



Start Here Review Test Submission: Quiz 1: Basic Java syntax & classes

Review Test Submission: Quiz 1: Basic Java syntax & classes

User	Khasiano Webb
Course	WTC-DeepDiveJava_Android-57588-nbennett-Sept2019
Test	Quiz 1: Basic Java syntax & classes
Started	9/26/19 4:32 PM
Submitted	9/26/19 4:33 PM
Status	Completed
Attempt Score	125 out of 130 points
Time Elapsed	1 minute
Results Displayed	Submitted Answers, Feedback, Incorrectly Answered Questions

Question 1

5 out of 5 points



In Java, **String** is a synonym for **char[]** – that is, a **String** is simply an array of **char** values, and any given **char** element of a **String** can be accessed using array notation (square brackets).

Selected Answer: False

Response Feedback: While a Java **String** uses an array of **char** internally for storage (and is thus composed, *in part*, from a **char[]**), **String** and **char[]** are fundamentally different types of objects. Among other differences, individual **char** values in a **String** are accessed not by array notation, but via the **charAt(int)** method of the **String** class – which gives only read access to the specified **char** .

Question 2

10 out of 10 points



What is printed out when this code is executed? (You should assume the code is written as part of a method, in a class.)

```
int a = 5;
int b = 3 + 2;

if (a == b) {
    System.out.println("True");
} else {
    System.out.println("False");
}
```

← OK

Selected Answer: True

Response Feedback: The two `int` values shown in the code are equal. For comparing primitive values (such as `int` values), the `==` operator is appropriate.

Question 3

15 out of 15 points



Which of the following are required, in order to iterate over an array or collection using the *enhanced-for* syntax?

Selected Answers:

Iteration must be over an array of elements of a primitive type.

Iteration must be in "natural" order – that is, from start to finish.

The iteration index variable will not be available, and thus must not be required by the code in the body of the `for` loop.

The contents of the array or collection must not be modified in the body of the `for` loop.

Response Feedback:

An *enhanced-for* loop bypasses the initialization, condition, and update of an index (counter) variable, and directly assigns each element value in turn (in natural order) to the variable declared in the parentheses of the `for` statement. Changes to this variable in the body of the `for` have no effect on the array or collection contents.

Question 4

10 out of 10 points



```
String s1 = "CAT";  
String s2 = "cat".toUpperCase();
```

Which of these is the best way to compare the value of the two `String` instances, `s1` and `s2`, and perform some operation only if the contents of the two are equal?

Selected Answer:

```
if (s1.equals(s2)) {  
    // Do Something  
}
```

Response Feedback:

The `==` operator isn't a reliable way to compare `String` (or other object) instances, since it will return true only if the two arguments are at the same location in memory. The `equals()` method of the string class will compare the contents of two `String` instances and give the expected result.

Question 5

10 out of 10 points



```
for (int i = 1; i < 20; i *= 2) {  
    System.out.println(i);  
}
```

What is the final value printed by the fragment above? (Assume it's inside a method that has been invoked in the usual fashion.)

Selected Answer: 16

Response

Feedback: The `*=` operator is assignment augmented by multiplication. For example, `x *= a` is equivalent to `x = x * a`.

Given the above, the following values are printed:

1
2
4
8
16

Question 6

10 out of 10 points



Assume that the file `Simple.java` contains the code that follows, and has been compiled successfully to `Simple.class`:

```
package edu.cnm.deepdive;

import java.util.Arrays;

public class Simple {

    public static void main(String[] args) {
        System.out.println(Arrays.toString(args));
    }

}
```

Now, assume that the compiled class is launched as a Java application, using the command line (you should assume that the classpath and/or working directory are set correctly, so that the application launches successfully):

```
java edu.cnm.deepdive.Simple
```

Note that the command has no command line arguments after the fully qualified class name.

What output is printed?

Selected Answer: []

Response

Feedback: When the `main` method is invoked by the Java application launcher, and no arguments are provided after the class name on the command line, the value passed to `main` in the `String[]` parameter is an array of length 0 (**not** the value `null`).

When invoked with a non-`null` value, the `Arrays.toString` method returns a comma-delimited string of element values (if any), enclosed within bracket

characters.

Question 7

10 out of 10 points



What must the condition (indicated by the gray section) be, so that the following fragment prints out the integers 5 through 15, inclusive ?

```
for (int j = 5; _____; j++) {  
    System.out.print(j + " ");  
}  
System.out.println();
```

Selected Answer: `j < 16`

Response Feedback: Remember that the condition is tested just before the body of the `for` is executed; if the condition fails, iteration stops. In order to print 15 as the last value, the condition must succeed when `j` is 15, and fail when `j` is 16.

Question 8

10 out of 10 points



Which tool in the JDK processes specially-formatted comments in Java source files to generates HTML pages and CSS files?

Selected Answer: `javadoc`

Response Feedback: See <https://docs.oracle.com/en/java/javase/11/tools/tools-and-command-reference.html> for details on the purpose and use of the primary tools in the JDK.

Question 9

10 out of 10 points



Which method will return `true` when (and only when) the value passed in the argument is between 10 and 20, inclusive?

Selected Answer:

```
public boolean isInRange(int x) {  
    return (x >= 10 && x <= 20);  
}
```

Response Feedback: In Java, to test whether a number `x` is between an upper and lower limit (both inclusive), you must use the greater-than-or-equal (`>=`) and less-than-or-equal (`<=`) operators for the lower and upper limits (respectively), and the `&&` (logical and) operator to combine the conditions.

Question 10

10 out of 10 points



Given the declaration `int[] arr = {1,2,3,4,5,6}` what is the value of `arr[3]`?

Selected Answer: 4

Response Feedback: The element at index 3 has a value of 4

Question 11

5 out of 5 points



In the general *switch* statement syntax,

```
switch (switchExpression) {  
    case value1:  
        // Do something for the value1 case.  
    case value2:  
        // Do something for the value2 case.  
    // etc.  
}
```

The expression represented by the *switchExpression* placeholder can only be of the `int` or the `String` type.

Selected Answer: False

Response Feedback: Switch expressions can be of the primitive integer types *except* `long` (`byte`, `short`, `char`, `int`), any of the wrapper types for those primitives (`Byte`, `Short`, `Character`, `Integer`), the `String` type, or any `enum` type.

Question 12

10 out of 10 points



The first line of the `TimeSeries.java` file is

```
package com.bigbiz.tools;
```

Assume that `TimeSeries.java` is compiled successfully to `TimeSeries.class`. Further, assume that the classpath is set (either with the `-cp` or `-classpath` option of the `java` command, or with the `CLASSPATH` environment variable) to the single location `/projects/forecasting`. Which of the following is a possible location of the compiled `TimeSeries.class` file, where the `TimeSeries` class could be loaded by the default class loader of the JVM?

(Note that the above, and the questions below, use Unix-style forward slashes. Windows users may assume that we would specify these with backslashes. In other words, don't assume that an answer is incorrect simply because it uses forward slashes instead of backslashes.)

Selected Answer: `/projects/forecasting/com/bigbiz/tools/TimeSeries.class`

Response Feedback: In order to load a class, the default class loader must be able to find the compiled (`.class`) file in a directory corresponding to the package name, relative to one of the entries in the classpath. In this case, since the classpath is set to `/projects/forecasting`, and the fully qualified class name is `com.bigbiz.tools.TimeSeries`, that compiled class must be in the `/projects/forecasting/com/bigbiz/tools/TimeSeries.class` file.

Question 13

0 out of 5 points



If a Java source file (.java) contains only declarations of **public static** variables or **public static final** constants, without them being inside the body of a **class**, **interface**, or **enum**, Java will treat these as global variables or global constants.

Selected Answer: True

Response
Feedback: Java doesn't allow "bare" variable or constant declarations; any such declarations must be located directly within **class**, **enum**, or **interface** bodies – or nested further, in method or statement bodies.

Question 14

10 out of 10 points



What is printed out when this code is executed?

```
int test = 5;
String result = "";

switch(test) {
    case 3:
        result = "Brown Fox ";
    case 5:
        result = "Happy Cow ";
        break;
    case 2:
        result = "Lazy Dog ";
    default:
        result = "White Wolf ";
        break;
}

System.out.println(result);
```

Selected Answer: Happy Cow

Response
Feedback: The code inside the **switch** statement is executed starting at **case 5**: and breaks out of the switch when the **break** statement is executed.

Thursday, October 10, 2019 8:45:20 AM MDT