

JB0230R REVIEW: FLOW OF CONTROL*

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 Jb0230: Java OOP: Flow of Control.

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) , 9 (p. 2) , 10 (p. 2) , 11 (p. 2) , 12 (p. 6) , 13 (p. 3) , 14 (p. 3) , 15 (p. 3) , 16 (p. 3) , 17 (p. 3) , 18 (p. 3) , 19 (p. 3) , 20 (p. 3) , 21 (p. 3)
- Answers (p. 5)
- Miscellaneous (p. 9)

2 Preface

This module contains review questions and answers keyed to the module titled Jb0230: Java OOP: Flow of Control¹.

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

List and describe eight of the statements used in Java programs to alter or control the logical flow of the program.

Answer 1 (p. 9)

3.2 Question 2

Provide pseudo-code that illustrates the general syntax of a **while** statement.

Answer 2 (p. 8)

*Version 1.2: Nov 25, 2012 11:37 am -0600

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

¹<http://cnx.org/content/m45196>

3.3 Question 3

True or false? During the execution of a **while** statement, the program will continue to execute the statement or compound statement for as long as the conditional expression evaluates to true, or until a **break** or **continue** statement is encountered. If false, explain why.

Answer 3 (p. 8)

3.4 Question 4

True or false? A **while** loop is an *entry condition* loop. If false, explain why.

Answer 4 (p. 8)

3.5 Question 5

What is the significance of an *entry condition* loop?

Answer 5 (p. 8)

3.6 Question 6

Provide pseudo-code illustrating the general syntax of the **if-else** statement.

Answer 6 (p. 8)

3.7 Question 7

Provide pseudo-code illustrating the general syntax of the **switch-case** statement.

Answer 7 (p. 7)

3.8 Question 8

Describe the behavior of a **switch-case** statement. Provide a pseudo-code fragment that illustrates your description of the behavior. Do not include a description of labeled break statements.

Answer 8 (p. 7)

3.9 Question 9

What are the three actions normally involved in the operation of a loop (*in addition to executing the code in the body of the loop*) ?

Answer 9 (p. 7)

3.10 Question 10

True or false? A **for** loop header consists of three clauses separated by colons. If false, explain why.

Answer 10 (p. 6)

3.11 Question 11

Provide pseudo-code illustrating the general syntax of a **for** loop

Answer 11 (p. 6)

3.12 Question 12

True or false? In a **for** loop, the first and third clauses within the parentheses can contain one or more expressions, separated by the comma operator. If False, explain why.

Answer 12 (p. 6)

3.13 Question 13

What is the guarantee made by the *comma operator* ?

Answer 13 (p. 6)

3.14 Question 14

True or false? The expressions within the first clause in the parentheses in a **for** loop are executed only once during each iteration of the loop. If false, explain why.

Answer 14 (p. 6)

3.15 Question 15

While any legal expression(s) may be contained in the first clause within the parentheses of a **for** loop, the first clause has a specific purpose. What is that purpose?

Answer 15 (p. 6)

3.16 Question 16

True or false? Variables can be declared and initialized within the first clause in the parentheses of a for loop. If false, explain why.

Answer 16 (p. 6)

3.17 Question 17

True or false? The second clause in the parentheses of a **for** loop consists of a single expression which must eventually evaluate to true to cause the loop to terminate. If false, explain why.

Answer 17 (p. 6)

3.18 Question 18

True or false? A **for** loop is an *exit condition* loop. If false, explain why.

Answer 18 (p. 5)

3.19 Question 19

True or false? Because a **for** loop is an *entry condition* loop, the third clause inside the parentheses is executed at the beginning of each iteration. If false, explain why.

Answer 19 (p. 5)

3.20 Question 20

True or false? A return statement is used to terminate a method and (*optionally*) return a value to the calling method. If False, explain why.

Answer 20 (p. 5)

3.21 Question 21

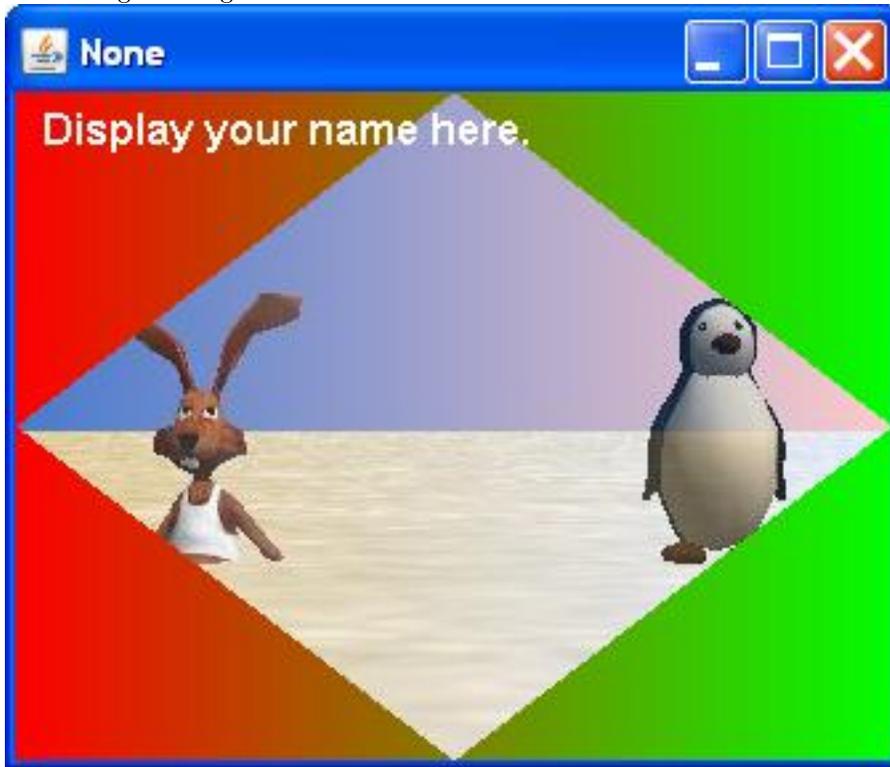
True or false? Exception handling modifies the flow of control of a Java program. If false, explain why.

Answer 21 (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 21

True.

[Back to Question 21 \(p. 3\)](#)

4.2 Answer 20

True.

[Back to Question 20 \(p. 3\)](#)

4.3 Answer 19

False. Although the third clause appears physically at the top of the loop, it isn't executed until the statements in the body of the loop have completed execution. This is an important point since this clause is typically used to update the control variable, and perhaps other variables as well. If variables are updated in the third clause and used in the body of the loop, it is important to understand that they do not get updated until the execution of the body is completed.

[Back to Question 19 \(p. 3\)](#)

4.4 Answer 18

False. The value of the second clause is tested when the statement first begins execution, and at the beginning of each iteration thereafter. Therefore, the **for** loop is an *entry condition* loop.

Back to Question 18 (p. 3)

4.5 Answer 17

False. The second clause consists of a single expression which must eventually evaluate to false (*not true*) to cause the loop to terminate.

Back to Question 17 (p. 3)

4.6 Answer 16

True.

Back to Question 16 (p. 3)

4.7 Answer 15

Typically the first clause is used for initialization. The intended purpose of the first clause is initialization.

Back to Question 15 (p. 3)

4.8 Answer 14

False. The expressions in the first clause are executed only once, at the beginning of the loop, regardless of the number of iterations.

Back to Question 14 (p. 3)

4.9 Answer 13

The *comma operator* guarantees that its left operand will be executed before its right operand.

Back to Question 13 (p. 3)

4.10 Answer 12

True.

Back to Question 12 (p. 2)

4.11 Answer 11

The general syntax of a **for** loop follows:

NOTE: **Syntax of a for loop**

```
for (first clause; second clause; third clause)
    single or compound statement
```

Back to Question 11 (p. 2)

4.12 Answer 10

False: A **for** loop header consists of three clauses separated by semicolons, not colons.

Back to Question 10 (p. 2)

4.13 Answer 9

The operation of a loop normally involves the following three actions in addition to executing the code in the body of the loop:

- Initialize a control variable.
- Test the control variable in a conditional expression.
- Update the control variable.

Back to Question 9 (p. 2)

4.14 Answer 8

The pseudo-code fragment follows:

NOTE: **Syntax of a switch-case statement**

```
switch(expression){
case constant:
sequence of optional statements
break; //optional
case constant:
sequence of optional statements
break; //optional
.
.
.
default //optional
sequence of optional statements
}
```

An expression is tested against a series of unique integer constants. If a match is found, the sequence of optional statements associated with the matching constant is executed. Execution of statements continues until an optional **break** is encountered. When **break** is encountered, execution of the **switch** statement is terminated and control is passed to the next statement following the **switch** statement.

If no match is found and the optional **default** keyword along with a sequence of optional statements has been provided, those statements will be executed.

Back to Question 8 (p. 2)

4.15 Answer 7

The general syntax of the **switch-case** statement follows:

NOTE: **Syntax of a switch-case statement**

```
switch(expression){
case constant:
sequence of optional statements
break; //optional
case constant:
sequence of optional statements
break; //optional
.
.
```

```
.  
.    
  default //optional  
  sequence of optional statements  
}
```

Back to Question 7 (p. 2)

4.16 Answer 6

The general syntax of the if-else statement is:

NOTE: **Syntax of an if-else statement**

```
  if(conditional expression)  
  statement or compound statement;  
  else //optional  
  statement or compound statement; //optional
```

Back to Question 6 (p. 2)

4.17 Answer 5

The significance of an *entry condition* loop is that the conditional expression is tested before the statements in the loop are executed. If it tests false initially, the statements in the loop will not be executed.

Back to Question 5 (p. 2)

4.18 Answer 4

True.

Back to Question 4 (p. 2)

4.19 Answer 3

True.

Back to Question 3 (p. 2)

4.20 Answer 2

The general syntax of a **while** statement follows :

NOTE: **Syntax of a while statement**

```
  while (conditional expression)  
  statement or compound statement;
```

Back to Question 2 (p. 1)

4.21 Answer 1

The following table lists the statements supported by Java for controlling the logical flow of the program.

NOTE: **Flow of control statements**

Statement	Type	if-else selection
switch-case		selection
for		loop
for-each		loop
while		loop
do-while		loop
try-catch-finally		exception handling
throw		exception handling
break		miscellaneous
continue		miscellaneous
label:		miscellaneous
return		miscellaneous
goto		reserved by Java but not supported

Back to Question 1 (p. 1)

5 Miscellaneous

This section contains a variety of miscellaneous information.

NOTE: **Housekeeping material**

- Module name: Jb0230r Review: Flow of Control
- File: Jb0230r.htm
- Originally published: 1997
- Published at cnx.org: November 25, 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-