Vendor: Oracle

Exam Code: 1Z0-808

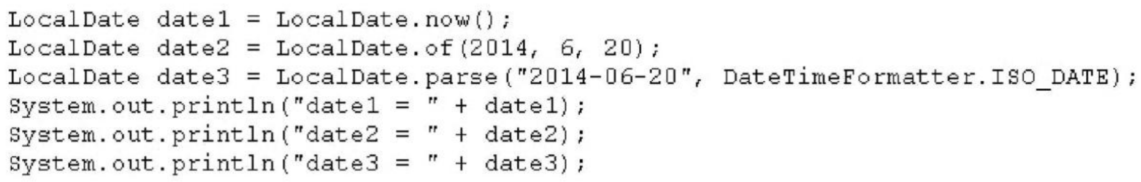
Exam Name: Java SE 8 Programmer I

Question 11—Question 20

Click to Download All 1Z0-808 Q&As From Lead2pass

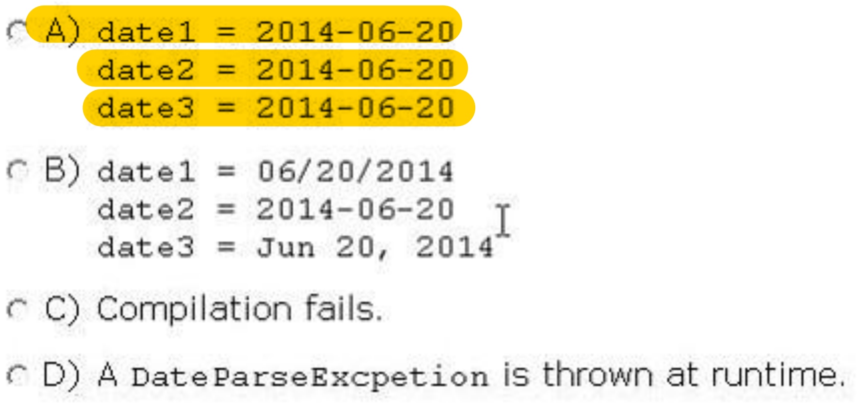
QUESTION 11

Given the code fragment:



```
LocalDate date1 = LocalDate.now();
LocalDate date2 = LocalDate.of(2014, 6, 20);
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);
System.out.println("date1 = " + date1);
System.out.println("date2 = " + date2);
System.out.println("date3 = " + date3);
```

Assume that the system date is June 20, 2014. What is the result?



```
○ A) date1 = 2014-06-20
     date2 = 2014-06-20
     date3 = 2014-06-20

○ B) date1 = 06/20/2014
     date2 = 2014-06-20
     date3 = Jun 20, 2014
```

○ C) Compilation fails.

○ D) A `DateParseExcpetion` is thrown at runtime.

A. Option A

B. Option B
C. Option C
D. Option D

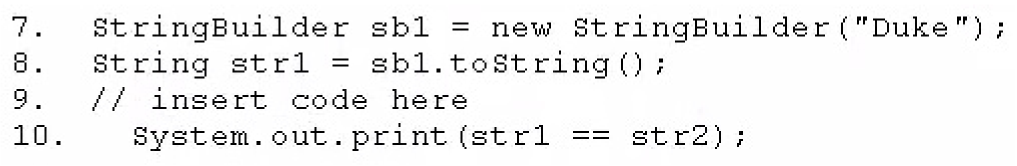
Answer: A
Explanation:
I've run the following code without any problem

```
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;

public class Main {
        public static void main(String[] args) {
                LocalDate date1 = LocalDate.now();
                LocalDate date2 = LocalDate.of(2014, 6, 20);
                LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);

                System.out.println("date 1 = " + date1);
                System.out.println("date 2 = " + date2);
                System.out.println("date 3 = " + date3);
        }

}
```

The output is

date 1 = 2015-09-05 (because run today, but problem statement says we must assume that the system data is June 20, 2014)
date 2 = 2014-06-20
date 3 = 2014-06-20

QUESTION 12
Given the code fragment:



```
StringBuilder sb1 = new StringBuilder("Duke");
String str1 = sb1.toString();
// insert code here
  System.out.print(str1 == str2);
```

Which code fragment, when inserted at line 9, enables the code to print true?

A. String str2 = str1;

B. String str2 = new String (str1);

C. String str2 = sb1. toString ();

D. String str2 = "Duke";

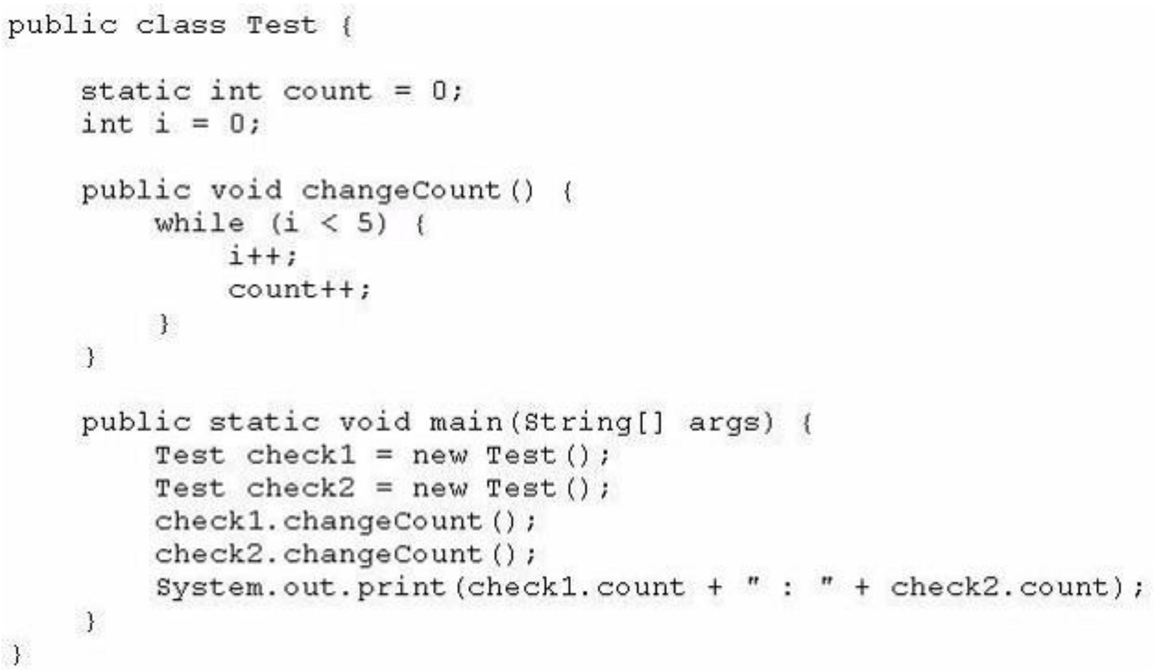
Answer: A

Explanation:

Operator == checks if two things are EXACTLY the same thing, not if they have the same content

QUESTION 13

Given the code fragment:



```
public class Test {

    static int count = 0;
    int i = 0;

    public void changeCount() {
        while (i < 5) {
            i++;
            count++;
        }
    }

    public static void main(String[] args) {
        Test check1 = new Test();
        Test check2 = new Test();
        check1.changeCount();
        check2.changeCount();
        System.out.print(check1.count + " : " + check2.count);
    }
}
```

What is the result?

A. 10 : 10

B. 5 : 5

C. 5 : 10

D. Compilation fails

Answer: A

Explanation:

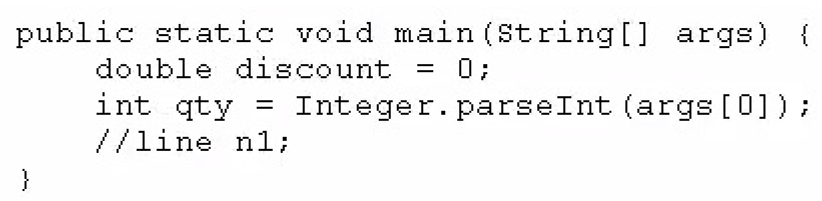
The variable i is local to all instances of class Test so each time we create an instance, i=0 and the loop add 5 to count.

The variable count (static) is global to all instances of class Test and all instances share the same variable. It's been initialized only once to zero an retains its value between the calls to changeCount

Since we call two times the method changeCount, the final result is 10 : 10

QUESTION 14

Given the code fragment:

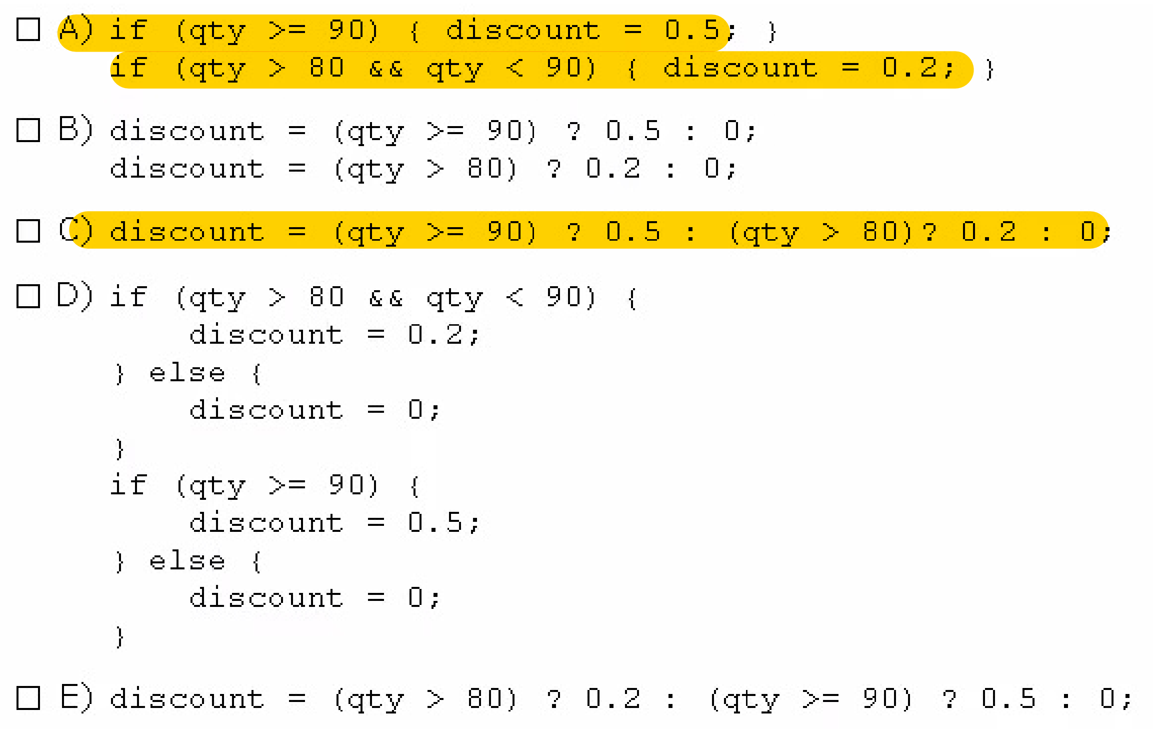


```
public static void main(String[] args) {
    double discount = 0;
    int qty = Integer.parseInt(args[0]);
    //line n1;
}
```

And given the requirements:

```
- If the value of the qty variable is greater than or equal to 90,
discount = 0.5
- If the value of the qty variable is between 80 and 90, discount =
0.2
```

Which two code fragments can be independently placed at line n1 to meet the requirements?



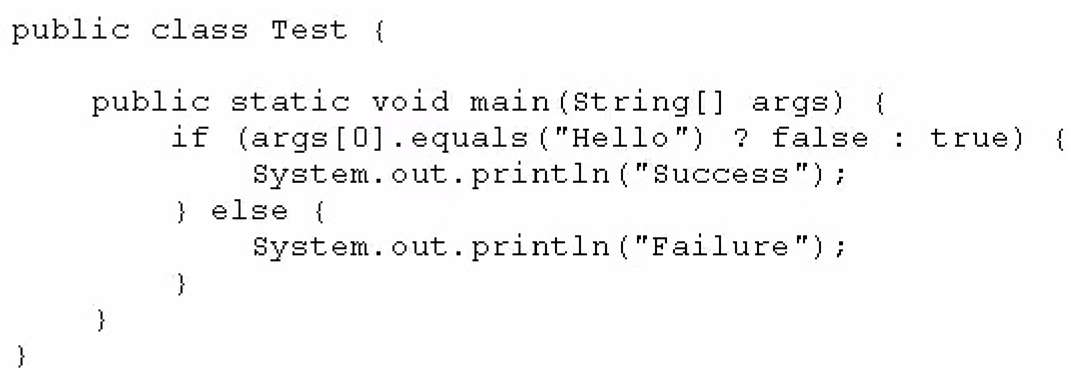
```
□ A) if (qty >= 90) { discount = 0.5; }
     if (qty > 80 && qty < 90) { discount = 0.2; }

□ B) discount = (qty >= 90) ? 0.5 : 0;
     discount = (qty > 80) ? 0.2 : 0;

□ C) discount = (qty >= 90) ? 0.5 : (qty > 80)? 0.2 : 0;

□ D) if (qty > 80 && qty < 90) {
         discount = 0.2;
     } else {
         discount = 0;
     }
     if (qty >= 90) {
         discount = 0.5;
     } else {
         discount = 0;
     }

□ E) discount = (qty > 80) ? 0.2 : (qty >= 90) ? 0.5 : 0;
```

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: AC

QUESTION 15

Given:



```
public class Test {

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

And given the commands:

```
javac Test.Java
Java Test Hello
```

What is the result?

A. Success
B. Failure
C. Compilation fails.
D. An exception is thrown at runtime

Answer: B

QUESTION 16

Which three statements describe the object-oriented features of the Java language?

A. Objects cannot be reused.
B. A subclass can inherit from a superclass.
C. Objects can share behaviors with other objects.
D. A package must contain more than one class.
E. Object is the root class of all other objects.
F. A main method must be declared in every class.

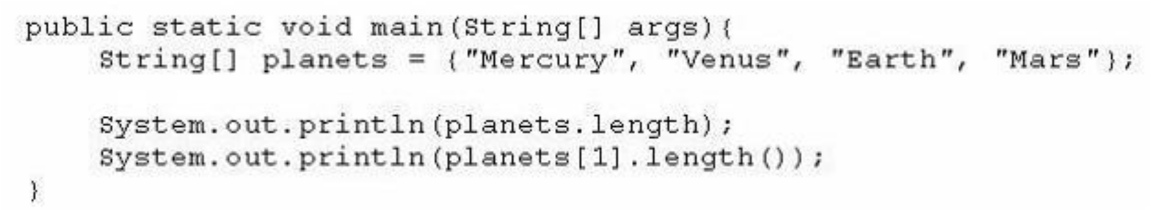
Answer: BCE

Explanation:

https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html http://www.artima.com/objectsandjava/webuscript/PolymorphismInterfaces1.html

QUESTION 17

Given the following code:



```
public static void main(String[] args){
    String[] planets = {"Mercury", "Venus", "Earth", "Mars"};

    System.out.println(planets.length);
    System.out.println(planets[1].length());
}
```

What is the output?

A. 4
   4

B. 3
   5

C. 4
   7

D. 5
   4

E. 4
   5

F. 4
   21

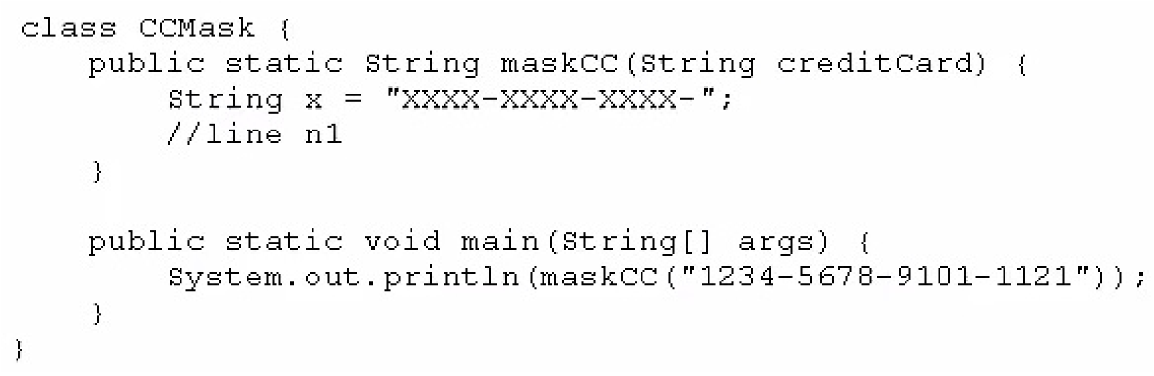
Answer: E

QUESTION 18

You are developing a banking module.

You have developed a class named ccMask that has a maskcc method.

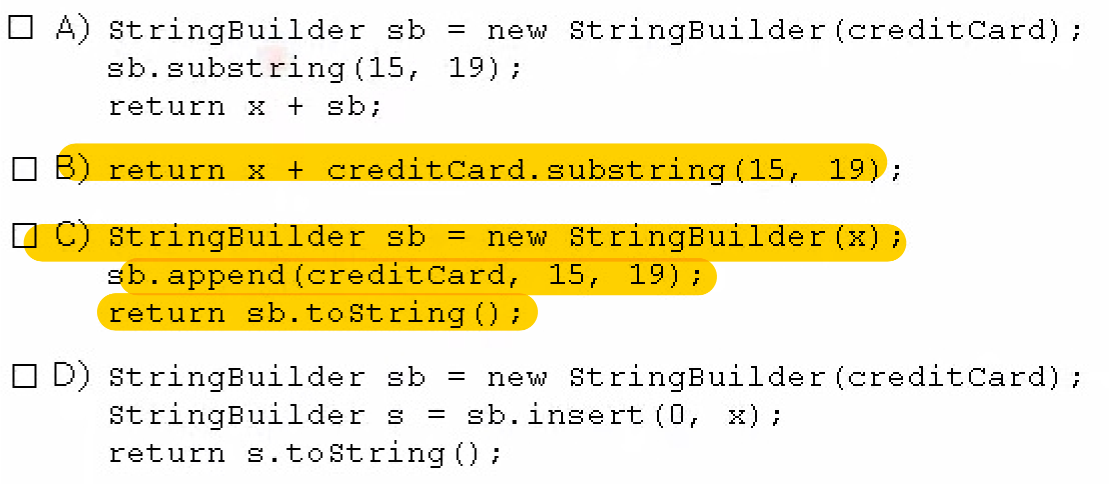
Given the code fragment:



```
class CCMask {
    public static String maskCC(String creditCard) {
        String x = "XXXX-XXXX-XXXX-";
        //line n1
    }

    public static void main(String[] args) {
        System.out.println(maskCC("1234-5678-9101-1121"));
    }
}
```

You must ensure that the maskcc method returns a string that hides all digits of the credit card number except the four last digits (and the hyphens that separate each group of four digits).

Which two code fragments should you use at line n1, independently, to achieve this requirement?



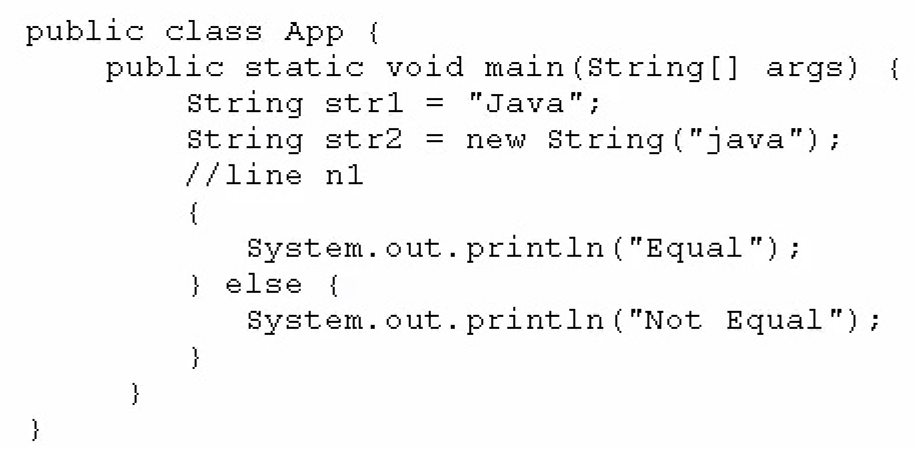
```
□ A) StringBuilder sb = new StringBuilder(creditCard);
     sb.substring(15, 19);
     return x + sb;

□ 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();
```

A. Option A
B. Option B
C. Option C
D. Option D

Answer: BC

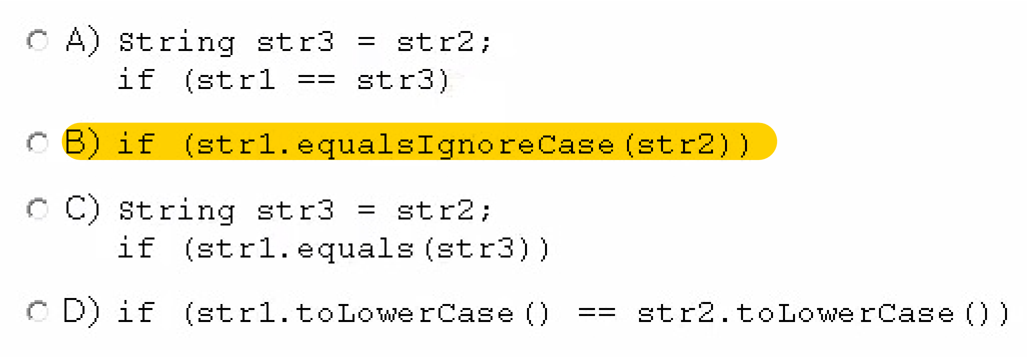
QUESTION 19

Given the code fragment:



```
public class App {
    public static void main(String[] args) {
        String str1 = "Java";
        String str2 = new String("java");
        //line n1
        {
           System.out.println("Equal");
        } else {
           System.out.println("Not Equal");
        }
     }
}
```

Which code fragment, when inserted at line n1, enables the App class to print Equal?



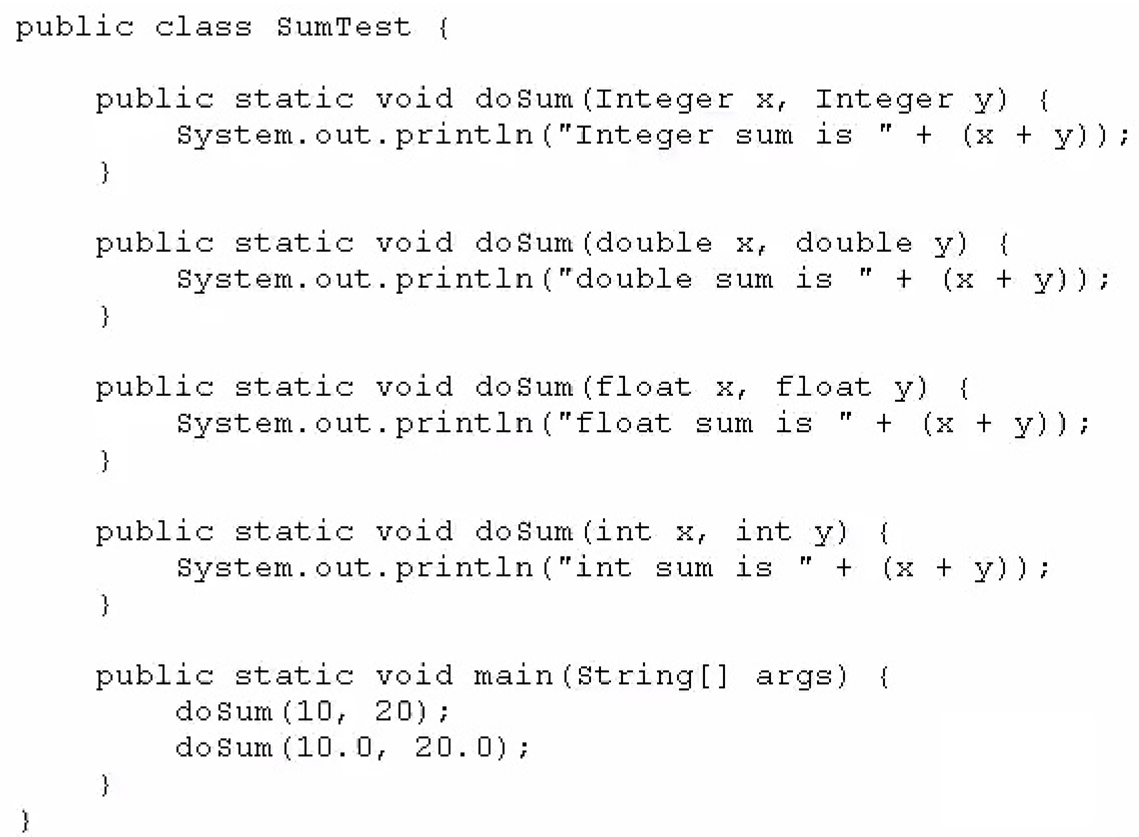
```
○ A) String str3 = str2;
     if (str1 == str3)

○ B) if (str1.equalsIgnoreCase(str2))

○ C) String str3 = str2;
     if (str1.equals(str3))

○ D) if (str1.toLowerCase() == str2.toLowerCase())
```

A. Option A
B. Option B
C. Option C
D. Option D

Answer: B

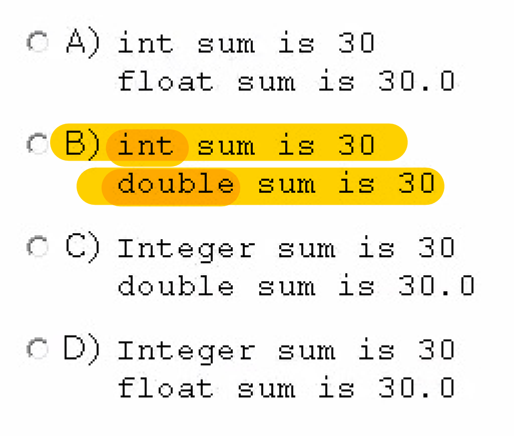
QUESTION 20

Given:



```
public class SumTest {

    public static void doSum(Integer x, Integer y) {
        System.out.println("Integer sum is " + (x + y));
    }

    public static void doSum(double x, double y) {
        System.out.println("double sum is " + (x + y));
    }

    public static void doSum(float x, float y) {
        System.out.println("float sum is " + (x + y));
    }

    public static void doSum(int x, int y) {
        System.out.println("int sum is " + (x + y));
    }

    public static void main(String[] args) {
        doSum(10, 20);
        doSum(10.0, 20.0);
    }
}
```

What is the result?



```
○ A) int sum is 30
     float sum is 30.0

○ B) int sum is 30
     double sum is 30

○ C) Integer sum is 30
     double sum is 30.0

○ D) Integer sum is 30
     float sum is 30.0
```

A. Option A

B. Option B

C. Option C

D. Option D

Answer: B

Explanation:

int is a primitive type and Integer is an object with an int. When we call doSum(10, 20), we are calling doSum(int, int).

By default, Java use double to represent its floating point literals. When we call doSum(10.0, 20.0), we are calling doSum(double, double).