1. Which of the following declaration statements create a reference type? (Check all that are correct.)

Reference type is stored in the heap
Value type (primitive) is stores in the stack
int[] myArray; -- stored in the heap

Show Results 21/22 Students Answered

A int x = 5;
B boolean isOn = true;
C String message = "Hello";
D int[] scores = new int[5];
E I don't know.

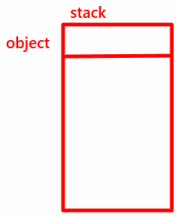
No Explanation

3. When a reference variable does not point to an object, it's value is _____. hull

Hide Answers Show Names 19/22 Students Answered

generic variable
null
null
null
null
null
null
null
primitive
null
null
null
null

MyClass object;



not until you finish:
object = new MyClass(); this is when reference to heap is added to stack

4. You can not change the value of a string. due to immutability

String name = "Margaret";

name = "Bob";

Show Results 19/22 Students Answered

True	true
False	

No Explanation

5. String s1 = "ALL MEN ARE CREATED EQUAL";
String s2 = s1.substring(16, 19);

System.out.println(s2);

What is the output of the preceding code snippet?

Hide Answers | Show Names | 19/22 Students Answered

s2 = TED

- 16, 19
- AD
- TED
- TE
- TED
- TED

6. char[] awesomeArray = new char[] { 'A', 'w', 'e', 's', 'o', 'm', 'e' };
String awesomeString1 = new String(awesomeArray);
String awesomeString2 = new String(awesomeArray);

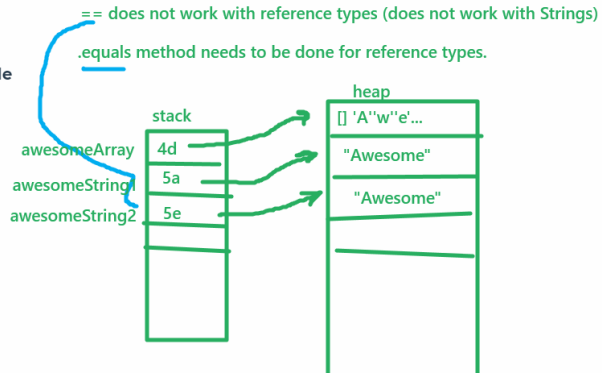
boolean areEqual = awesomeString1 == awesomeString2;

What is the value of the *areEqual* variable after this code executes?

Show Results | 19/22 Students Answered

- A true
- B false
- C I don't know

i No Explanation




```
public String stringX(String str) {
   String result = "";  // to hold my answer
   // if str contains 1 char, just return it.  if str contains 2 chars, just
return it.  EDGE CASES!
   if (str.length() <= 2){
      return str;
   }
   // we know to reach this spot in the code, we must have at least 3 chars
in str
   result += str.charAt(0);  // whether 'x' or not, we keep the first char
   // i starts with 1 because we have handled index 0 with line above
   for (int i = 1; i < str.length() - 1; i++){  // condition will stop 1 char
before last, so we can save that if 'x'
      if (str.charAt(i) != 'x'){
         result += str.charAt(i);
      }
   }
   // now have to append the last char
   result += str.charAt(str.length() - 1); // add last char whether 'x' or
not
```

```
    return result;
}
```