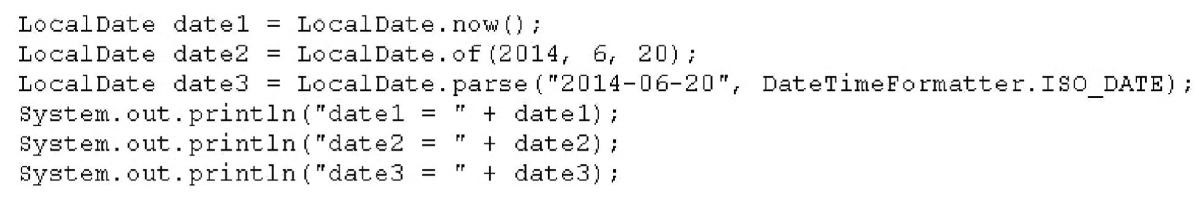
**Vendor:** Oracle

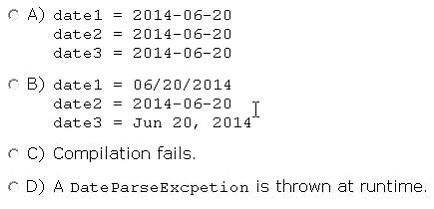
**Question 11—Question 20**

**QUESTION 11**

Given the code fragment:



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



1. Option A
2. Option B
3. Option C
4. 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);

}

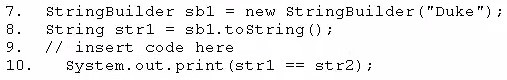
}

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:



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

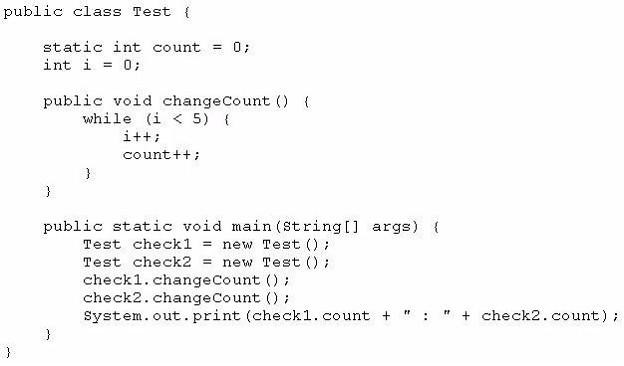
1. String str2 = str1;
2. String str2 = new String (str1);
3. String str2 = sb1. toString ();
4. String str2 = "Duke";

**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:



What is the result?

1. 10 : 10
2. 5 : 5
3. 5 : 10
4. Compilation fails

**Answer:** A **Explanation:**

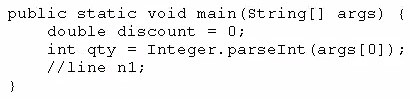
The variable i is local to all instances of class Test so each time we create an instance, i=0 and the loop add 5 to count.

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:

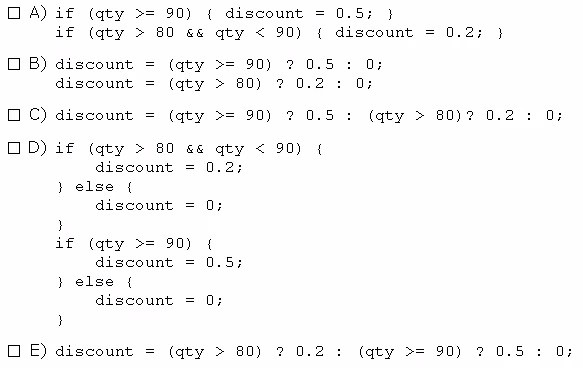


And given the requirements:

* 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?

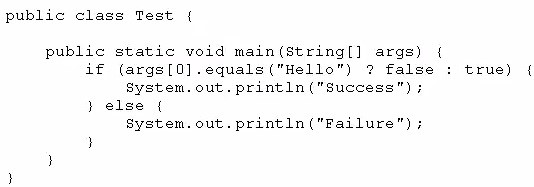


1. Option A
2. Option B
3. Option C
4. Option D
5. Option E

**Answer:** AC

**QUESTION 15**

Given:



And given the commands:

javac Test.Java

Java Test Hello

What is the result?

1. Success
2. Failure
3. Compilation fails.
4. An exception is thrown at runtime

**Answer:** B

**QUESTION 16**

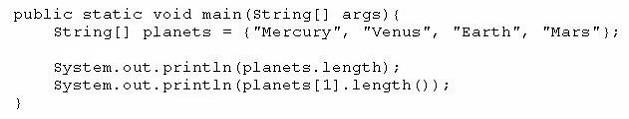
Which three statements describe the object-oriented features of the Java language?

1. Objects cannot be reused.
2. A subclass can inherit from a superclass.
3. Objects can share behaviors with other objects.
4. A package must contain more than one class.
5. Object is the root class of all other objects.
6. A main method must be declared in every class.

**Answer:** BCE **Explanation:**

**QUESTION 17**

Given the following code:



What is the output?

1. 4

4

1. 3

5

1. 4

7

1. 5

4

1. 4

5

1. 4

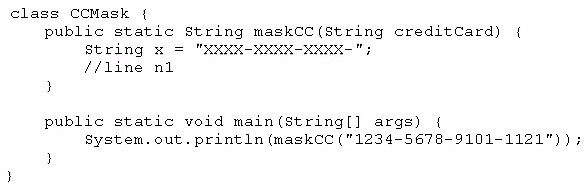
21

**Answer:** E

**QUESTION 18**

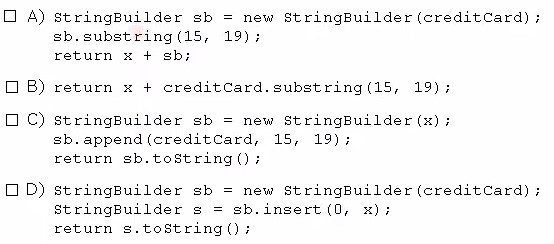
You are developing a banking module.

You have developed a class named ccMask that has a maskcc method. Given the code fragment:



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?

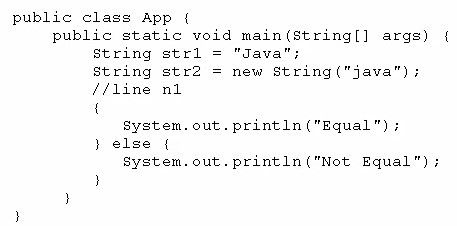


1. Option A
2. Option B
3. Option C
4. Option D

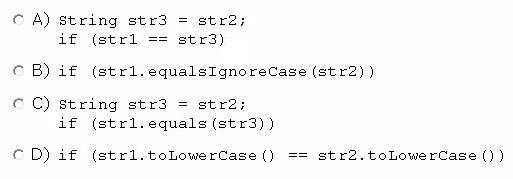
**Answer:** BC

**QUESTION 19**

Given the code fragment:



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

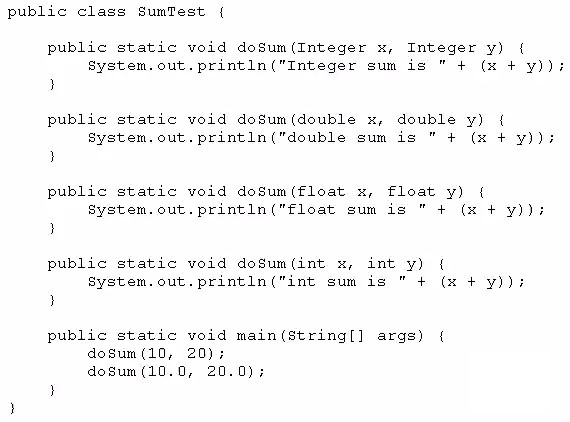


1. Option A
2. Option B
3. Option C
4. Option D

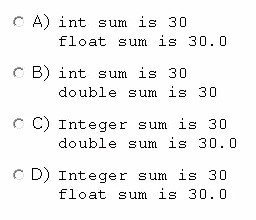
**Answer:** B

**QUESTION 20**

Given:



What is the result?



1. Option A
2. Option B
3. Option C
4. 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).