

String builder is always non-Synchronized

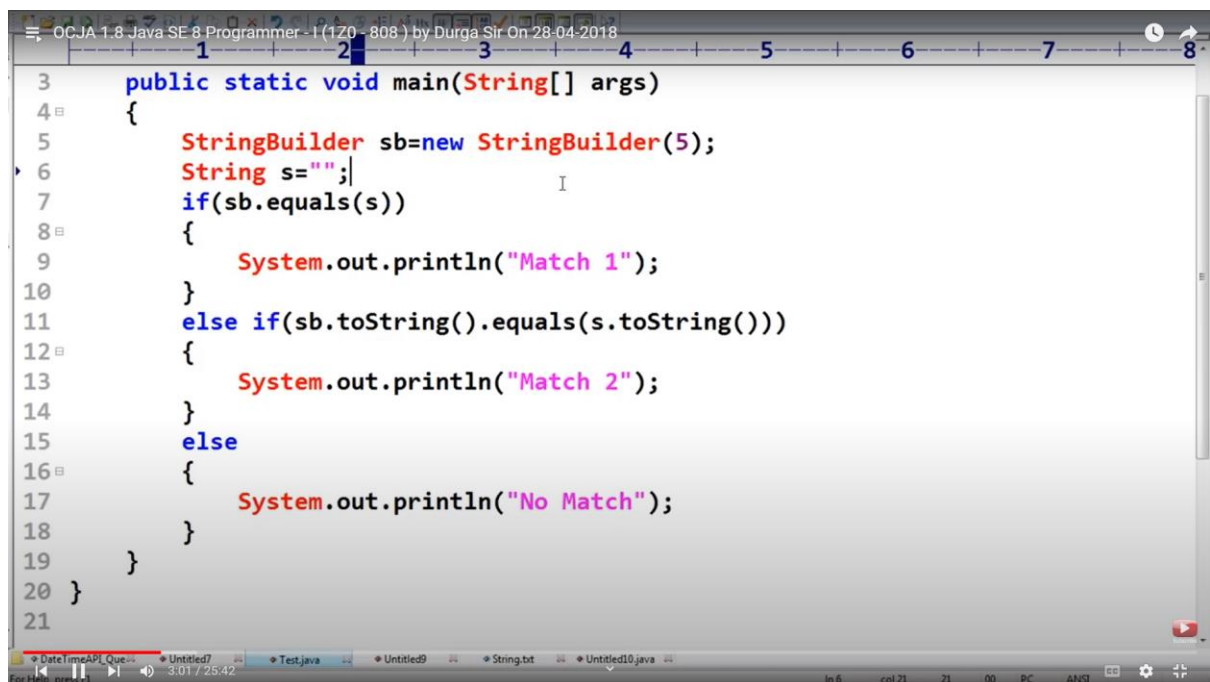
StringBuffer Synchronized

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

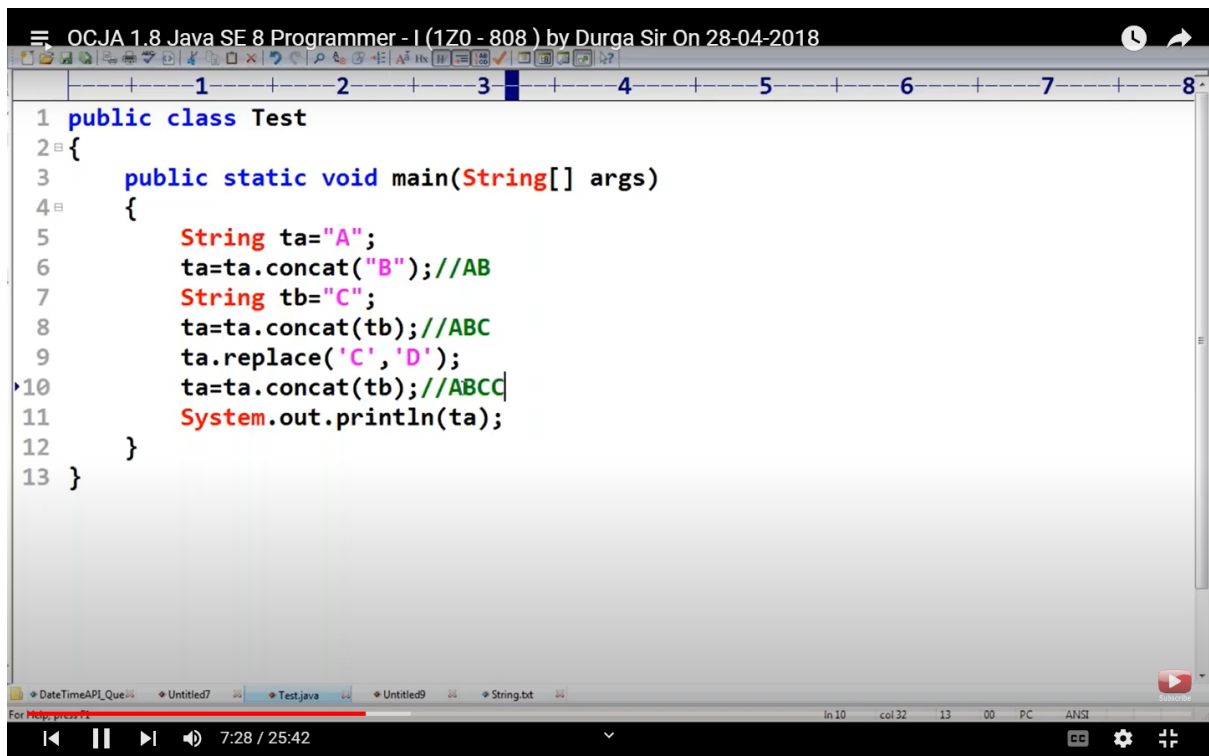
Match 2 is the answer



```
3 public static void main(String[] args)
4 {
5     StringBuilder sb=new StringBuilder(5);
6     String s="";
7     if(sb.equals(s))
8     {
9         System.out.println("Match 1");
10    }
11    else if(sb.toString().equals(s.toString()))
12    {
13        System.out.println("Match 2");
14    }
15    else
16    {
17        System.out.println("No Match");
18    }
19 }
20 }
21
```

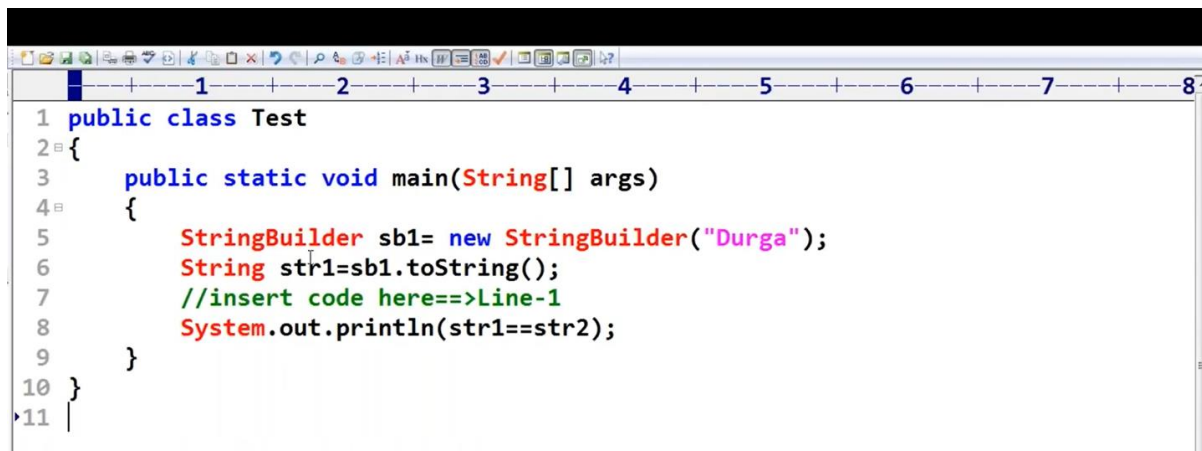
ABCC

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



```
1 public class Test
2 {
3     public static void main(String[] args)
4     {
5         String ta="A";
6         ta=ta.concat("B");//AB
7         String tb="C";
8         ta=ta.concat(tb);//ABC
9         ta.replace('C', 'D');
10        ta=ta.concat(tb);//ABCC
11        System.out.println(ta);
12    }
13 }
```

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



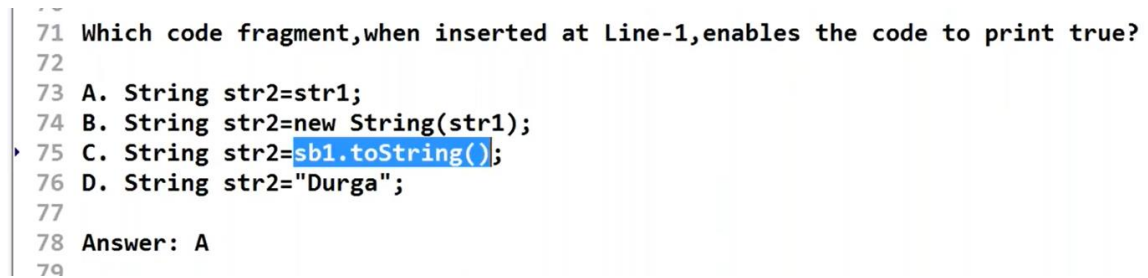
```
1 public class Test
2 {
3     public static void main(String[] args)
4     {
5         StringBuilder sb1= new StringBuilder("Durga");
6         String str1=sb1.toString();
7         //insert code here==>Line-1
8         System.out.println(str1==str2);
9     }
10 }
11
```

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
```

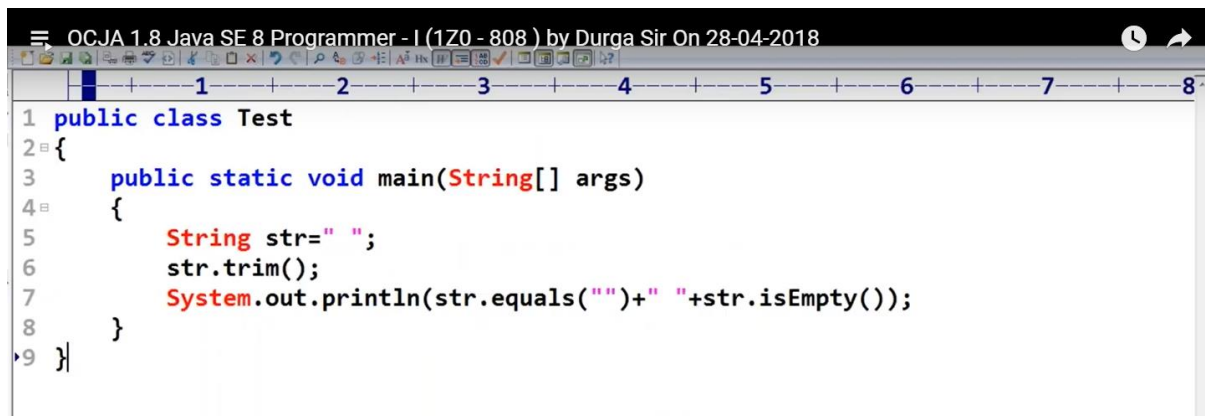
```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808 ) by Durga Sir On 28-04-2018
79
80 Q. You are developing a banking module. You have developed a class named ccMask
81 Given the code fragment:
82
83 class CCMask
84 {
85     public static String maskCC(String creditCard)
86     {
87         String x="XXXX-XXXX-XXXX-";
88         //Line-1
89     }
90     public static void main(String[] args)
91     {
92         System.out.println(maskCC("1234-5678-9101-1121"));
93     }
94 }
```

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

```
96 You must ensure that maskCC method returns a String that hides all digits of the
97 Which two code fragments should you use at line 1, independently to achieve the
98
99
100 A.
101 StringBuilder sb=new StringBuilder(creditCard);
102 sb.substring(15,19);
103 return x+sb;
104
105 B.
106 return x+creditCard.substring(15,19);
107
108 C.
109 StringBuilder sb=new StringBuilder(x);
110 sb.append(creditCard,15,19);
111 return sb.toString();
112
113 D.
114 StringBuilder sb=new StringBuilder(creditCard);
115 StringBuilder s=sb.insert(0,x);
116 return s.toString();
117
118 Answer: B,C
119
```

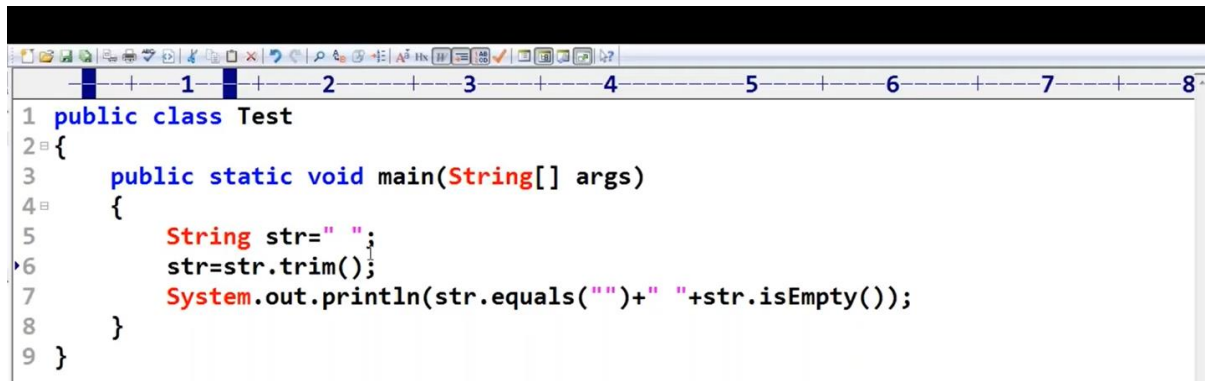
- A- Line102 is not assigned to SB
- B- Is right
- C- Is right
- D- Insert will shift the sb characters to right and will not produce the right result

6 has not been assigned to str



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

False False



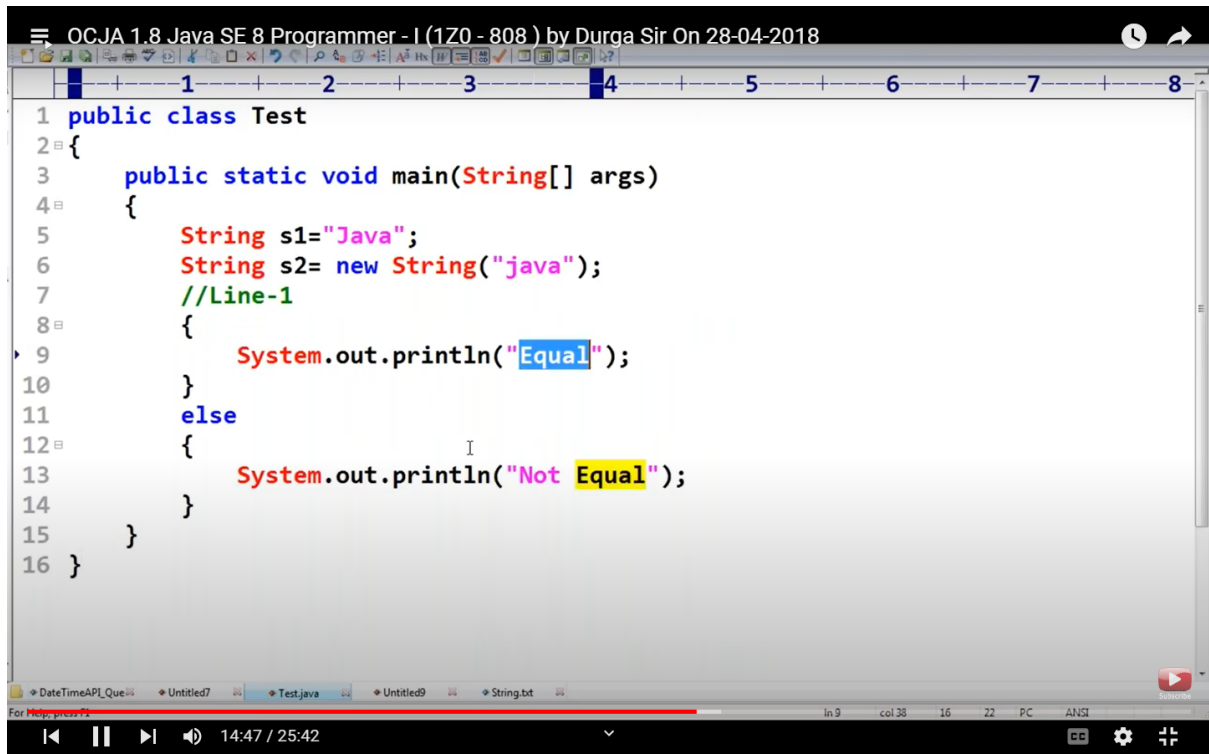
```
1 public class Test
2 {
3     public static void main(String[] args)
4     {
5         String str= " ";
6         str=str.trim();
7         System.out.println(str.equals("")+" "+str.isEmpty());
8     }
9 }
```

True True

"" and .isEmpty() -> the same

Need to Print Equal, what code should be inserted

If(s1.equalsIgnoreCase(s2))



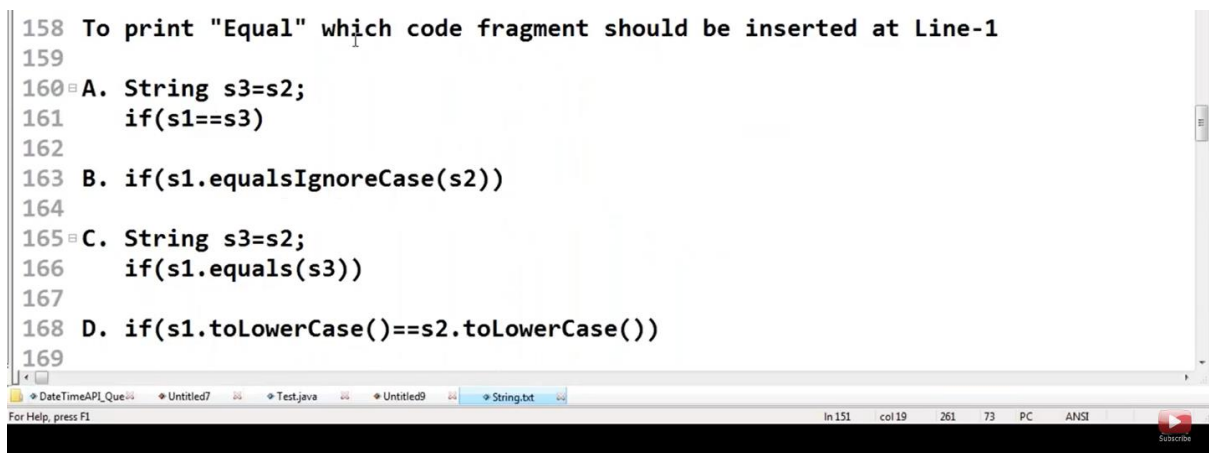
```
1 public class Test
2 {
3     public static void main(String[] args)
4     {
5         String s1="Java";
6         String s2= new String("java");
7         //Line-1
8         {
9             System.out.println("Equal");
10        }
11        else
12        {
13            System.out.println("Not Equal");
14        }
15    }
16 }
```

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

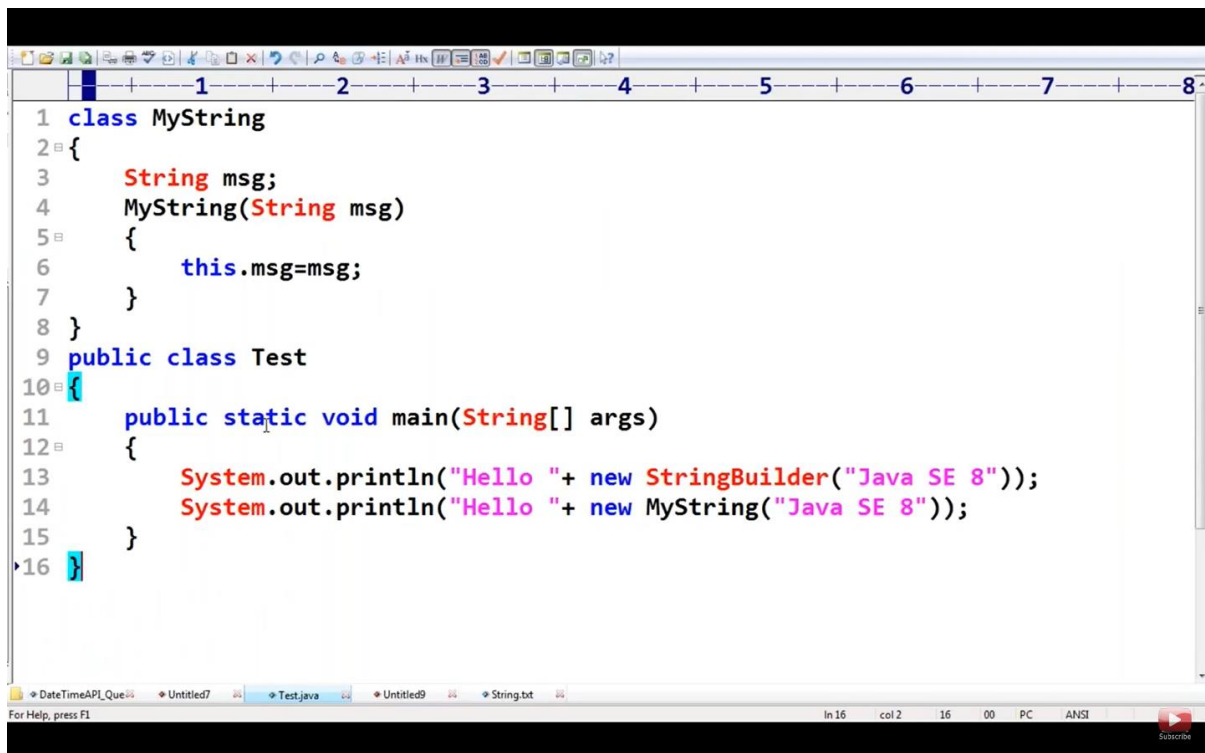
B - true

c- is False because case is Java and java

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



```
158 To print "Equal" which code fragment should be inserted at Line-1
159
160 A. String s3=s2;
161     if(s1==s3)
162
163 B. if(s1.equalsIgnoreCase(s2))
164
165 C. String s3=s2;
166     if(s1.equals(s3))
167
168 D. if(s1.toLowerCase()==s2.toLowerCase())
169
```

A screenshot of an IDE window showing a Java file named 'Test.java'. The code defines a 'MyString' class with a 'String msg' attribute and a constructor that sets 'this.msg = msg'. It also defines a 'Test' class with a 'main' method that prints 'Hello ' followed by the 'toString()' output of a new 'StringBuilder' object and a new 'MyString' object, both containing the string 'Java SE 8'. The IDE interface includes a toolbar at the top, a line number margin on the left, and a status bar at the bottom showing 'Ln 16, col 2, 16, 00, PC, ANSI'.

```
1 class MyString
2 {
3     String msg;
4     MyString(String msg)
5     {
6         this.msg=msg;
7     }
8 }
9 public class Test
10 {
11     public static void main(String[] args)
12     {
13         System.out.println("Hello " + new StringBuilder("Java SE 8"));
14         System.out.println("Hello " + new MyString("Java SE 8"));
15     }
16 }
```

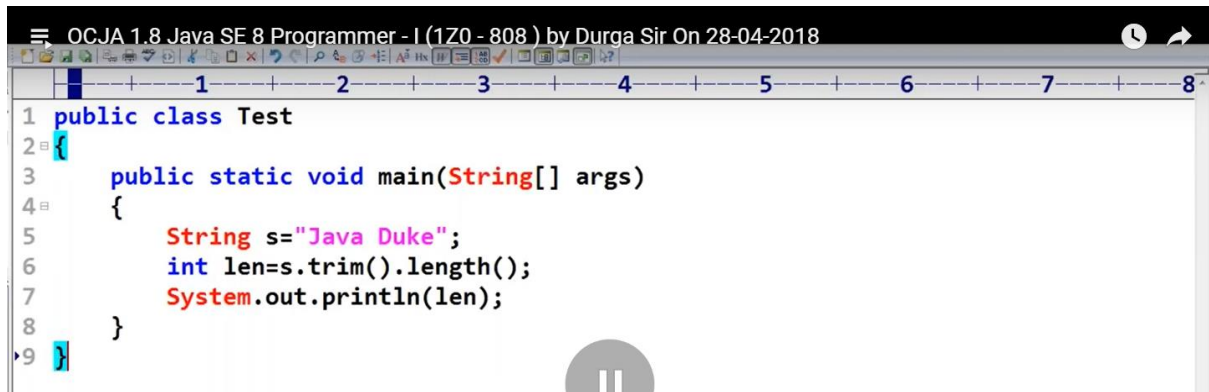
Line 13-> Hello Java SE 8 -> will call toString method on new StringBuilder Object

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>





```
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 }
```

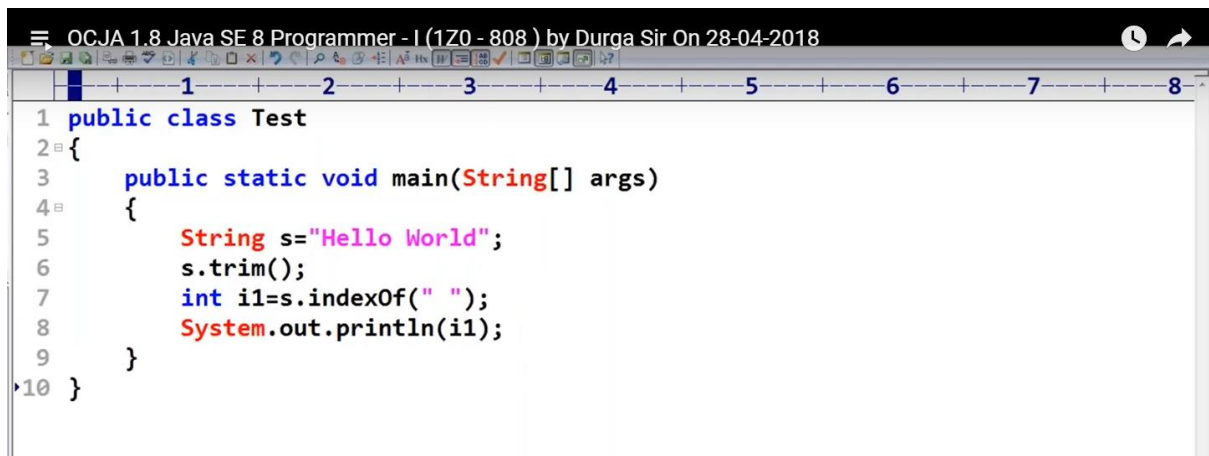
OutPut -9

```
231 Q. Which statement will empty the contents of a StringBuilder variable named
    sb?
232
233 A. sb.deleteAll();
234 B. sb.delete(0, sb.size());
235 C. sb.delete(0, sb.length());
236 D. sb.removeAll();
```

Sb.delete(0,Length()),

Delete all contents from 0 to length

Rest of the options are available for collections not StringBuilder



```
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:

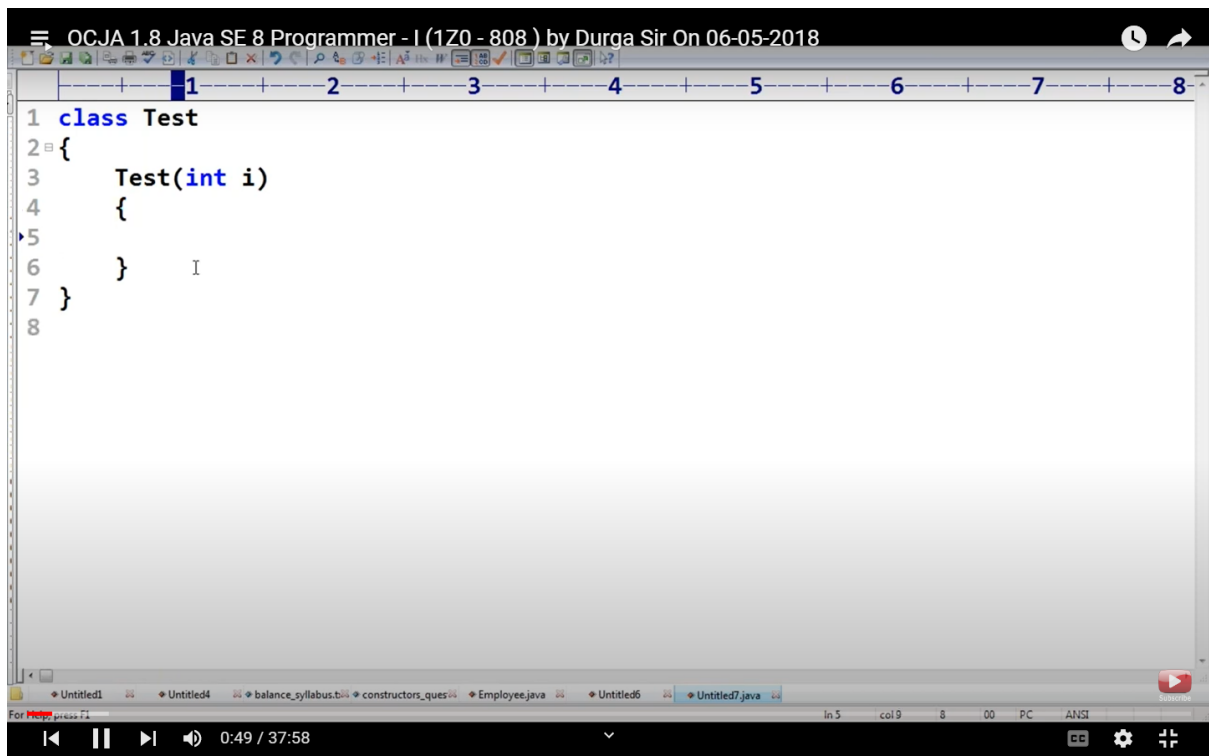
5

## Constructor

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

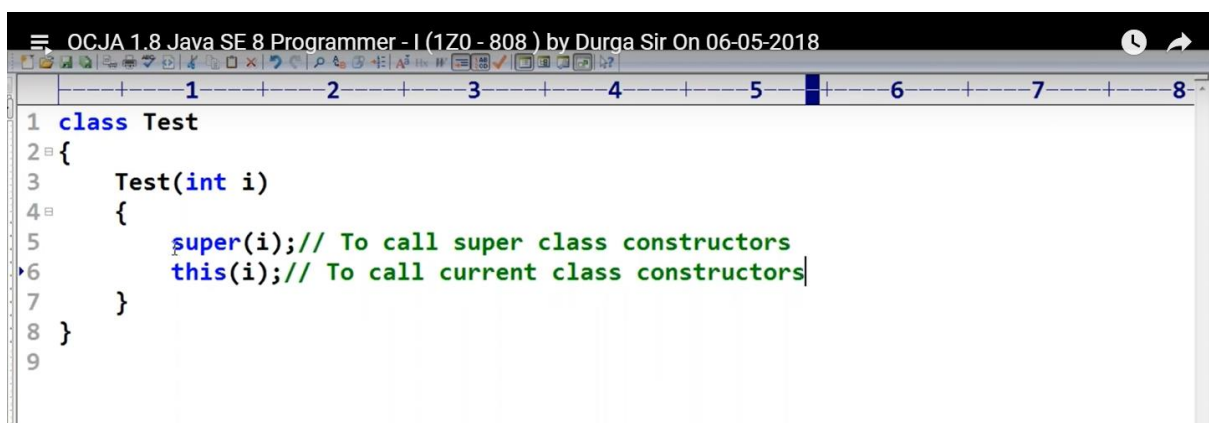
To call super class constructors. If u r not writing super will be place by compiler.

U can write this() to call current class constructor



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

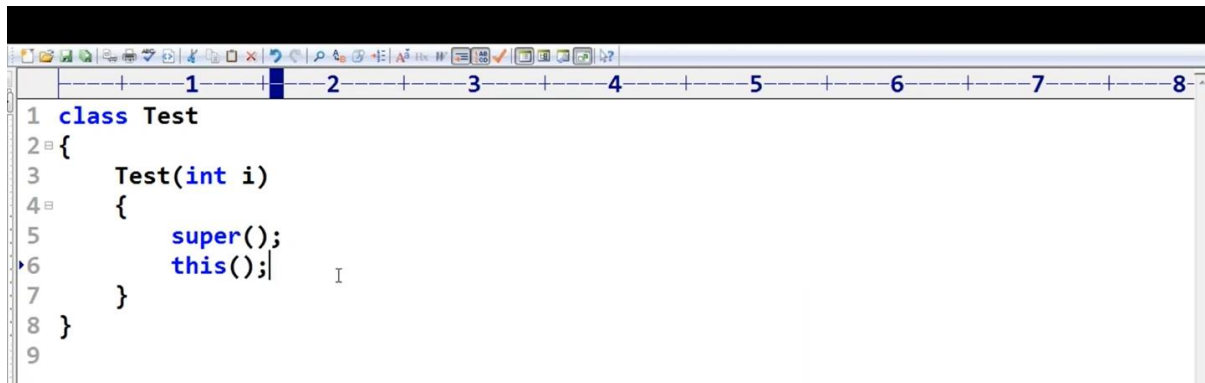
U can use either super() or this() but not both simultaneously , else CE



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

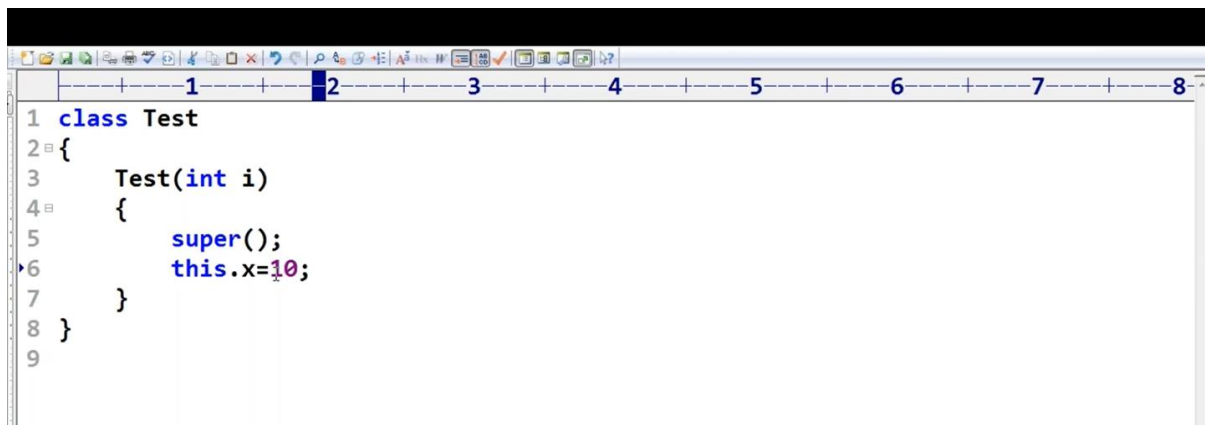
First line should be super or this, if you add it in second line or upcoming lines CE

Invalid



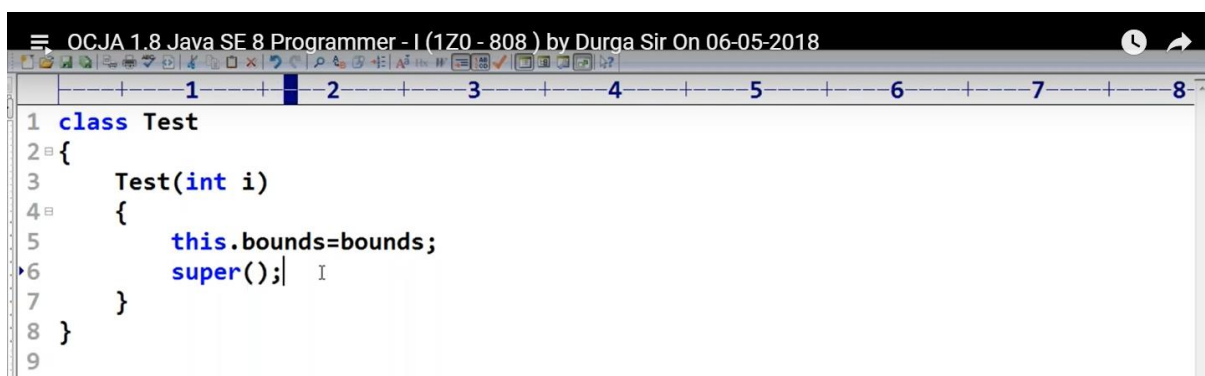
```
1 class Test
2 {
3     Test(int i)
4     {
5         super();
6         this();
7     }
8 }
9
```

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
```

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808 ) by Durga Sir On 06-05-2018
1 class Vehicle
2 {
3     String type="4w";
4     int maxSpeed=120;
5     Vehicle(String type,int maxSpeed)
6     {
7         this.type=type;
8         this.maxSpeed=maxSpeed;
9     }
10 }
```

```
11 class Car extends Vehicle
12 {
13     String trans;
14     Car(String trans)
15     {
16         //Line-1
17         this.trans=trans;
18     }
19     Car(String type,int maxSpeed,String trans)
20     {
21         super(type,maxSpeed);
22         this(trans); //Line-2
23     }
24 }
25
```

```
26 And given the code fragment:
27
28 Car c1= new Car("Auto");
29 Car c2= new Car("4w",150,"Manual");
30 System.out.println(c1.type+".."+c1.maxSpeed+".."+c1.trans);
31 System.out.println(c2.type+".."+c2.maxSpeed+".."+c3.trans);
32
33 What is the result?
```

Very Important

Line 16 – Compiler 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 Line 2 -> CE

CE – Line 2

```
OCJA 1.8 Java SE 8 Programmer - I (1Z0 - 808 ) by Durga Sir On 06-05-2018
1 class CD
2 {
3     int r;
4     CD(int r)
5     {
6         this.r=r;
7     }
8 }
9 class DVD extends CD
10 {
11     int c;
12     DVD(int r, int c)
13     {
14         //line-1
15     }
16 }
17
```

```
18 And Given the code Fragment:
19
20 DVD dvd= new DVD(10,20);
21
22 Which code fragment should be inserted at Line-1 to instantiate dvd object
23 successfully?
```

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

```
71
72 A. super.r=r;
73     this.c=c;
74
75 B. super(r);
76     this(c);
77
78 C. super(r);
79     this.c=c;
80
81 D. this.c=r;
82     super(c)
83
84 Answer: C
85
```

```
OCJA 1.8 Java SE 8 Programmer - I (170 - 808 ) by Durga Sir On 06-05-2018
1 public class Employee
2 {
3     String name;
4     boolean contract;
5     double salary;
6     Employee()
7     {
8         //line-1
9     }
10    public String toString()
11    {
12        return name+":"+contract+":"+salary;
13    }
14    public static void main(String[] args)
15    {
16        Employee e = new Employee();
17        //Line-2
18        System.out.println(e);
19    }
20 }
21 Which 2 modifications,when made independently,enable to the code to print
22 Durga:true:100.0
```

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

```
111 A. Replace line-2 with
112     e.name="Durga";
113     e.contract=true;
114     e.salary=100;
115
116 B. Replace line-2 with
117     this.name="Durga";
118     this.contract=true;
119     this.salary=100;
120
121 C. Replace line-1 with
122     this.name=new String("Durga");
123     this.contract= new Boolean(true);
124     this.salary= new Double(100);
125
126 D. Replace line-1 with
127     name="Durga";
128     contract=TRUE;
129     salary=100.0f;
130
131 E. Replace line-1 with:
132     this("Joe",true,100)
133
134 Answer: A and C
```

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



```
1 class A
2 {
3     public A()
4     {
5         System.out.println("A");
6     }
7 }
8 class B extends A
9 {
10    public B()
11    {
12        //line-1
13        System.out.println("B");
14    }
15 }
16 class C extends B
17 {
18     public C()
19     {
20        //line-2
21        System.out.println("C");
22    }
23     public static void main(String[] args)
24     {
25         C c = new C();
26     }
27 }
```

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

```
OCJA 1.8 Java SE 8 Programmer - I (170 - 808 ) by Durga Sir On 06-05-2018

1 class Vehicle
2 {
3     int x;
4     Vehicle()
5     {
6         this(10); // line-1
7     }
8     Vehicle(int x)
9     {
10        this.x=x;
11    }
12 }
13 class Car extends Vehicle
14 {
15     int y;
16     Car()
17     {
18         super();
19         this(20); //line-2
20     }
21     Car(int y)
22     {
23         this.y= y;
24     }
25     public String toString()
26     {
27         return super.x+" "+this.y;
28     }
29 }
30 And given the code fragment:
31
32 Vehicle v= new Car();
33 System.out.println(v);
```

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

Compilation fails at line 2

```

1 public class Person
2 {
3     String name;
4     int age=25;
5     public Person(String name)
6     {
7         this(); //line-1
8         setName(name);
9     }
10    public Person(String name,int age)
11    {
12        Person(name); //Line-2
13        setAge(age);
14    }
15    //setter and getter methods go here
16    public String show()
17    {
18        return name+ " "+age+ " "+number;
19    }
20    public static void main(String[] args)
21    {
22        Person p1= new Person("Durga");
23        Person p2= new Person("Ravi",50);
24        System.out.println(p1.show());
25        System.out.println(p2.show());
26    }
27 }

```

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?
32
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

```
296 A. Replace line-1 with
297     super();
298     this.bounds=bounds;
299 B. Repalce line-1 with
300     this.bounds=bounds;
301     super();
302 C. Replace line-2 with
303     super(type,maxSpeed);
304     this(bounds);
305 D. Repalce line-1 with
306     this("Canine",60);
307     this.bounds=bounds;
301
302 E. Replace line-2 with
303     super(type,maxSpeed);
304     this.bounds=bounds;
305
```

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
364
```

```
OC:JA 1.8 Java SE 8 Programmer - I (170 - 808 ) by Durga Sir On 06-05-2018

1 public class CheckingAccount
2 {
3     public int amount;
4     //line-1
5 }
6 And the given the following main method located in another class:
7
8 public static void main(String[] args)
9 {
10     CheckingAccount acct= new CheckingAccount();
11     //line-2
12 }
13 Which 3 pieces of code inserted independently, set the value of amount to 100?
14
15 A. At line-2 insert:
16     amount=100;
17
18 B. At line-2 insert:
19     this.amount=100;
20
21 C. At line-2 insert:
22     acct.amount=100;
23
24 D. At line-1 insert:
25     public CheckingAccount()
26     {
27         amount=100;
28     }
29
30 E. At line-1 insert:
31     public CheckingAccount()
32     {
33         this.amount=100;
34     }
35
36 F. At line-1 insert:
37     public CheckingAccount()
38     {
39         acct.amount=100;
40     }
```

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

