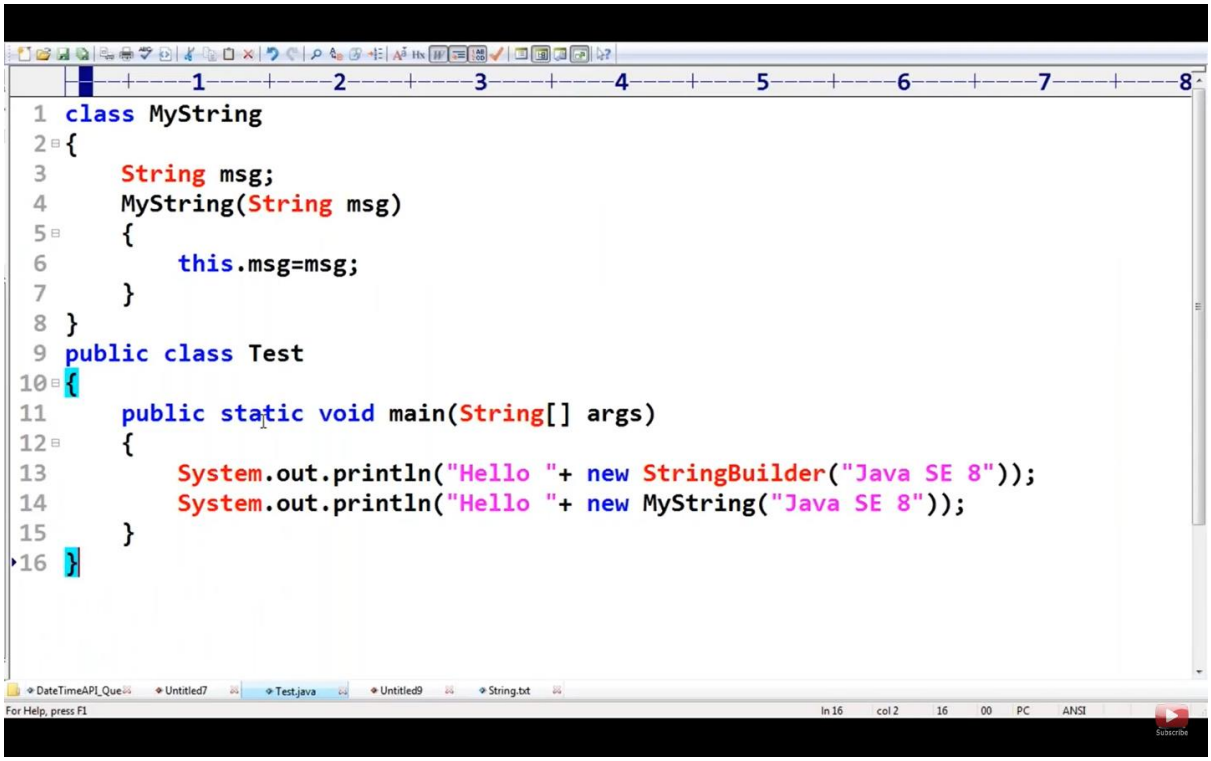
A is wrong - S1 in SCP, S2 is new object, S3 =S2 will poiint to S2 , S1==S3 will be False

B - true

c- is False because case is Java and java

E- Both method will return new objects and ref comparison will return False





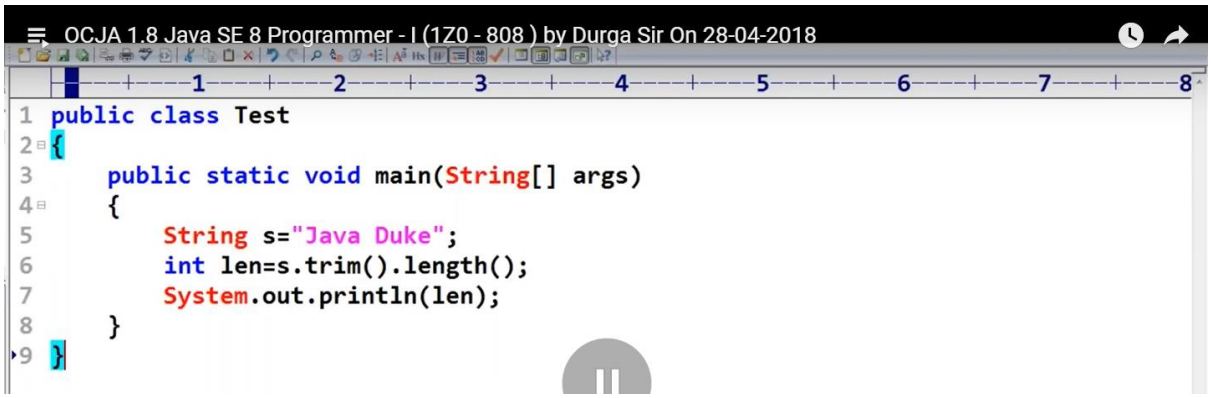
```java
class MyString
{
    String msg;
    MyString(String msg)
    {
        this.msg=msg;
    }
}
public class Test
{
    public static void main(String[] args)
    {
        System.out.println("Hello "+ new StringBuilder("Java SE 8"));
        System.out.println("Hello "+ new MyString("Java SE 8"));
    }
}
```


Line 13-> Hello Java SE 8 -> will call toString method on new StringBuilder Object

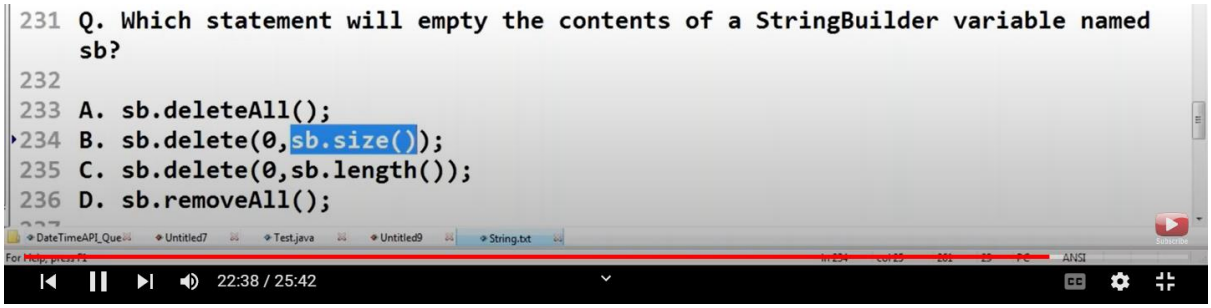
Line 14-> Hello MyString@<HashCode> -> this wil call the toString method on Object class which is class name @ hashcode -> To String is not overriden

Hello Java SE 8

Hello MyString@<HashCode>


OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) by Durga Sir On 28-04-2018
1 public class Test
2 {
3 public static void main(String[] args)
4 {
5 String s="Java Duke";
6 int len=s.trim().length();
7 System.out.println(len);
8 }
9 }

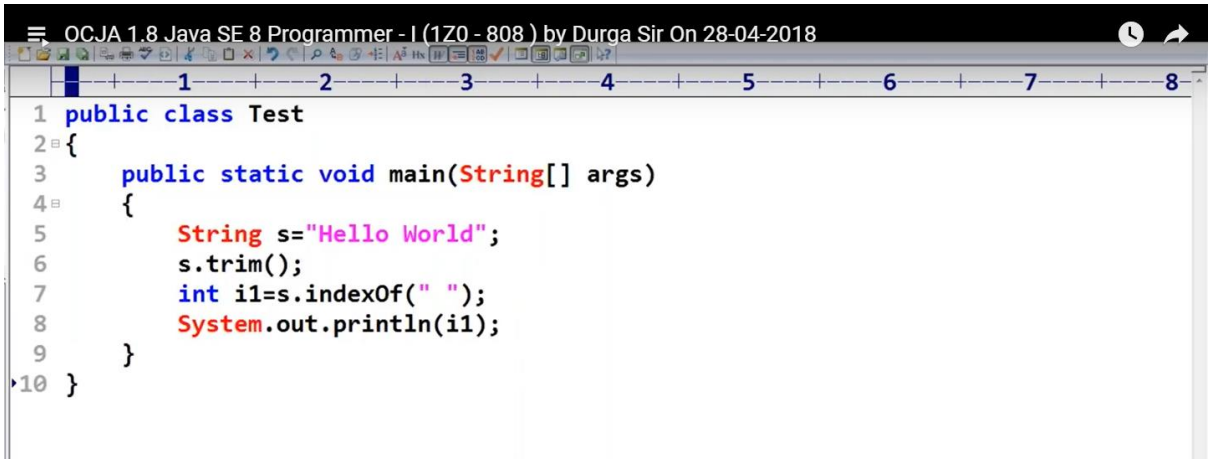
OutPut -9



Sb.delete(0,Length()),

Delete all contents from 0 to length

Rest of the options are available for collections not StringBuilder


OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808) by Durga Sir On 28-04-2018
1 2 3 4 5 6 7 8
1 public class Test
2 {
3 public static void main(String[] args)
4 {
5 String s="Hello World";
6 s.trim();
7 int i1=s.indexOf(" ");
8 System.out.println(i1);
9 }
10 }

Output:

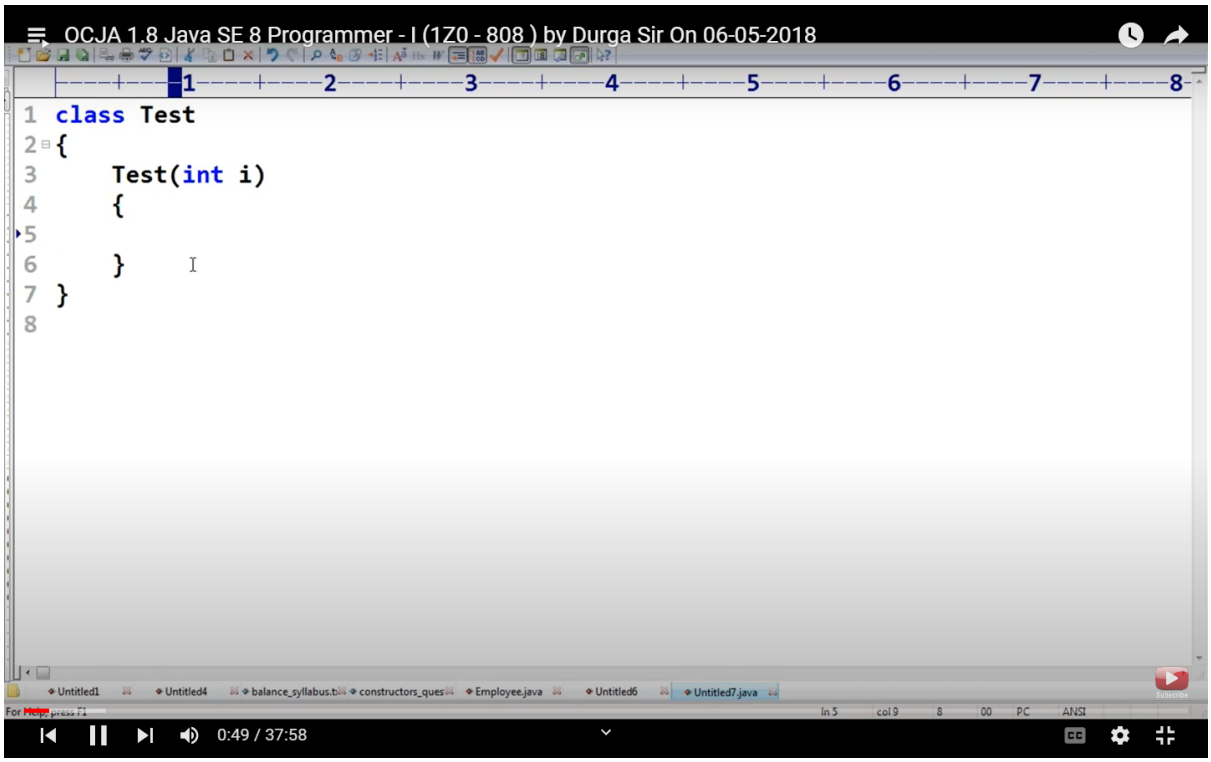
5

Constructor

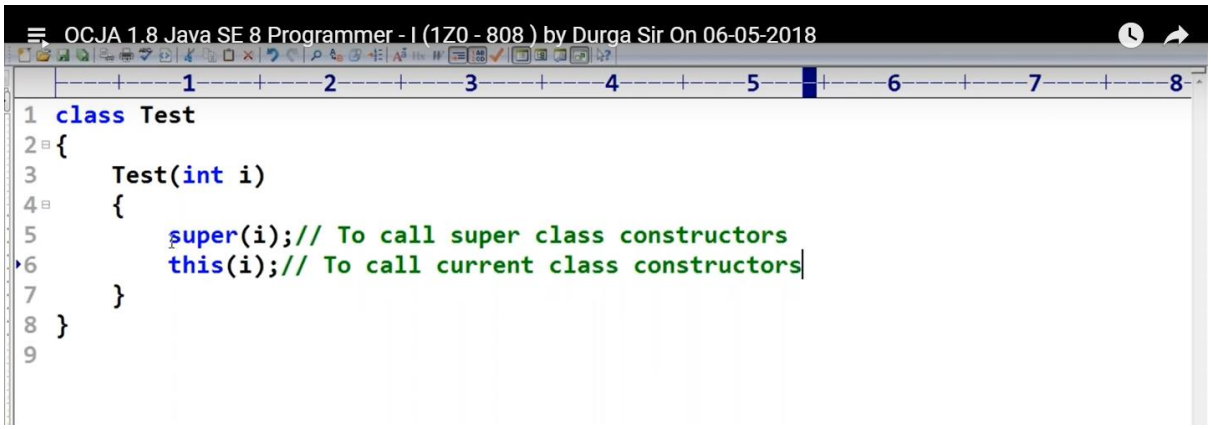
First Line compiler will always add super(); if u are not writing

To call super class contructors. If u r not writing super will be place by compiler.

U can write this() to call current class contructor

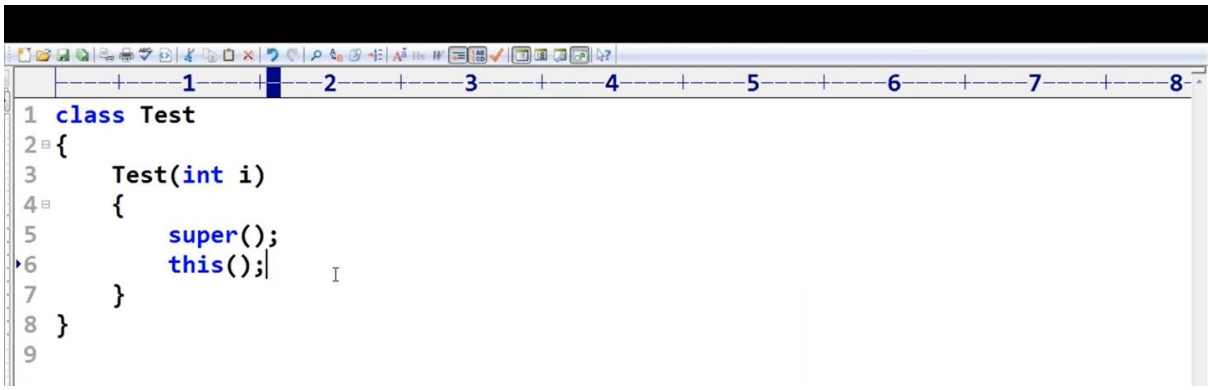


U can use either super() or this() but not both simultanously , else CE



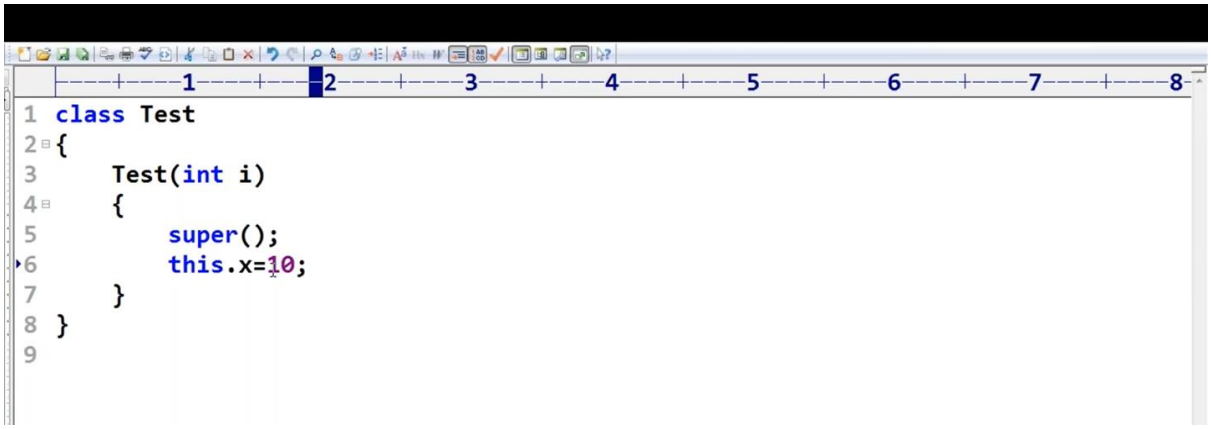
First line should be super or this, if you add it in second line or upcoming lines CE

Invalid



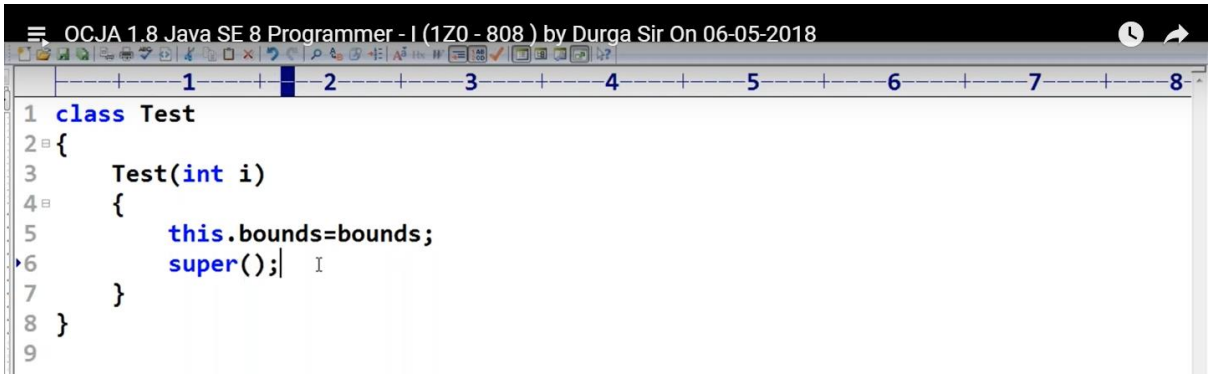
```
1 class Test
2 {
3     Test(int i)
4     {
5         super();
6         this();
7     }
8 }
9
```

The below is valid , line 6 is not constructor call, but variable assignment

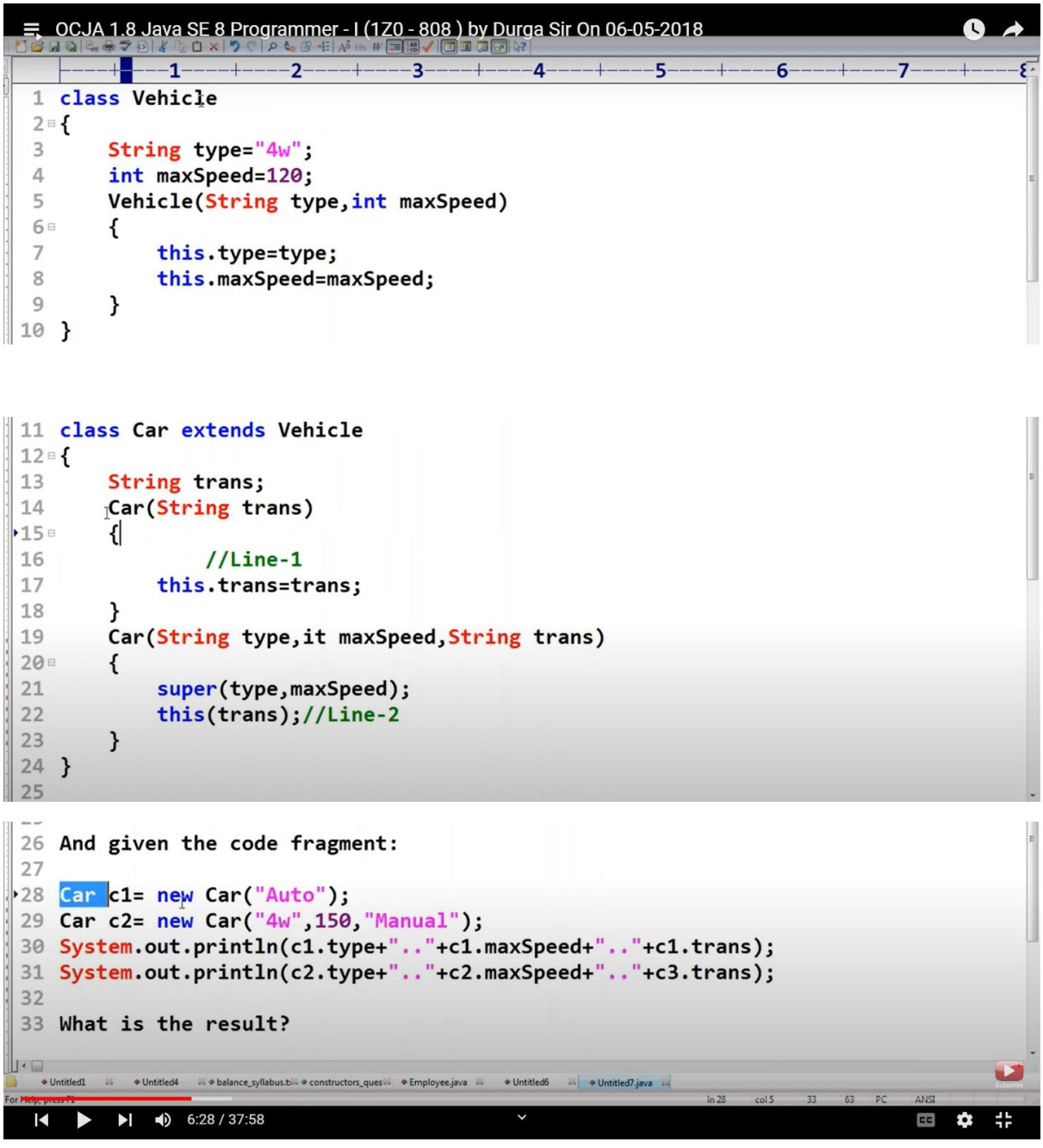


```
1 class Test
2 {
3     Test(int i)
4     {
5         super();
6         this.x=10;
7     }
8 }
9
```

This is invalid, super is in second line




```
1 class Test
2 {
3     Test(int i)
4     {
5         this.bounds=bounds;
6         super();
7     }
8 }
9
```



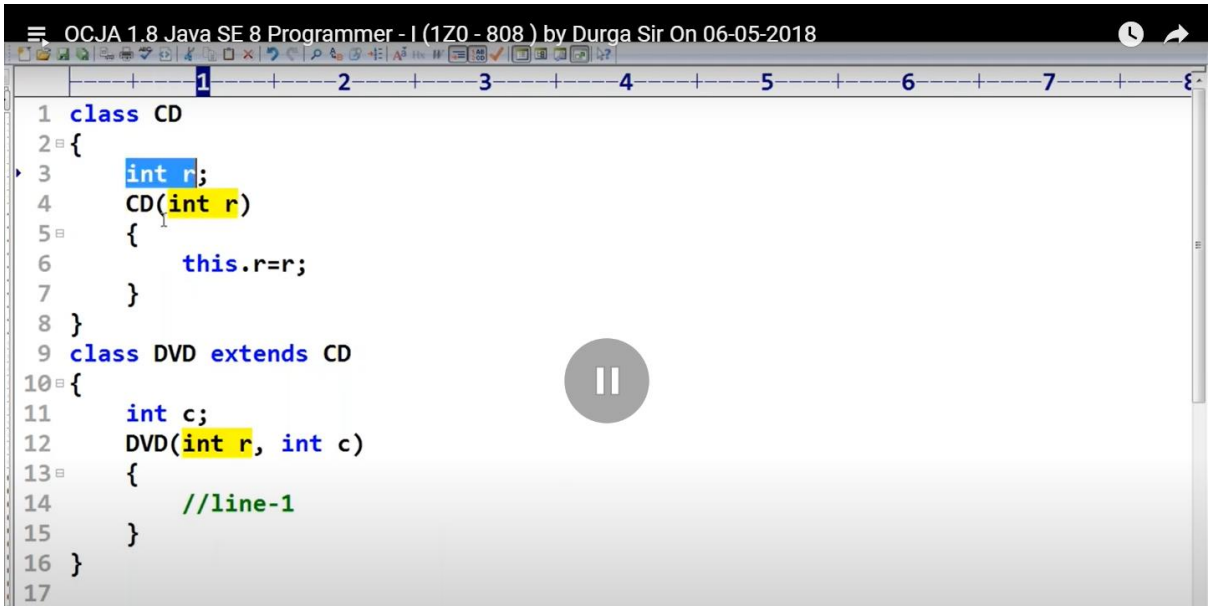
Very Important

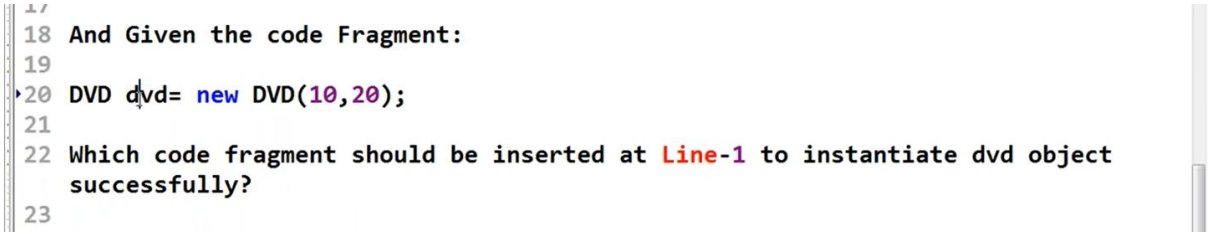
Line 16 – Compile is going to place Super() and in super class , no argument constructor is not available. Hence CE

Line 22. Super and This are used simultaneously . CE

Both Line 1 and Line2 -> CE

CE – Line2



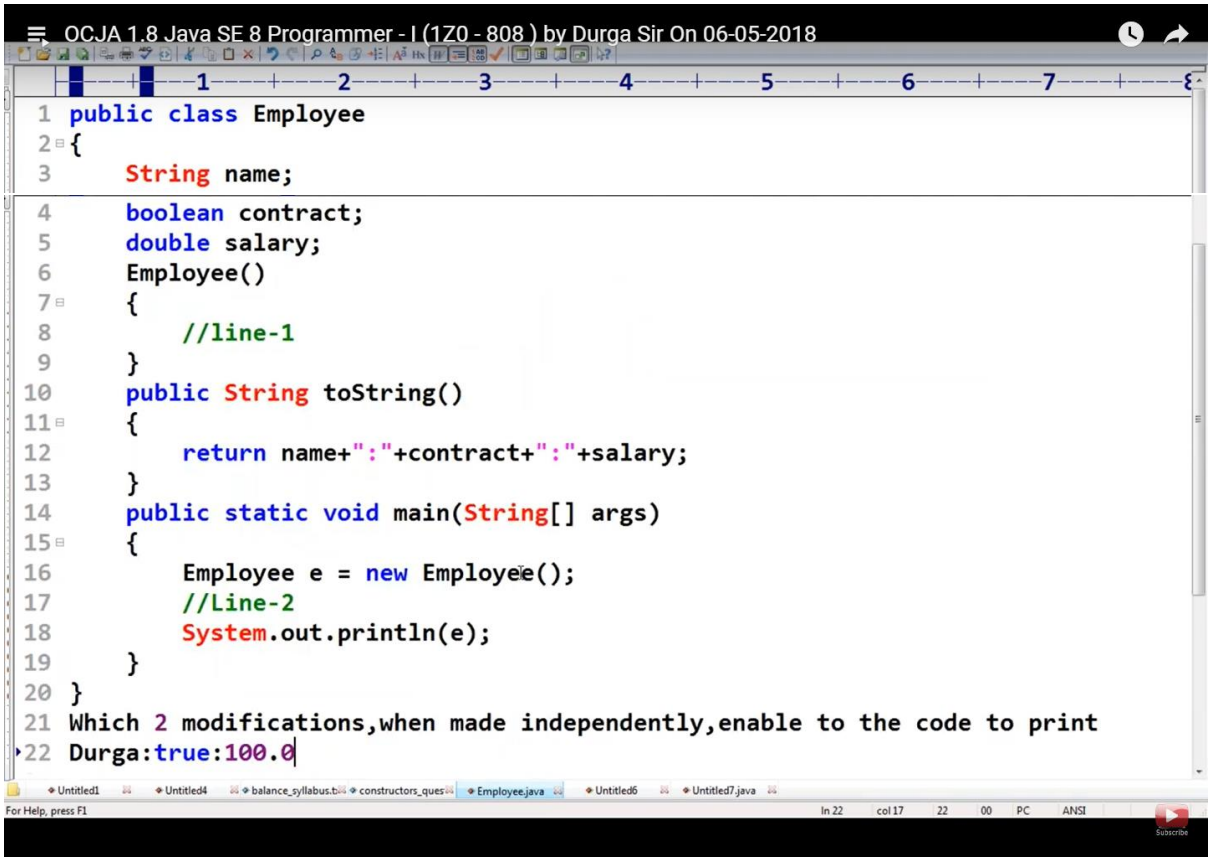


B – CE

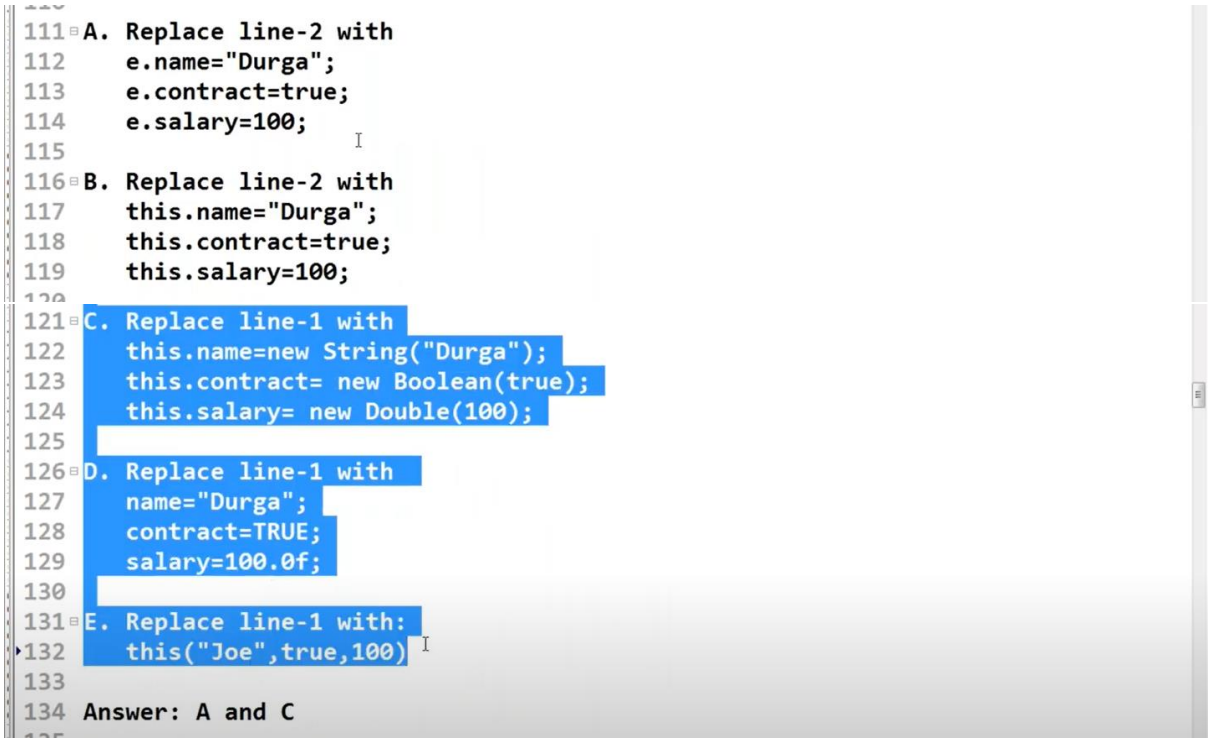
A compiler will place super at line 1 and has no no argument const. Super.r = r is a right statement check once

C- right

D- Super used in Second line Wrong

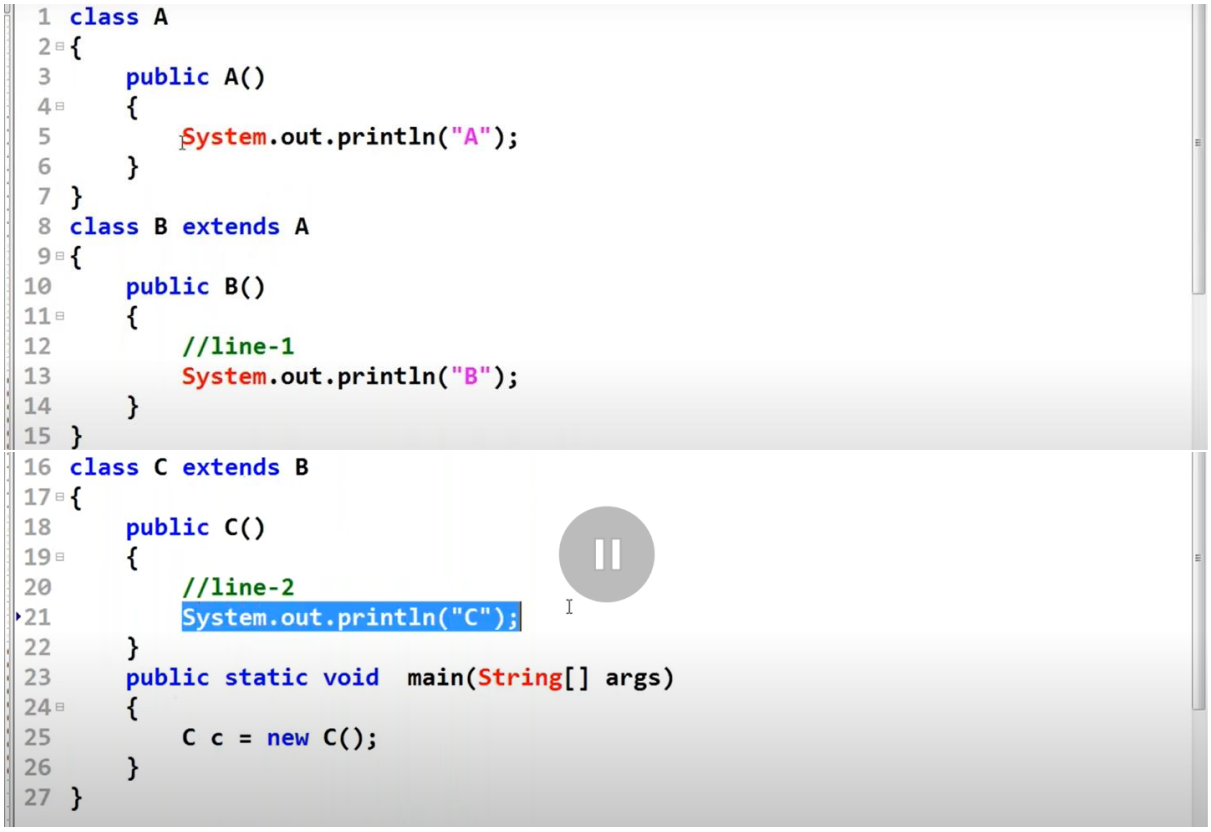


Independently means Line 1 independently should work and Line 2 separately should work . Not both



A- After creating an object you can use e.salary e.name after creating an object.

D- CE. E There is no constructor with 3 args



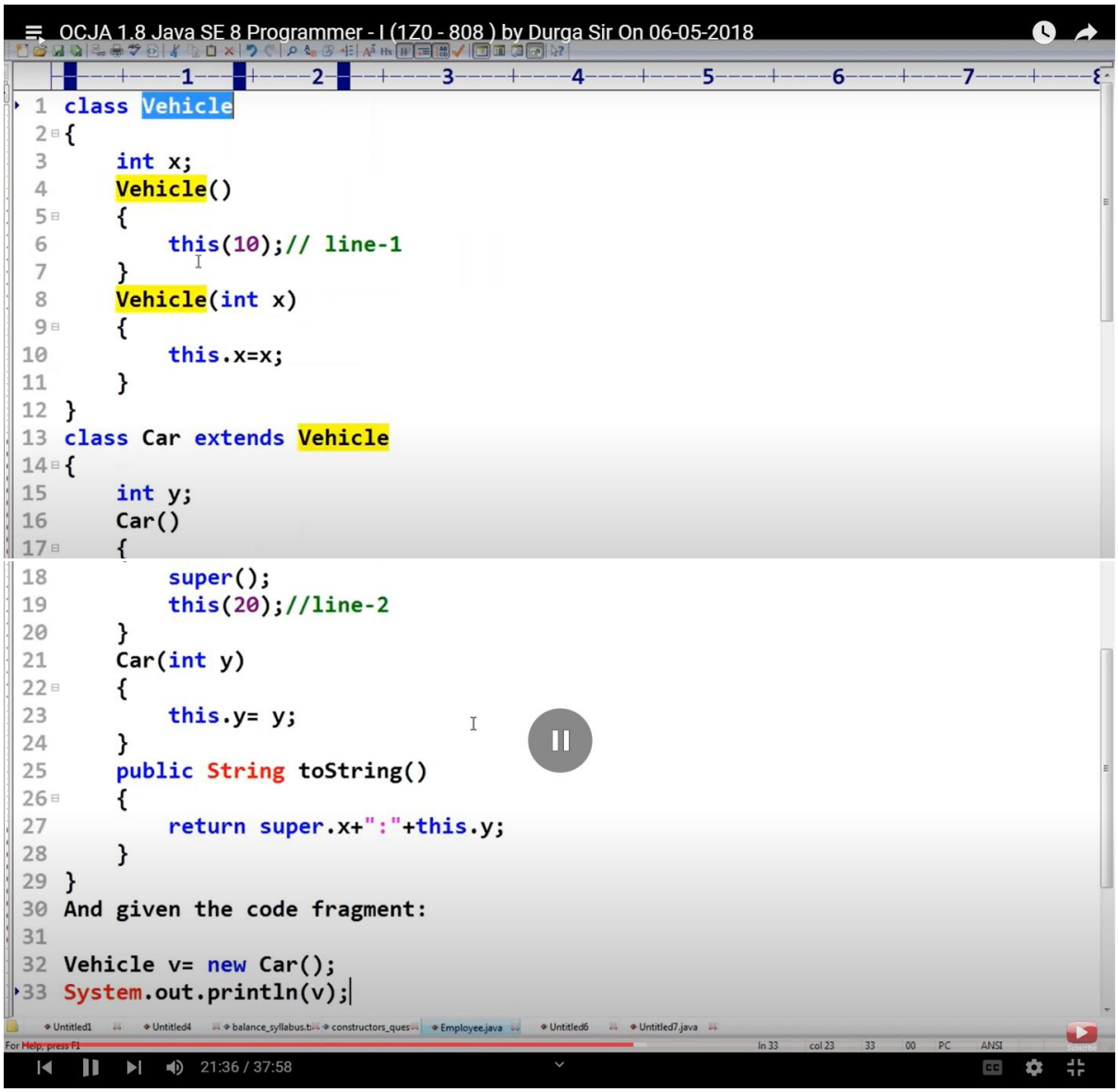
```java
class A
{
    public A()
    {
        System.out.println("A");
    }
}
class B extends A
{
    public B()
    {
        //line-1
        System.out.println("B");
    }
}
class C extends B
{
    public C()
    {
        //line-2
        System.out.println("C");
    }
    public static void  main(String[] args)
    {
        C c = new C();
    }
}
```

When ever we create child class object parent class object will be executed. First top most parent const will execute. Super() method will be placed in the first line of all constructors

A

B

C



Line 18 0-> both super and this has been used

Compilation fails at line 2

```java
public class Person
{
    String name;
    int age=25;
    public Person(String name)
    {
        this(); //line-1
        setName(name);
    }
    public Person(String name,int age)
    {
        Person(name);//Line-2
        setAge(age);
    }
    //setter and getter methods go here
    public String show()
    {
        return name+" "+age+" "+number;
    }
    public static void main(String[] args)
    {
        Person p1= new Person("Durga");
        Person p2= new Person("Ravi",50);
        System.out.println(p1.show());
        System.out.println(p2.show());
    }
}
```

Line 7- Class person does not have a no argument constructor

Line 12 – Person(name) – u cant call a constructor with the class name. u should use only

This() .

Line 1 and Line2 CE

```
1  class Animal
2  {
3      String type="Canine";
4      int maxSpeed=60;
5      Animal(){}
6      Animal(String type,int maxSpeed)
7      {
8          this.type=type;
9          this.maxSpeed=maxSpeed;
10     }
11 }
12 class WildAnimal extends Animal
13 {
14     String bounds;
15     WildAnimal(String bounds)
16     {
17         //line-1
18     }
19     WildAnimal(String type,int maxSpeed)
20     {
21         //line-2
22     }
23 }
24 And the code fragment:
25
26 WildAnimal wolf= new WildAnimal("Long");
27 WildAnimal tiger= new WildAnimal("Feline",,80,"Short");
28 System.out.println(wolf.type+" "+wolf.maxSpeed+" "+wolf.bounds);
29 System.out.println(tiger.type+" "+tiger.maxSpeed+" "+tiger.bounds);
30
31 Which 2 modifications enable to the code to print the following output?
33 Casine 60 Long
34 Feline 80 Short
```

Line1-> this.bounds = bounds;

Line 19-> WildAnimal(String type, int maxSpeed, String bounds)

Line 2->

super(type, maxSpeed);

this.bounds = bounds;

A Right

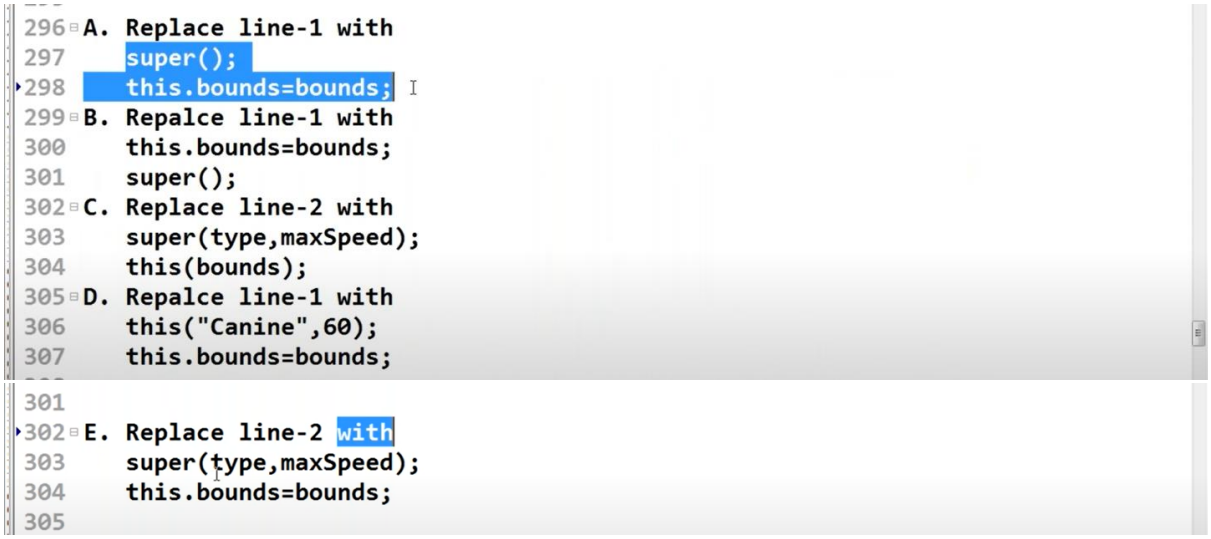
B. Super ins econd line

C. Super and This simulataneously

D. there are no 2 arg const in this – wrong

E- Right

A and E change req



This(name age) – Line 16 – CE this comes in second line

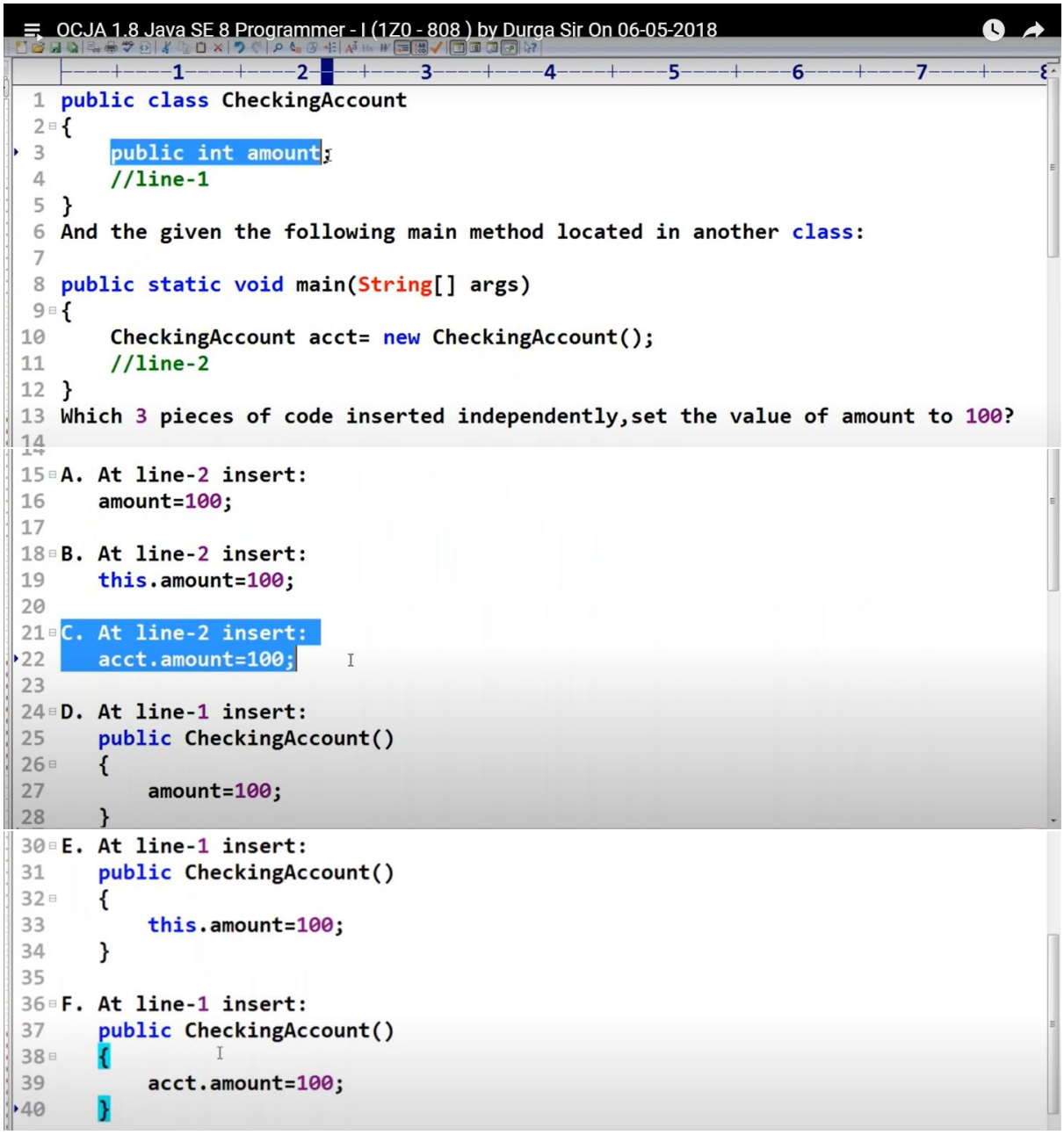
```
1  class Employee
2  {
3      private String name;
4      private int age;
5      private int salary;
6
7      public Employee(String name,int age)
8      {
9          setName(name);
10         setAge(age);
11         setSalary(2000);
12     }
13     public Employee(String name,int age,int salary)
14     {
15         setSalary(salary);
16         this(name,age);
17     }
18     //getter and setter methods goes here
19     public void printDetails()
20     {
21         System.out.println(name+":"+age+":"+salary);
22     }
23 }
24 Test.java:
25
26 class Test
27 {
28     public static void main(String[] args)
29     {
30         Employee e1= new Employee();
31         Employee e2= new Employee("Durga",50);
32         Employee e3= new Employee("Ravi",40,5000);
33         e1.printDetails();
34         e2.printDetails();
35         e3.printDetails();
36     }
37 }
```

Line 30 – No args const not available. Line 16 This is in second line CE

Ans E – CE in both the class

```
354 A. Compilation fails in the Employee class
355 B. null:0:0
356    Durga:50:0
357    Ravi:40:5000
358 C. null:0:0
359    Durga:50:2000
360    Ravi:40:5000
361 D. Compilation Fails in the Test class
362 E. Compilation Fails in both Test and Employee classes
363
```

\



A is wrong . Can access public instance variable like acct.amount

B is wrong . Cant access This.amount

C. is right

D. is right you can access as amount or this.amount in the const

E. is right

F. is wrong

C,D and Es