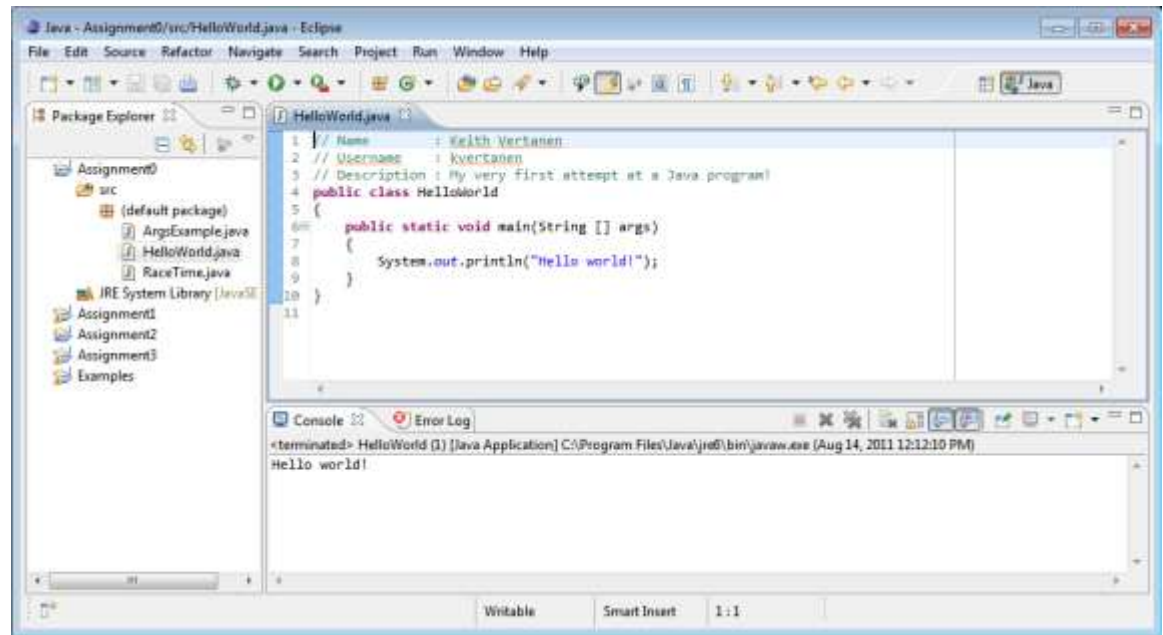
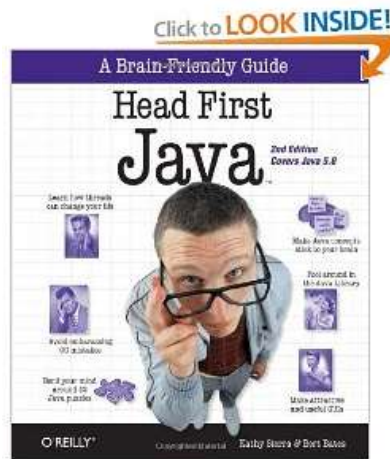


CSCI 135: Fundamentals of Computer Science I Fall 2011



```
1 // Name : Keith Vertanen
2 // Username : kvertanen
3 // Description : My very first attempt at a Java program!
4 public class HelloWorld
5 {
6 //
7 public static void main(String [] args)
8 {
9 System.out.println("Hello world!");
10 }
11 }
```

<terminated> HelloWorld [1] [Java Application] C:\Program Files\Java\jdk6\bin\javaw.exe (Aug 14, 2011 12:12:10 PM)
Hello world!

Keith Vertanen
Museum 102
496-4385
kvertanen@mtech.edu

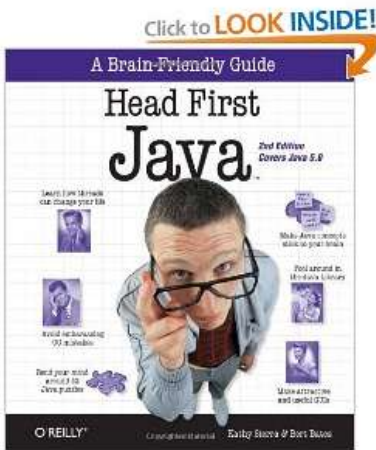


Wouldn't it be dreamy
if there was a Java book
that was more stimulating
than waiting in line at the
DMV to renew your driver's
license? It's probably just a
fantasy...

The Book



The Book



Head First Java, 2nd Edition [Paperback]

[Kathy Sierra](#) (Author), [Bert Bates](#) (Author), [Bert Bates](#) (Author)

★★★★★ (320 customer reviews) | Like (17)

List Price: ~~\$44.95~~

Price: **\$24.72** & eligible for **FREE Super Saver Shipping** on orders over \$25. [Details](#)

You Save: **\$20.23 (45%)**

In Stock.

Ships from and sold by **Amazon.com**. Gift-wrap available.

Want it delivered Tuesday, August 16? Order it in the next **49 hours and 0 minutes**, and choose **One-Day Shipping** at checkout. [Details](#)

[51 new](#) from \$23.99 [72 used](#) from \$19.99

Formats

Amazon Price: New from Used from

<input type="checkbox"/> Paperback	\$24.72	\$23.99	\$19.99
------------------------------------	---------	---------	---------

[Show 2 more formats](#)

Course web site

<http://katie.mtech.edu/classes/csci135>

Moodle for grades and submitting programs.

Why learn to program?

Lots of existing software:



Knot Guide

Learn the ropes.

Sail a boat, make your own jewelry, or tie a Christmas tree to the top of your car — Knot Guide can help. Watch and learn with picture and video tutorials for 88 different types of knots.

[Learn more](#) ▶



MyNature Animal Tracks

Watch their step.

Download MyNature Animal Tracks and find out if those paw marks belong to a bobcat or a grizzly bear. Digital photos, illustrations, and range maps for over 40 animals — big and small — help you identify tracks based on size and shape.

[Learn more](#) ▶



Weber's On the Grill

Master the grill.

Get ready for your next cookout with Weber's On the Grill. Search over 250 classic Weber recipes plus 40 recipes for rubs, marinades, and sauces. Tag your favorites and tap to turn ingredients into an itemized grocery list. Before you start grilling, read through expert tips and watch instructional videos from celebrity chef Jamie Purviance.

[Learn more](#) ▶



Yoga Stretch

Strike a pose.

With Yoga Stretch, you can achieve perfect physical and mental balance without leaving home. Choose a meditative background sound and follow along with voice prompts and images for each pose. Try a preloaded routine or create a custom session that fits your body and your skill level.

[Learn more](#) ▶

..and 425,996 more, and that's just iPhone apps

Why learn to program?

Well...

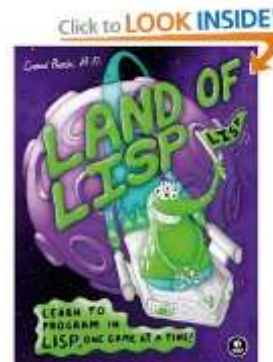
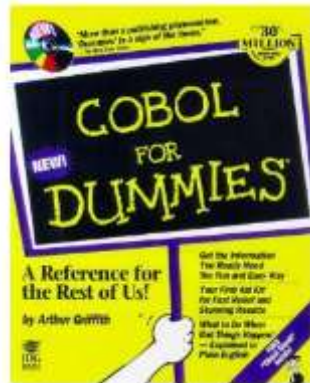
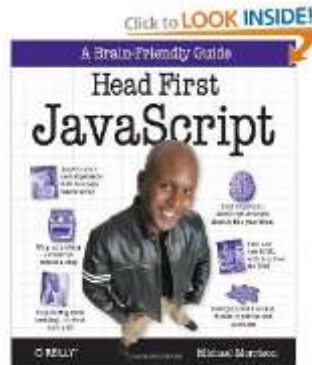
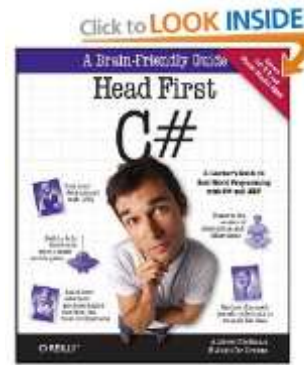
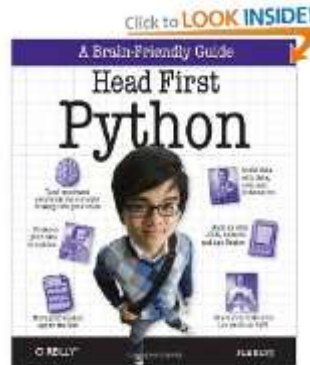
- 1 Someone had to program all those iPhone apps (and rake in the sweet sweet profits)
- 2 Many problems are so specific to your company/problem nobody has an app for that
- 3 Programming is fun, creative and a challenge.
- 4 Enables you to make your computer do (almost) anything you want.

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 allowing novices to focus on learning to program



James Gosling, father of Java.

- No perfect single language

- You'll learn many other languages
- Programming skills translate easily between them



“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

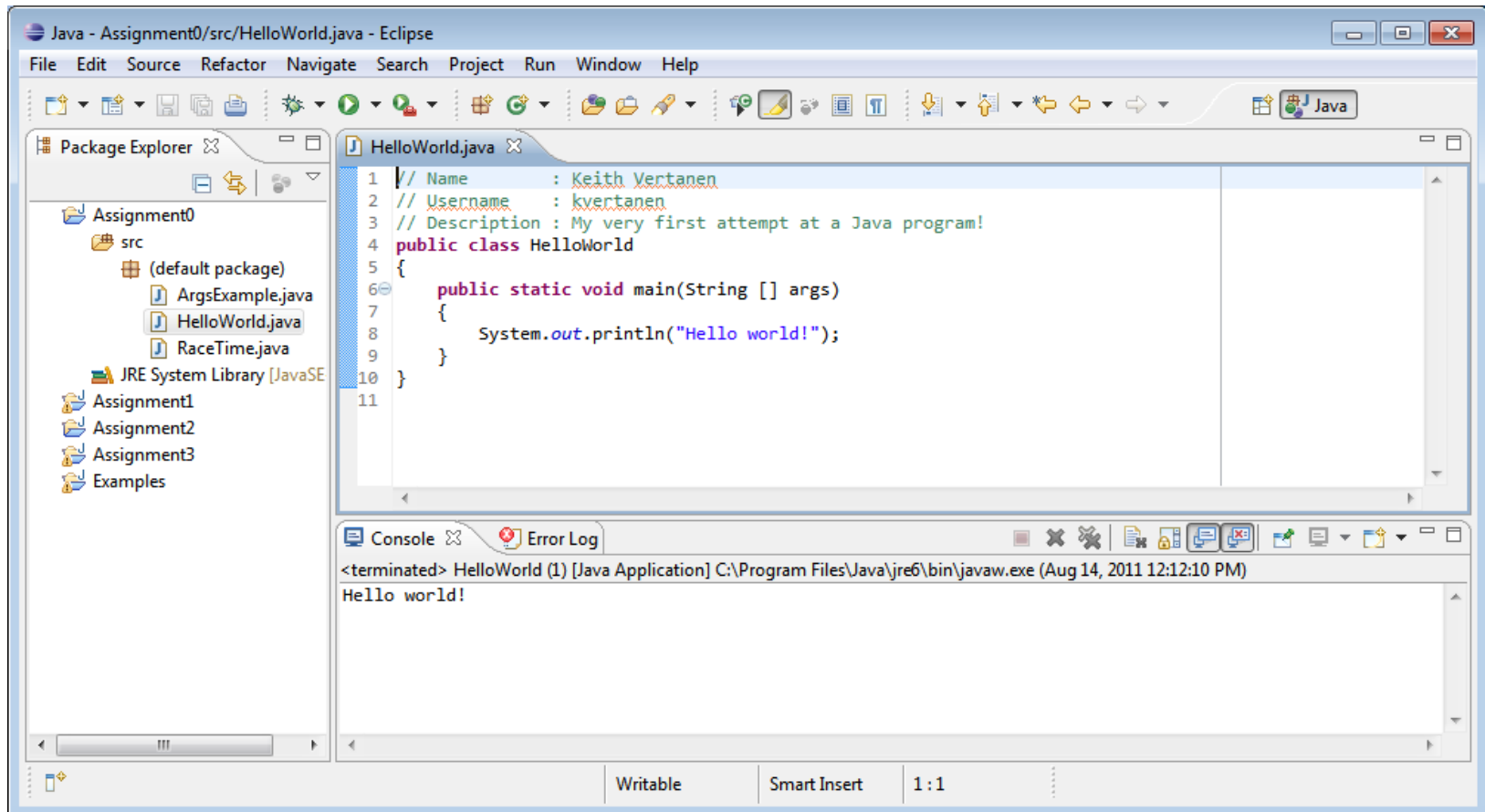
% java HelloWorld



```
Administrator: cmd
c:\Users\keith\workspace\Assignment0\src>java HelloWorld
Hello world!
c:\Users\keith\workspace\Assignment0\src>
```

Eclipse

- Eclipse IDE (integrated development environment)



Eclipse

- **Eclipse IDE** (integrated development environment)
 - Recommended but not required
 - Free to download
 - Helpful syntax highlighting and other features
 - You still need to learn to run Java programs on the command line
 - We will mostly use as an editor ignoring 95% of its features
 - How to install? See course web site

Variables and data types

- Variables

- Stores information your program needs
- Each has a unique name
- Each has a specific type

Java built-in type	what it stores	example values	operations
int	integer values	42 1234	add, subtract, multiply, divide, remainder
double	floating-point values	9.95 3.0e8	add, subtract, multiply, divide
boolean	truth values	true false	and, or, not
char	characters	'a', 'b', '!'	compare
String	sequence of characters	"Hello world!" "I love this!"	concatenate

Some definitions

Declaration statement

“I’m going to need an integer and let’s call it a”

```
int a;
```

Variable name

“Whenever I say a, I mean the value stored in a”

```
a = 10;
```

Literal

“I want the value 10”

```
int b;
```

Assignment statement

“Variable b gets the literal value 7”

```
b = 7;
```

Combined declaration and assignment

“Make me an integer variable called c and assign it the value obtained by adding together a and b”

```
int c = a + b;
```

Text

- **String** data type
 - A sequence of characters
 - Double quote around the characters
 - Concatenation using the + operator

```
String firstName = "Keith";  
String lastName = "Vertanen";  
String fullName = firstName + " " + lastName;  
String favNumber = "42";  
  
System.out.println(fullName +  
                    "'s favorite number is " +  
                    favNumber);
```

```
Keith Vertanen's favorite number is 42
```

Integers

- **int** data type

- An integer value between -2^{31} and $+2^{31}-1$

- Operations:

add	subtract	multiply	divide	remainder
+	-	*	/	%

example	result	comment
10 + 7	17	
10 - 7	3	
10 * 7	70	
10 / 7	1	integer division, no fractional part
10 % 7	3	remainder after dividing by 7
10 / 0		runtime error, you can't divide an integer by 0!

Integers

- **int** data type
 - Normal rules of math precedence
 - Use ()'s to force order of calculation

example	result	comment
$10 + 7 * 2$	24	multiplication comes before addition
$(10 + 7) * 2$	34	()'s force addition to occur first
$10 / 7 + 2$	3	integer division result is 1 which is added to 2
$10 - 7 - 2$	1	
$(10 - 7) - 2$	1	
$10 - (7 - 2)$	5	

Floating-point numbers

- **double** data type

- Floating-point number (as specified by IEEE 754)

Operations:

add	subtract	multiply	divide
+	-	*	/

example	result
9.95 + 2.99	12.94
1.0 - 2.0	-1.0
1.0 / 2.0	0.5
1.0 / 3.0	0.3333333333333333
1.0 / 0.0	Infinity
0.0 / 123.45	0.0
0.0 / 0.0	NaN

Type conversion

- Java is **strongly typed**
 - Helps protect you from bugs

```
public class TypeExample0
{
    public static void main(String [] args)
    {
        int orderTotal;
        double costItem = 29.95;
        orderTotal = costItem * 1.06;
        System.out.println("total=" + orderTotal);
    }
}
```

```
% javac TypeExample0.java
```

```
TypeExample0.java:7: possible loss of precision
found    : double
```

```
required: int
```

```
    orderTotal = costItem * 1.06;
```

```
        ^
```

Type conversion

- Convert from one type to another
 - Manually → using a cast
 - Casting to `int` drops fractional part

```
public class TypeExample1
{
    public static void main(String [] args)
    {
        int orderTotal;
        double costItem = 29.95;
        orderTotal = (int) (costItem * 1.06);
        System.out.println("total=" + orderTotal);
    }
}
```

```
% java TypeExample1
total=31
```

Type conversion

- Automatic conversion
 - `String` concatenation converts numeric types
 - If **no loss of precision** → automatic promotion

```
public class TypeExample2
{
    public static void main(String [] args)
    {
        double orderTotal;
        int costItem = 30;
        orderTotal = costItem * 1.06;
        System.out.println("total=" + orderTotal);
    }
}
```

```
% java TypeExample2
total=31.8
```


Summary

- Source code → byte code → program output
- Wrote our first program, **Hello world!**
- Variables **store and compute on data**
 - `String, int, double`
- Java is **strongly typed**
 - Conversion between types sometimes needed

First lab

- Wednesday 3:00 – 5:50pm, Main 205 (maybe, if not then in Museum lab)
- Goals
 - Start work on assignment #0, due next Wednesday!
 - Help installing/using Java and Eclipse
 - Bring your own laptop if you want
- Read through assignment before lab