

JB0170: JAVA OOP: A LITTLE MORE INFORMATION ABOUT CLASSES.*

R.G. (Dick) Baldwin

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

Abstract

Class definitions form the foundation for Java OOP. They are discussed in increasing detail in subsequent modules. This module sheds just a little more light on classes.

1 Table of Contents

- Preface (p. 1)
- Listings (p. 1)
- Introduction (p. 1)
- Defining a class in Java (p. 2)
- Miscellaneous (p. 2)

2 Preface

This module is part of a sub-collection of modules designed to help you learn to program computers. This module sheds a little more light on the Java construct called a *class* .

3 Listings

- Listing 1 (p. 2) . General syntax for defining a Java class.

4 Introduction

New types

Java makes extensive use of classes. When a class is defined in Java, a new *type* comes into being. The new type definition can then be used to instantiate (*create instances of*) one or more objects of that new type.

A blueprint

The class definition provides a *blueprint* that describes the *data* contained within, and the *behavior* of objects instantiated according to the new type.

*Version 1.1: Nov 17, 2012 7:35 pm -0600

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

The data

The data is contained in variables defined within the class (*often called member variables, data members, attributes, fields, properties, etc.*).

The behavior

The behavior is controlled by methods defined within the class.

State and behavior

An object is said to have *state* and *behavior* . At any instant in time, the *state* of an object is determined by the values stored in its *variables* and its behavior is determined by its *methods* .

Class vs. instance

It is possible to define:

- instance variables and instance methods
- static or *class* variables and static or *class* methods.

Instance variables and instance methods can only be accessed through an object instantiated from the class. They belong to the individual objects, (*which is why they are called instance variables and instance methods*) .

Class variables and *class* methods can be accessed without first instantiating an object. They are shared among all of the objects instantiated from the class and are even accessible in the total absence of an object of the class.

The class name alone is sufficient for accessing *class* variables and *class* methods by joining the name of the class to the name of the variable or method using a period.

5 Defining a class in Java

The general syntax for defining a class in Java is shown in Listing 1 (p. 2) .

Listing 1: General syntax for defining a Java class.

```
class MyClassName{
    . . .
} //End of class definition.
```

This syntax defines a class and creates a new type named **MyClassName** .

The definitions of variables, methods, constructors, and a variety of other members are inserted between the opening and closing curly brackets.

6 Miscellaneous

This section contains a variety of miscellaneous information.

NOTE: Housekeeping material

- Module name: Jb0170: Java OOP: A little more information about classes.
- File: Jb0170.htm
- Originally published: 1997
- Published at cnx.org: November 17, 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-