Java Software Development Quiz 6

1. Which of the following are wrapper classes?

Select the three correct answers.

(a) java.lang.Void

(b) java.lang.Int

(c) java.lang.Boolean

(d) java.lang.Long

(e) java.lang.String

2. What will the program print when compiled and run?

```
public class RQ200_70 {
    public static void main(String[] args) {
        Integer i = new Integer(-10);
        Integer j = new Integer(-10);
        Integer k = -10;
        System.out.print(i == j);
        System.out.print(i.equals(j));
        System.out.print(i == k);
        System.out.print(i.equals(k));
    }
}
```

Select the one correct answer.

(a) falsetruefalsetrue

(b) truetruetruetrue

(c) falsetruetruetrue

(d) truetruefalsetrue

(e) None of the above.

3. Which of the following operators cannot have an operand of type String?

Select the two correct answers.

(a) +

(b) -

(c) +=

(d) .

(e) &

4. Which of these expressions are legal?

Select the four correct answers.

(a) "co".concat("ol")

(b) ("co" + "ol")

(c) ('c' + 'o' + 'o' + 'l')

(d) ("co" + new String('o' + 'l'))

(e) ("co" + new String("co"))

5. Which expression will evaluate to true?

Select the one correct answer.

1. "hello: there!".equals("hello there")
2. "HELLO THERE".equals("hello there")
3. ("hello".concat("there")).equals("hello there")
4. "Hello There".compareTo("hello there") == 0
5. "Hello there".toLowerCase().equals("hello there")

6. What will the following program print when run?

```
public class Search {
    public static void main(String[] args) {
        String s = "Contentment!";
        int middle = s.length() / 2;
        String nt = s.substring(middle - 1, middle + 1);
        System.out.println(s.lastIndexOf(nt, middle));
    }
}
```

Select the one correct answer.

(a) 2

(b) 4

(c) 5

(d) 7

(e) 9

(f) 11

7. What will the following program print when run?

```
public class Uppity {
  public static void main(String[] args) {
    String str1 = "lower", str2 = "LOWER", str3 = "UPPER";
    str1.toUpperCase();
    str1.replace("LOWER","UPPER");
    System.out.println((str1.equals(str2)) + " " + (str1.equals(str3)));
  }
}
```

Select the one correct answer.

(a) The program will print false true.

(b) The program will print false false.

(c) The program will print true false.

(d) The program will print true true.

(e) The program will fail to compile.

(f) The program will compile, but throw an exception at runtime.

Answer

1. (a), (c), and (d)

The class java.lang.Void is considered a wrapper class, although it does not wrap any value. There is no class named java.lang.Int, but there is a wrapper class named java.lang.Integer. A class named java.lang.String also exists, but it is not a wrapper class since all strings in Java are objects.

2. (a)

Using the new operator creates a new object. Boxing also creates a new object if one is not already interned from before.

3. (b) and (e)

The operators - and & cannot be used in conjunction with a String object. The operators + and += perform concatenation on strings, and the dot operator accesses members of the String object.

4. (a), (b), (c), and (e)

The expressions ('c' + 'o' + 'o' + 'l') and ('o' + 'l') are of type int due to numeric promotion. Expression (d) is illegal, since the String class has no constructor taking a single int parameter. Expression (a) is legal, since string literals denote String objects and can be used just like any other object.

5. (e)

The expression "Hello there".toLowerCase().equals("hello there") will evaluate to true. The equals() method in the String class will only return true if the two strings have the same sequence of characters.

6. (c)

The variable middle is assigned the value 6. The variable nt is assigned the string "nt". The substring "nt" occurs three times in the string "Contentment!", starting at indices 2, 5, and 9. The call s.lastIndexOf(nt, middle) returns the start index of the last occurrence of "nt", searching backwards from position 6.

7. (b)

The reference value in the reference str1 never changes and it refers to the string literal "lower" all the time. The calls to toUpperCase() and replace() return a new String object whose reference value is ignored.