

QUESTION

How many String objects are created when the following method is invoked?

```
11. public String makingStrings() {  
12.     String s="Fred";  
13.     s=s+"47";  
14.     s=s.substring(2,5);  
15.     s=s.toUpperCase();  
16.     return s.toString();  
17. }
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 5
- F. 6

Correct Answer:

QUESTION

How many String objects are created when the following method is invoked?

```
11. public String makingStrings() {  
12.     String s="FRED";  
13.     s=s+"47";  
14.     s=s.substring(2,5);  
15.     s=s.toUpperCase();  
16.     return s.toString();  
17. }
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 5
- F. 6

Correct Answer:

QUESTION

Given the following code, when the last line of main method is reached, how many objects are eligible for GC?

```
public class ImmutableStrings
{
    public static void main(String[] args)
    {
        String one = "someString";
        String two = new String("someString");
        one = null;
        two = null;
    }
}
```

- A. 0 objects
- B. 1 object
- C. 2 objects
- D. Compilation fails.
- E. It is not possible to know
- F. An exception is thrown at runtime

Correct Answer:

QUESTION

Which of the given expressions is true, for the given code?

```
String s1 = new String("abc");
String s2 = new String("abc");
```

- A. `s1 == s2`
- B. `s1.equals(s2)`
- C. Both A and B
- D. Any answer is valid

Correct Answer:

QUESTION

Which of the given options is the output produced by the code given below?

```
String s1 = new String("hello");  
String s2 = "hello";  
String s3 = "hello";  
System.out.println(s1==s3);  
System.out.println(s2==s3);  
System.out.println(s1.equals(s2));
```

- A. true true false
- B. false true true
- C. false false true
- D. false true false
- E. true false false
- F. false false false
- G. true true true

Correct Answer:

QUESTION

What is displayed when the following code is compiled and executed?

```
String s1 = new String("Test");  
String s2 = new String("Test");  
if (s1==s2)  
    System.out.println("Same");  
if (s1.equals(s2))  
    System.out.println("Equals");
```

- A. Same
- B. Equals
- C. The code compiles, but nothing is displayed upon execution
- D. The code fails to compile

Correct Answer:

QUESTION

Consider the following program:

```
class StrEqual {  
    public static void main(String []args) {  
        String s1 = "hi";  
        String s2 = new String("hi");  
        String s3 = "hi";  
        if(s1 == s2) {  
            System.out.println("s1 and s2 equal");  
        } else {  
            System.out.println("s1 and s2 not equal");  
        }  
        if(s1 == s3) {  
            System.out.println("s1 and s3 equal");  
        } else {  
            System.out.println("s1 and s3 not equal");  
        }  
    }  
}
```

Which one of the following options provides the output of this program when executed?

- a)
s1 and s2 equal
s1 and s3 equal
- b)
s1 and s2 equal
s1 and s3 not equal
- c)
s1 and s2 not equal
s1 and s3 equal
- d)
s1 and s2 not equal
s1 and s3 not equal

Correct Answer:

QUESTION

Consider the following program and predict the output:

```
class Test {  
    public static void main(String []args) {  
        String s = new String("5");  
        System.out.println(1+10+s+1+10);  
    }  
}
```

- a) 11511
- b) 1105110
- c) 115110
- d) 27

Correct Answer:

QUESTION

Consider the following program and predict the output:

```
class Test {  
    public static void main(String []args) {  
        String s = new String("5");  
        System.out.println(1.0+10.5+s+(1.0+10.5));  
    }  
}
```

- a) 11.5511.5
- b) 11.551.010.5
- c) 1.010.551.010.5
- d) 11.55(1.010.5)
- e) 11.55(11.5)

Correct Answer:

QUESTION

Consider the following program:

```
class PrintlnTest {  
    public static void main(String[] args) {  
        String two = "2";  
        System.out.println("1 + 2 + 3 + 4 = "  
                            + 1 + Integer.parseInt(two) + 3 + 4); // PARSE  
    }  
}
```

Which one of the following options correctly describes the behavior of this program?

- a) When compiled, this program will give a compiler error in line marked with comment PARSE for missing catch handler for NumberFormatException.
- b) When executed, the program prints the following: $1 + 2 + 3 + 4 = 1234$.
- c) When executed, the program prints the following: $1 + 2 + 3 + 4 = 10$.
- d) When executed, the program prints the following: $1 + 2 + 3 + 4 = 127$.
- e) When executed, the program prints the following: $1 + 2 + 3 + 4 = 19$.
- f) When executed, the program throws a NumberFormatException in the line marked with comment PARSE.

Correct Answer:

QUESTION

Consider the following program:

```
class NullAccess {  
    public static void main(String []args) {  
        String str = null;  
        System.out.println(str.valueOf(10));  
    }  
}
```

Which of the following statements correctly describes the behavior of this program?

- a) This program will result in a compiler error.
- b) This program will throw a NullPointerException.
- c) This program will print 10 in console.
- d) This program will print null in console.

Correct Answer:

QUESTION

Consider the following code segment:

```
String str = "A.B.C!";  
System.out.println(str.replaceAll(".", ",").replace("!", "?"));
```

When executed, this code segment will print the following:

- a) A,B,C!
- b) A,B,C?
- c) ,,,,,
- d) A.B.C?

Correct Answer:

QUESTION

What will happen when you attempt to compile and run the following code snippet?

```
String str = "Java";
StringBuffer buffer = new StringBuffer(str);
if(str.equals(buffer))
{
    System.out.println("Both are equal");
}
else
{
    System.out.println("Both are not equal");
}
```

- A. It will print - Both are not equal
- B. It will print - Both are equal
- C. Compile time error as you can not use equals for objects of different classes
- D. Runtime error as you can not use equals for objects of different classes
- E. None of these

Correct Answer:

QUESTION

Given:

```
1. public class KungFu {
2.     public static void main(String[] args) {
3.         Integer x = 400;
4.         Integer y = x;
5.         x++;
6.         StringBuilder sb1 = new StringBuilder("123");
7.         StringBuilder sb2 = sb1;
8.         sb1.append("5");
9.         System.out.println((x == y) + " " + (sb1 == sb2));
10.    }
11. }
```

What is the result?

- A. true true
- B. false true
- C. true false
- D. false false
- E. Compilation fails
- F. An exception is thrown at runtime

Correct Answer:

QUESTION

Given this method in a class:

```
public String toString() {  
    StringBuffer buffer = new StringBuffer();  
    buffer.append("<");  
    buffer.append(this.name);  
    buffer.append(">");  
    return buffer.toString();  
}
```

Which is true?

- A. This code is NOT thread-safe
- B. The programmer can replace StringBuffer with StringBuilder with no other changes
- C. This code will perform well and converting the code to use StringBuilder will not enhance the performance
- D. This code will perform poorly. For better performance, the code should be rewritten: return "<"+this.name + ">";

Correct Answer:

QUESTION

Given:

```
1. public class MyLogger {  
2.     private StringBuilder logger = new StringBuilder();  
3.     public void log(String message, String user) {  
4.         logger.append(message);  
5.         logger.append(user);  
6.     }  
7. }
```

The programmer must guarantee that a single MyLogger object works properly for a multi-threaded system.

How must this code be changed to be thread-safe?

- A. synchronize the log method
- B. replace StringBuilder with StringBuffer
- C. No change is necessary, the current MyLogger code is already thread-safe
- D. replace StringBuilder with just a String object and use the string concatenation (+) within the log method

Correct Answer:

QUESTION

Consider the following program:

```
class SBAppend {
    public static void main(String []args) {
        Object nullObj = null;
        StringBuffer strBuffer = new StringBuffer(10);
        strBuffer.append("hello ");
        strBuffer.append("world ");
        strBuffer.append(nullObj);
        strBuffer.insert(11, '!');
        System.out.println(strBuffer);
    }
}
```

Which one of the following options correctly describes the behavior of this program?

- a) This program prints the following: hello world!
- b) This program prints the following: hello world! null
- c) This program throws a NullPointerException.
- d) This program throws an InvalidArgumentException.
- e) This program throws an ArrayIndexOutOfBoundsException

Correct Answer:

QUESTION

Consider the following program:

```
import java.util.*;
class AsList {
    public static void main(String []args) {
        String hello = "hello";
        String world = "world";
        StringBuffer helloWorld = new StringBuffer(hello + world);
        List<String> list = Arrays.asList(hello, world, helloWorld.toString());
        helloWorld.append("!");
        list.remove(0); // REMOVE
        System.out.println(list);
    }
}
```

Which one of the following options is correct?

- a) When compiled, this program will result in a compiler error in linked marked with comment REMOVE.

- b) When run, this program will crash with throwing the exception `UnsupportedOperationException` when executing the line marked with comment `REMOVE`.
- c) When run, this program will print the following output: [hello, world, helloworld]
- d) When run, this program will print the following output: [world, helloworld!]
- e) When run, this program will print the following output: [world, helloworld]

Correct Answer:

QUESTION

Consider the following code segment:

```
StringBuffer strBuffer = new StringBuffer("This, that, etc.!");
System.out.println(strBuffer.replace(12, 15, "etcetera"));
```

Which one of the following options correctly describes the behavior of this code segment?

- a) This code segment: This, that, etcetera.!
- b) This code segment: This, that, etcetera!
- c) This code segment: This, that, etc.
- d) This program throws in an `ArrayIndexOutOfBoundsException`.

Correct Answer:

QUESTION

Which two scenarios are NOT safe to replace a `StringBuffer` object with a `StringBuilder` object? (Choose two.)

- A. When using versions of Java technology earlier than 5.0
- B. When sharing a `StringBuffer` among multiple threads
- C. When using the `java.io` class `StringBufferInputStream`
- D. When you plan to reuse the `StringBuffer` to build more than one string

Correct Answer:

QUESTION

Given a code fragment:

```
StringBuilder sb = new StringBuilder();
String h1 = "HelloWorld";
sb.append("Hello").append("world");
if (h1 == sb.toString()) {
    System.out.println("They match");
}
if (h1.equals(sb.toString())) {
    System.out.println("They really match");
}
```

What is the result?

- A. They match
They really match
- B. They really match
- C. They match
- D. Nothing is printed to the screen

Correct Answer:

QUESTION

Given:

```
22. StringBuilder sb1 = new StringBuilder("123");
23. String s1 = "123";
24. // insert code here
25. System.out.println(sb1 + " " + s1);
```

Which code fragment, inserted at line 24, outputs "123abc 123abc"?

- A. sb1.append("abc");
s1.append("abc");
- B. sb1.append("abc");
s1.concat("abc");
- C. sb1.concat("abc");
s1.append("abc");
- D. sb1.concat("abc");
s1.concat("abc");
- E. sb1.append("abc");
s1 = s1.concat("abc");
- F. sb1.concat("abc");
s1 = s1.concat("abc");
- G. sb1.append("abc");
s1 = s1 + s1.concat("abc");
- H. sb1.concat("abc");
s1 = s1 + s1.concat("abc");

Correct Answer:

QUESTION

Given:

```
01. public class TestString3 {
02.     public static void main(String[] args) {
03.         // insert code here
04.         System.out.println(s);
05.     }
06. }
```

Which two code fragments, inserted independently at line 3, generate the output 4247?
(Choose two.)

- A. String s = "123456789";
s = (s-"123").replace(1,3,"24") - "89";
- B. StringBuffer s = new StringBuffer("123456789");
s.delete(0,3).replace(1,3,"24").delete(4,6);
- C. StringBuffer s = new StringBuffer("123456789");
s.substring(3,6).delete(1,3).insert(1, "24");
- D. StringBuilder s = new StringBuilder("123456789");
s.substring(3,6).delete(1,2).insert(1, "24");
- E. StringBuilder s = new StringBuilder("123456789");
s.delete(0,3).delete(1,3).delete(2,5).insert(1, "24");

Correct Answer:

QUESTION

Given:

```
class Feline {  
    public static void main(String[] args) {  
        Long x = 42L;  
        Long y = 44L;  
        System.out.print(" " + 7 + 2 + " ");  
        System.out.print(foo() + x + 5 + " ");  
        System.out.println(x + y + foo());  
    }  
    static String foo() {  
        return "foo";  
    }  
}
```

What is the result?

- A. 9 foo47 86foo
- B. 9 foo47 4244foo
- C. 9 foo425 86foo
- D. 9 foo425 4244foo
- E. 72 foo47 86foo
- F. 72 foo47 4244foo
- G. 72 foo425 86foo
- H. 72 foo425 4244foo
- I. Compilation fails

Correct Answer:

QUESTION

How many objects will be eligible for GC just after the method returns?

```
public void compute()
{
    Object a = new Object();
    int x = 100;
    String str = "abc";
}
```

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4

Correct Answer:

QUESTION

What is the result of compiling and running the following program.

```
public class test {
    public static void main(String args[]) {
        String str1="abc";
        String str2="def";
        String str3=str1.concat(str2);
        str1.concat(str2);
        System.out.println(str1);
    }
}
```

- A. abc
- B. def
- C. abcabc
- D. abcdef
- E. defabc
- F. abcdefdef

Correct Answer:

QUESTION

What is displayed when the following code is compiled and executed?

```
StringBuilder s1 = new StringBuilder("Test");
StringBuilder s2 = new StringBuilder("Test");
```

```
if (s1==s2)
    System.out.println("Same");
if (s1.equals(s2))
    System.out.println("Equals");
```

- A. Same
- B. Equals
- C. The code compiles, but nothing is displayed upon execution
- D. The code fails to compile

Correct Answer: