

JB0260R REVIEW: COMMAND-LINE ARGUMENTS*

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 Jb0260: Java OOP: Command-Line Arguments.

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)
- Listings (p. 2)
- Answers (p. 4)
- Miscellaneous (p. 5)

2 Preface

This module contains review questions and answers keyed to the module titled Jb0260: Java OOP: Command-Line Arguments 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 .

Provide a common example of a command-line statement that illustrates the use of command-line-arguments.

Answer 1 (p. 5)

3.2 Question 2

Describe the purpose of command-line-arguments.

Answer 2 (p. 5)

*Version 1.2: Nov 27, 2012 12:18 pm -0600

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

¹<http://cnx.org/content/m45246>

3.3 Question 3

True or false? In Java, syntax provisions must be made in the method signature for the **main** method to accommodate command-line-arguments even if the remainder of the program is not designed to make use of them. If False, explain why.

Answer 3 (p. 5)

3.4 Question 4

Provide the method signature for the **main** method in a Java application that is designed to accommodate the use of command-line-arguments. Identify the part of the method signature that applies to command-line-arguments and explain how it works.

Answer 4 (p. 5)

3.5 Question 5

Explain how a Java application can determine the number of command-line-arguments actually entered by the user.

Answer 5 (p. 5)

3.6 Question 6

Write a program that illustrates the handling of command-line arguments in Java.

Answer 6 (p. 4)

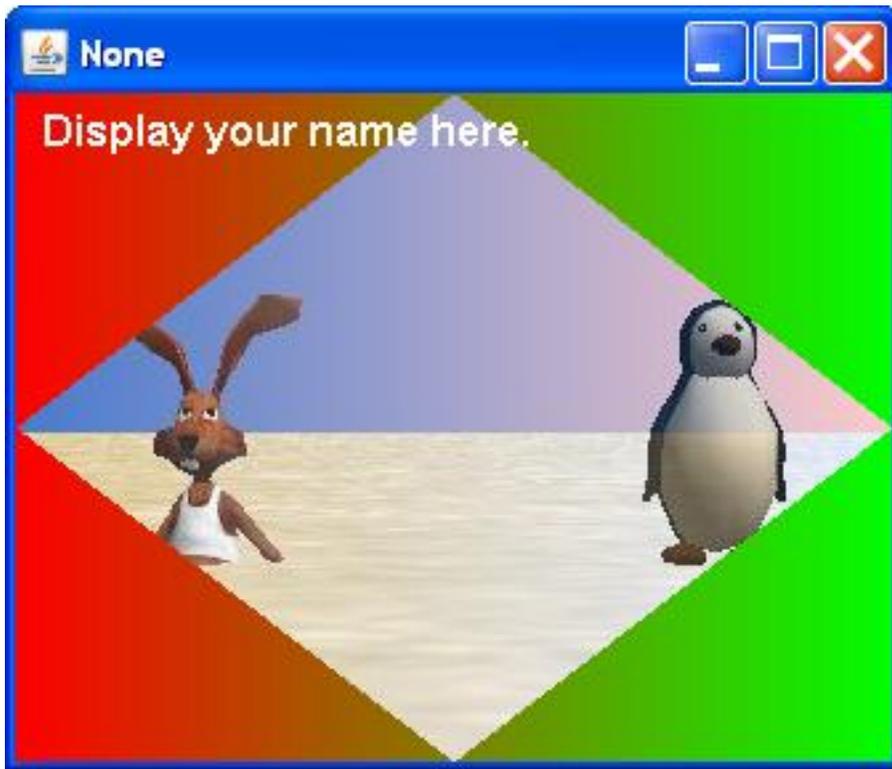
4 Listings

- Listing 1 (p. 4) . Handling command-line arguments in Java.

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.



5 Answers

5.1 Answer 6

Listing 1: Handling command-line arguments in Java.

```

/*File SampProg11.java from module 32
Copyright 1997, R.G.Baldwin
Without reviewing the following solution, write a Java
application that illustrates the handling of command-line
arguments in Java.

```

```

Provide a termination message that displays your name.
*****/
class SampProg11 { //define the controlling class
    public static void main(String[] args){ //define main
        for(int i=0; i < args.length; i++)
            System.out.println( args[i] );
        System.out.println("Terminating, Dick Baldwin");
    } //end main
} //End SampProg11 class.

```

Back to Question 6 (p. 2)

5.2 Answer 5

The number of command-line arguments is equal to the number of elements in the array of references to **String** objects referred to by **args** . The number of elements is indicated by the value of the **length** property of the array. If the value is zero, the user didn't enter any command-line arguments.

Back to Question 5 (p. 2)

5.3 Answer 4

The Java syntax for command-line arguments is shown below.

NOTE: **Java syntax for command-line arguments.**

```
public static void main(String[] args){  
    . . .  
} //end main method
```

Each of the elements in the array object referred to by **args** (*including the element at position zero*) contains a reference to a **String** object that encapsulates one of the command-line arguments.

Back to Question 4 (p. 2)

5.4 Answer 3

True.

Back to Question 3 (p. 2)

5.5 Answer 2

Command-line-arguments are used in many programming and computing environments to provide information to the program at startup that it will need to fulfill its mission during that particular invocation.

Back to Question 2 (p. 1)

5.6 Answer 1

DOS users will be familiar with commands such as the following:

NOTE: **Command-line arguments in DOS**

```
copy fileA fileB
```

In this case, **copy** is the name of the program to be executed, while **fileA** and **fileB** are command-line arguments.

Back to Question 1 (p. 1)

6 Miscellaneous

This section contains a variety of miscellaneous information.

NOTE: **Housekeeping material**

- Module name: Jb0260r Review: Command-Line Arguments
- File: Jb0260r.htm
- Originally published: 1997
- Published at cnx.org: November 27, 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-