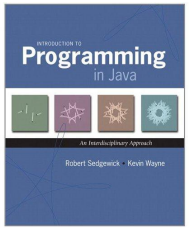
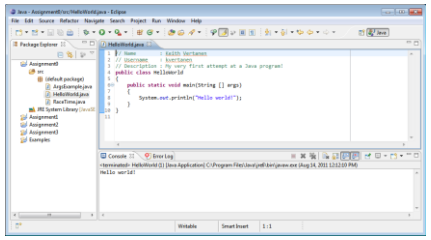
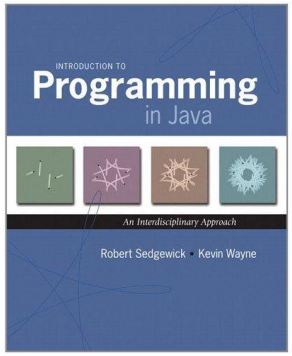
Fundamentals of Computer Science



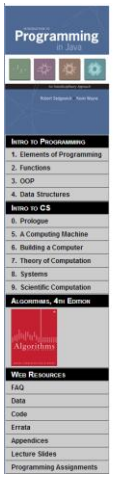


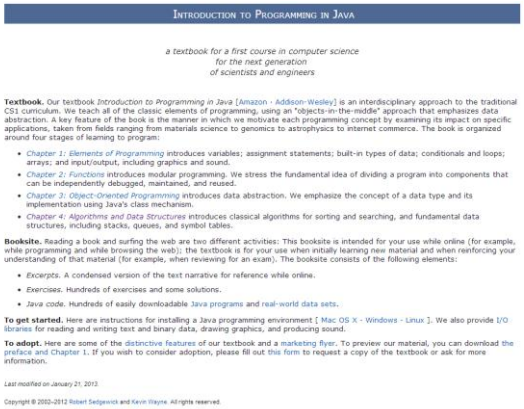
Keith Vertanen

The Book



The Booksite





INTRODUCTION TO PROGRAMMING IN JAVA

a textbook for a first course in computer science
for the next generation
of scientists and engineers

Textbook. Our textbook *Introduction to Programming in Java* [Amazon · Addison-Wesley] is an interdisciplinary approach to the traditional CS1 curriculum. We teach all of the classic elements of programming, using an "objects-in-the-middle" approach that emphasizes data abstraction. A key feature of the book is the manner in which we motivate each programming concept by examining its impact on specific applications, taken from fields ranging from materials science to genomics to astrophysics to internet commerce. The book is organized around four stages of learning to program:

- *Chapter 1: Elements of Programming* introduces variables; assignment statements; built-in types of data; conditionals and loops; arrays; and input/output, including graphics and sound.
- *Chapter 2: Functions* introduces modular programming. We stress the fundamental idea of dividing a program into components that can be independently debugged, maintained, and reused.
- *Chapter 3: Object-Oriented Programming* introduces data abstraction. We emphasize the concept of a data type and its implementation using Java's class mechanism.
- *Chapter 4: Algorithms and Data Structures* introduces classical algorithms for sorting and searching, and fundamental data structures, including stacks, queues, and symbol tables.

Booksite. Reading a book and surfing the web are two different activities: This booksite is intended for your use while online (for example, while programming and while browsing the web); the textbook is for your use when initially learning new material and when reinforcing your understanding of that material (for example, when reviewing for an exam). The booksite consists of the following elements:

- *Excerpts.* A condensed version of the text narrative for reference while online.
- *Exercises.* Hundreds of exercises and some solutions.
- *Java code.* Hundreds of easily downloadable Java programs and real-world data sets.

To get started. Here are instructions for installing a Java programming environment [Mac OS X - Windows - Linux]. We also provide I/O libraries for reading and writing text and binary data, drawing graphics, and producing sound.

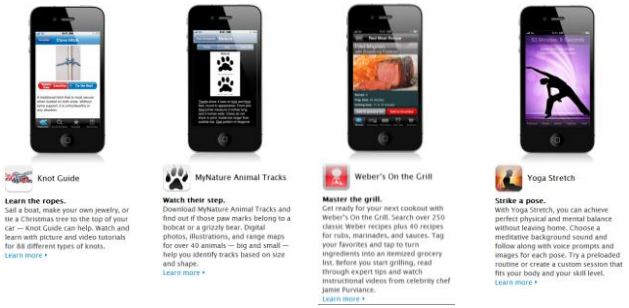
To adopt. Here are some of the distinctive features of our textbook and a marketing flyer. To preview our material, you can download the preface and Chapter 1. If you wish to consider adoption, please fill out this form to request a copy of the textbook or ask for more information.

Last modified on January 21, 2012.

http://introcs.cs.princeton.edu/java/home/

Why learn to program?

Lots of existing software:



…and 499,996 more and that's just iPhone apps

Reasons to program

Well…

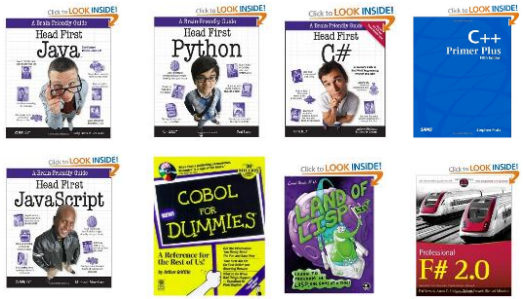
1. Someone had to program all those iPhone apps (and rake in the sweet sweet profits)
2. Many problems are so specific to your company/problem nobody has an app for that
3. Programming is fun, creative and a challenge.
4. Enables you to make your computer do (almost) anything you want.

Languages

- Machine language
  - Low level, what the hardware understands
  - Tedious and error-prone to write
  - Specific to a particular type of computer
- Natural language
  - Imprecise and ambiguous
  - Hard to convert to instructions for the hardware
- High level programming language
  - Good balance between the two extremes

Becoming a programmer: step 1

Choose a language…



and hundreds more…

Our choice: Java

- Advantages
  - Widely used
  - Freely available
  - Features help novices learn to program
- No perfect single language
  - You'll learn many other languages
    - C/C++, assembly, Python, C#, JavaScript, PHP...
  - Programming skills translate easily between them

James Gosling, father of Java.

"There are only two kinds of languages: the ones people complain about and the ones nobody uses."
-Bjarne Stroustrup, father of C++

Your first program

http://www.zazzle.com/baby_girls_first_java_program_hello_world_tshirt-235063563751392326 $23.95

How Java works

Source code:
Plain text file created in some editor (notepad, vi, TextEdit, Eclipse, DrJava, …)

```
public class HelloWorld
{
   public static void main(String [] args)
   {
      System.out.println("Hello world!");
   }
}
```

HelloWorld.java

% javac HelloWorld.java

Java bytecode:
Intermediate language that any device running Java can understand (humans generally ignore this)

```
Compiled from "HelloWorld.java"
public class HelloWorld extends java.lang.Object{
public HelloWorld();
  Code:
   0:   aload_0
   1:   invokespecial   #1; //Method java/lang/Object."<init>":()V
   4:   return
public static void main(java.lang.String[]);
  Code:
   0:   getstatic       #2; //Field
java/lang/System.out:Ljava/io/PrintStream;
   3:   ldc             #3; //String Hello world!
   5:   invokevirtual   #4; //Method
java/io/PrintStream.println:(Ljava/lang/String;)V
   8:   return
}
```

HelloWorld.class

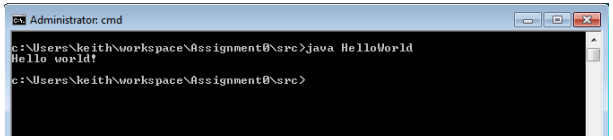
How Java works

Java bytecode:
Intermediate language that any device running Java can understand (humans generally ignore this)

```
Compiled from "HelloWorld.java"
public class HelloWorld extends java.lang.Object{
public HelloWorld();
  Code:
   0:   aload_0
   1:   invokespecial   #1; //Method java/lang/Object."<init>":()V
   4:   return
public static void main(java.lang.String[]);
  Code:
   0:   getstatic       #2; //Field
java/lang/System.out:Ljava/io/PrintStream;
   3:   ldc             #3; //String Hello world!
   5:   invokevirtual   #4; //Method
java/io/PrintStream.println:(Ljava/lang/String;)V
   8:   return
}
```

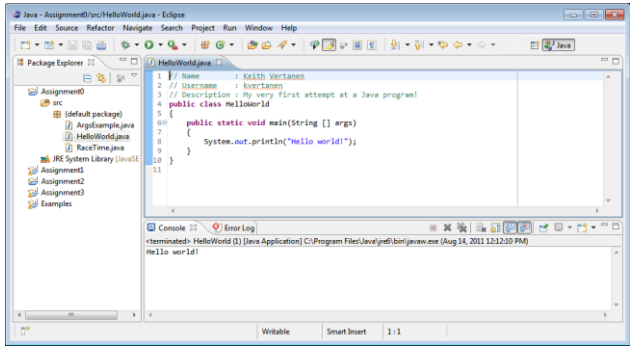
HelloWorld.class

% java HelloWorld



Eclipse

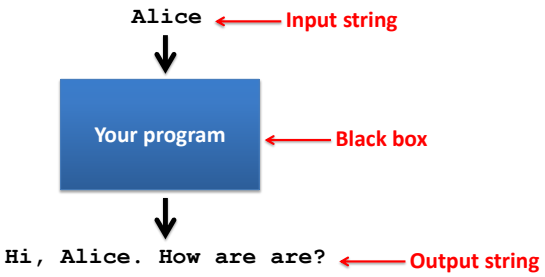
- Eclipse IDE (integrated development environment)



Eclipse

- Eclipse IDE (integrated development environment)
  - Recommended but not required
  - Free
  - Helpful features:
    - Syntax highlighting
    - Flagging likely mistakes
  - We will use mostly as a text editor
    - Ignoring 95% of its features
  - How to install?
    - See course web site, resources page
  - We'll still learn to work on the command line

View of programming from 10,000 feet



Anatomy of a Java program

Name of the class, must be in file named `CostCalc.java` (case sensitive!)

```
public class CostCalc
{
   public static void main(String [] args)
   {

   }
}
```

All the action goes here (for now)

Extra things from the command line
Allows program's **output** to depend on its **input**

```
% java CostCalc bananas 12 0.21
To buy 12 bananas you will need $2.52
```

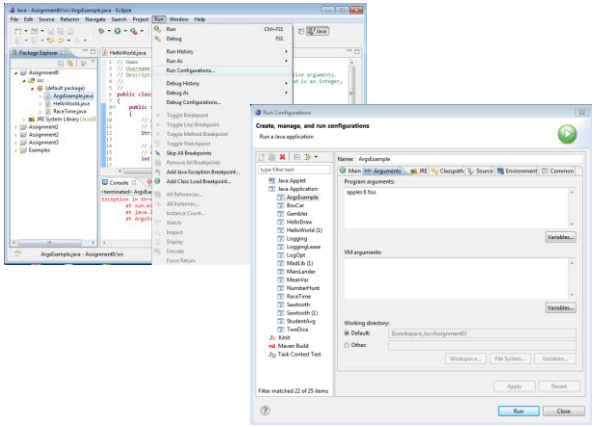

args array

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

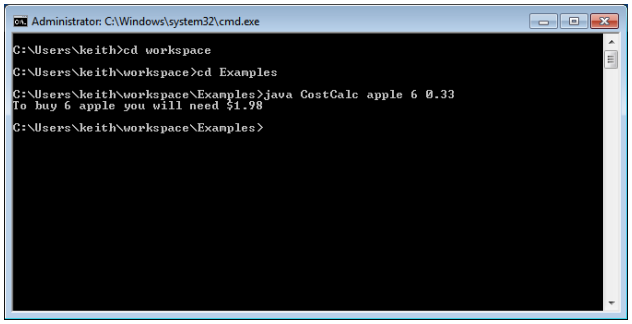
```
% java CostCalc bananas 12 0.21
To buy 12 bananas you will need $2.52
```

identifier	meaning	value	type
`args[0]`	1st thing on command line after Java class name	`"bananas"`	`String`
`args[1]`	2nd thing on command line	`"12"`	`String`
`args[2]`	3rd thing on command line after Java class	`"0.21"`	`String`
`args.length`	# of things on command line	`3`	`int`

Command line args in Eclipse



Command line args in command shell



Summary

- Source code → byte code → program output
- Wrote our first program
  - Hello world!
- Program have input
  - e.g. arguments passed into `main()`
- Programs have output
  - e.g. text printed from `System.out.println()`