

JB0180R REVIEW: THE MAIN METHOD.*

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 Jb0180: Java OOP: The main method.

1 Table of Contents

- Preface (p. 1)
- Questions (p. 1)
 - 1 (p. 1) , 2 (p. 1) , 3 (p. 2) , 4 (p. 2) , 5 (p. 2) , 6 (p. 2) , 7 (p. 2) , 8 (p. 2)
- Answers (p. 4)
- Miscellaneous (p. 5)

2 Preface

This module contains review questions and answers keyed to the module titled Jb0180: Java OOP: The main method 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 .

Write the method signature for the **main** method in a Java application.

Answer 1 (p. 5)

3.2 Question 2

Briefly explain the reason that the **main** method in a Java application is declared **public** .

Answer 2 (p. 5)

*Version 1.2: Nov 21, 2012 7:37 pm -0600

[†]<http://creativecommons.org/licenses/by/3.0/>

¹<http://cnx.org/content/m45145>

3.3 Question 3

Explain the reason that the **main** method in a Java application must be declared **static** .

Answer 3 (p. 5)

3.4 Question 4

Describe the purpose of the keyword **void** when used as the return type for the **main** method.

Answer 4 (p. 5)

3.5 Question 5

True or false? If the Java application is not designed to use command-line arguments, it is not necessary to include a formal parameter for the **main** method.

Answer 5 (p. 4)

3.6 Question 6

True or false? When using command-line arguments in Java, if the name of the string array is **args** , the **args.length** variable can be used by the code in the program to determine the number of arguments actually entered.

Answer 6 (p. 4)

3.7 Question 7

True or false? The first string in the array of command-line arguments contains the name of the Java application

Answer 7 (p. 4)

3.8 Question 8

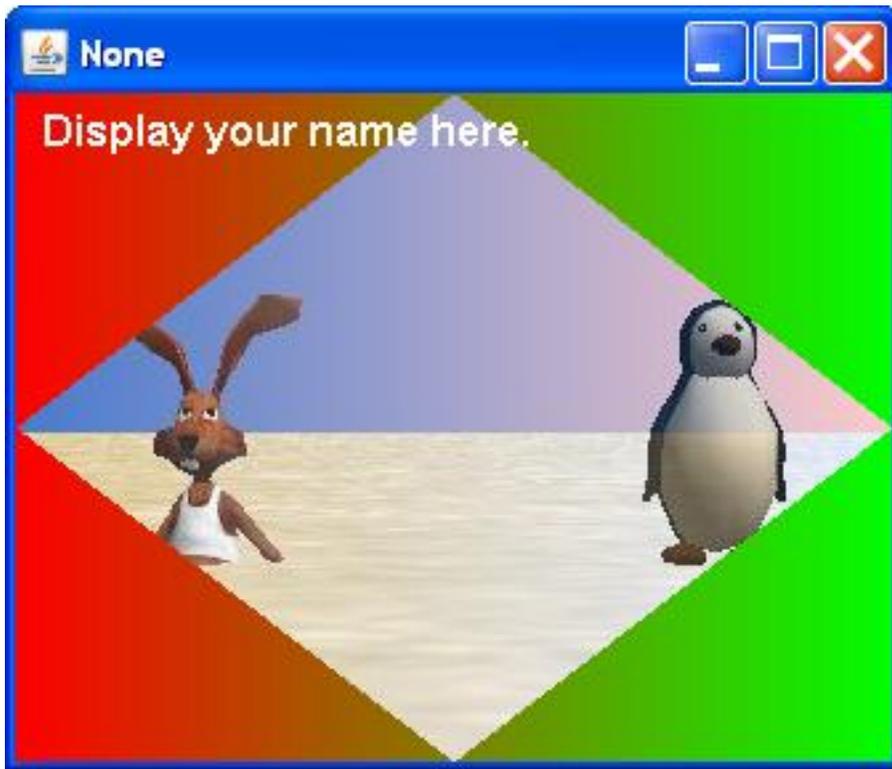
The *controlling class* of every Java application must contain a **main** method. Can other classes in the same application also have a **main** method? If not, why not? If so, why might you want to do that?

Answer 8 (p. 4)

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 8

Any and all classes in a Java application can have a **main** method. Only the one in the *controlling class* for the program being executed is actually called.

It is often desirable to provide a **main** method for a class that will not ultimately be the *controlling class* to allow the class to be tested in a stand-alone mode, independent of other classes.

Back to Question 8 (p. 2)

4.2 Answer 7

False. Unlike C++, the first string in the array of command-line arguments in a Java application does **not** contain the name of the application.

Back to Question 7 (p. 2)

4.3 Answer 6

True.

Back to Question 6 (p. 2)

4.4 Answer 5

False. The **main** method in a Java program must always provide the formal argument list regardless of whether it is actually used in the program.

Back to Question 5 (p. 2)

4.5 Answer 4

The **void** keyword when used as the return type for any Java method indicates that the method does not **return** anything.

Back to Question 4 (p. 2)

4.6 Answer 3

The keyword **static** indicates that the method is a *class* method which can be called without the requirement to instantiate an object of the class. This is used by the Java virtual machine to launch the program by calling the **main** method of the class identified in the command to start the program.

Back to Question 3 (p. 2)

4.7 Answer 2

The keyword **public** indicates that the method can be called by any object.

Back to Question 2 (p. 1)

4.8 Answer 1

NOTE:

```
public static void main(String[] args)
```

Back to Question 1 (p. 1)

5 Miscellaneous

This section contains a variety of miscellaneous information.

NOTE: **Housekeeping material**

- Module name: Jb0180r Review: The main method
- File: Jb0180r.htm
- Published: November 21, 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.

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

-end-