Jb0190r: Review: Using the System and PrintStream Classes[*]

R.G. (Dick) Baldwin

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 3.0[†]

Abstract

This module contains review questions and answers keyed to the module titled Jb0190: Java OOP:
Using the System and PrintStream Classes

1 Table of Contents

2 Preface

This module contains review questions and answers keyed to the module titled Jb0190: Java OOP: Using the System and PrintStream Classes [1] .

The questions and the answers are connected by hyperlinks to make it easy for you to navigate from the question to the answer and back again.

3 Questions

3.1 Question 1 .

True or false? The **main** method in the controlling class of a Java application controls the flow of the program.

Answer 1 (p. 7)

3.2 Question 2

True or false? The **main** method cannot access the variables and methods of objects instantiated from other classes.

Answer 2 (p. 7)

3.3 Question 3

True or false? The **main** method must instantiate objects of other classes in order for the program to execute.

Answer 3 (p. 7)

3.4 Question 4

True or false? In order to be useful, the **System** class must be used to instantiate objects in a Java application.

Answer 4 (p. 7)

3.5 Question 5

True or false? *Class* variables such as the **out** variable of the **System** class must be of some specific type.

Answer 5 (p. 7)

3.6 Question 6

True or false? *Class* variables must be of a primitive type such as **int** or **float**.

Answer 6 (p. 6)

3.7 Question 7

True or false? The **out** variable in the **System** class is of a primitive type.

Answer 7 (p. 6)

3.8 Question 8

True or false? What does the following code fragment access?

NOTE:

```
System.out
```

Answer 8 (p. 6)

3.9 Question 9

True or false? An object of type **PrintStream** is automatically instantiated when the **System** class is loaded into an application.

Answer 9 (p. 6)

3.10 Question 10

True or false? The **out** variable in the **System** class refers to an instance of what class?

Answer 10 (p. 6)

3.11 Question 11

True or false? The **println** method is an instance method of what class?
    Answer 11 (p. 6)

3.12 Question 12

What is the primary behavior of the **println** method?
    Answer 12 (p. 6)

3.13 Question 13

How can the **println** method be accessed?
    Answer 13 (p. 6)

3.14 Question 14

Assuming that the standard output device has not been redirected, write a code fragment that will cause
your name to be displayed on the screen.
    Answer 14 (p. 6)

3.15 Question 15

Explain how your code fragment in Answer 14 (p. 6) produces the desired result.
    Answer 15 (p. 5)

3.16 Question 16

If you have a class named **MyClass** that has a class variable named **myClassVariable** that requires
four bytes of memory and you instantiate ten objects of type **MyClass** , how much total memory will
be allocated to contain the allocated variables *(assume that the class definition contains no other class,
instance, or local variables)* .
    Answer 16 (p. 5)

3.17 Question 17

How many actual instances of the variable named **out** are allocated in memory by the following code
fragment?

    NOTE:

```
System.out.println("Dick Baldwin");
```

Answer 17 (p. 5)

3.18 Question 18

If you have a class named **MyClass** that has an instance variable named **myInstanceVariable** that
requires four bytes of memory and you instantiate ten objects of type **MyClass** , how much total memory
will be allocated to contain the allocated variables *(assume that the class definition contains no other class,
instance, or local variables)* .
    Answer 18 (p. 5)
    What is the meaning of the following two images?

This image was inserted here simply to insert some space between the questions and the answers to keep them from being visible on the screen at the same time.

The image is also an example of the kinds of things that we do in my course titled ITSE 2321, Object-Oriented Programming.

This image was also inserted for the purpose of inserting space between the questions and the answers.

4 Answers

4.1 Answer 18

Every instance of a class has its own set of instance variables. You can only access *instance variables* and *instance methods* through an object of the class. In this case, forty bytes of memory would be required to contain the instance variables of the ten objects.

Back to Question 18 (p. 3)

4.2 Answer 17

Only one, because **out** is a class variable of the **System** class.

Back to Question 17 (p. 3)

4.3 Answer 16

The runtime system allocates a *class variable* only once no matter how many instances of the class are instantiated. Thus, all objects of the class share the same physical memory space for the class variable, and in this case, only four bytes of memory will be allocated to contain the allocated variables.

Back to Question 16 (p. 3)

4.4 Answer 15

The statement in Answer 14 (p. 6) calls the **println** method belonging to an object of the **PrintStream** class, which is referenced *(pointed to)* by the **out** variable, which is a *class* variable of the **System**

class.
    Back to Question 15 (p. 3)

4.5 Answer 14

NOTE:

```
    System.out.println("Dick Baldwin");
```

Back to Question 14 (p. 3)

4.6 Answer 13

The **println** method can be accessed by joining the name of a variable that references a **PrintStream** object to the name of the **println** method using a period.
    Back to Question 13 (p. 3)

4.7 Answer 12

The **println** method causes its argument to be displayed on the standard output device. *(The standard output device is the screen by default, but can be redirected by the user at the operating system level.)*
    Back to Question 12 (p. 3)

4.8 Answer 11

The **println** method is an instance method of the **PrintStream** class.
    Back to Question 11 (p. 3)

4.9 Answer 10

The **out** variable in the **System** class refers to an instance of the **PrintStream** class *(a PrintStream object),* which is automatically instantiated when the **System** class is loaded into the application.
    Back to Question 10 (p. 2)

4.10 Answer 9

True.
    Back to Question 9 (p. 2)

4.11 Answer 8

The code fragment accesses the contents of the *class* variable named **out** in the class named **System** .
    Back to Question 8 (p. 2)

4.12 Answer 7

False. the variable named **out** defined in the **System** class is a reference variable that points to an object of another type.
    Back to Question 7 (p. 2)

4.13 Answer 6

False. A *class* variable can be a primitive type, or it can be a reference variable that points to another object.
    Back to Question 6 (p. 2)

4.14 Answer 5

True.
    Back to Question 5 (p. 2)

4.15 Answer 4

False. The **System** class has several *class* variables *(including **out** and **in**)* that are useful without
the requirement to instantiate an object of the **System** class.
    Back to Question 4 (p. 2)

4.16 Answer 3

False. While it is probably true that the **main** method must instantiate objects of other classes in order
to accomplish much that is of value, this is not a requirement. The **main** method in the "Hello World"
program of this module [2] does not instantiate objects of any class at all.
    Back to Question 3 (p. 2)

4.17 Answer 2

False. The **main** method can access the variables and methods of objects instantiated from other classes.
Otherwise, the flow of the program would be stuck within the **main** method itself and wouldn't be very
useful.
    Back to Question 2 (p. 2)

4.18 Answer 1

True.
    Back to Question 1 (p. 1)

5 Miscellaneous

This section contains a variety of miscellaneous information.

> NOTE: **Housekeeping material**
>
> - Module name: Jb0190r: Review: Using the System and PrintStream Classes
> - File: Jb0190r.htm
> - Published: November 22, 2012

> NOTE: **Disclaimers: Financial** : Although the Connexions site makes it possible for you to
> download a PDF file for this module at no charge, and also makes it possible for you to purchase a
> pre-printed version of the PDF file, you should be aware that some of the HTML elements in this
> module may not translate well into PDF.
>
> I also want you to know that, I receive no financial compensation from the Connexions website even
> if you purchase the PDF version of the module.
>
> In the past, unknown individuals have copied my modules from cnx.org, converted them to Kindle
> books, and placed them for sale on Amazon.com showing me as the author. I neither receive
> compensation for those sales nor do I know who does receive compensation. If you purchase such
> a book, please be aware that it is a copy of a module that is freely available on cnx.org and that it
> was made and published without my prior knowledge.

---

[2]http://cnx.org/content/m45148/latest/

Affiliation : I am a professor of Computer Information Technology at Austin Community College in Austin, TX.

-end-