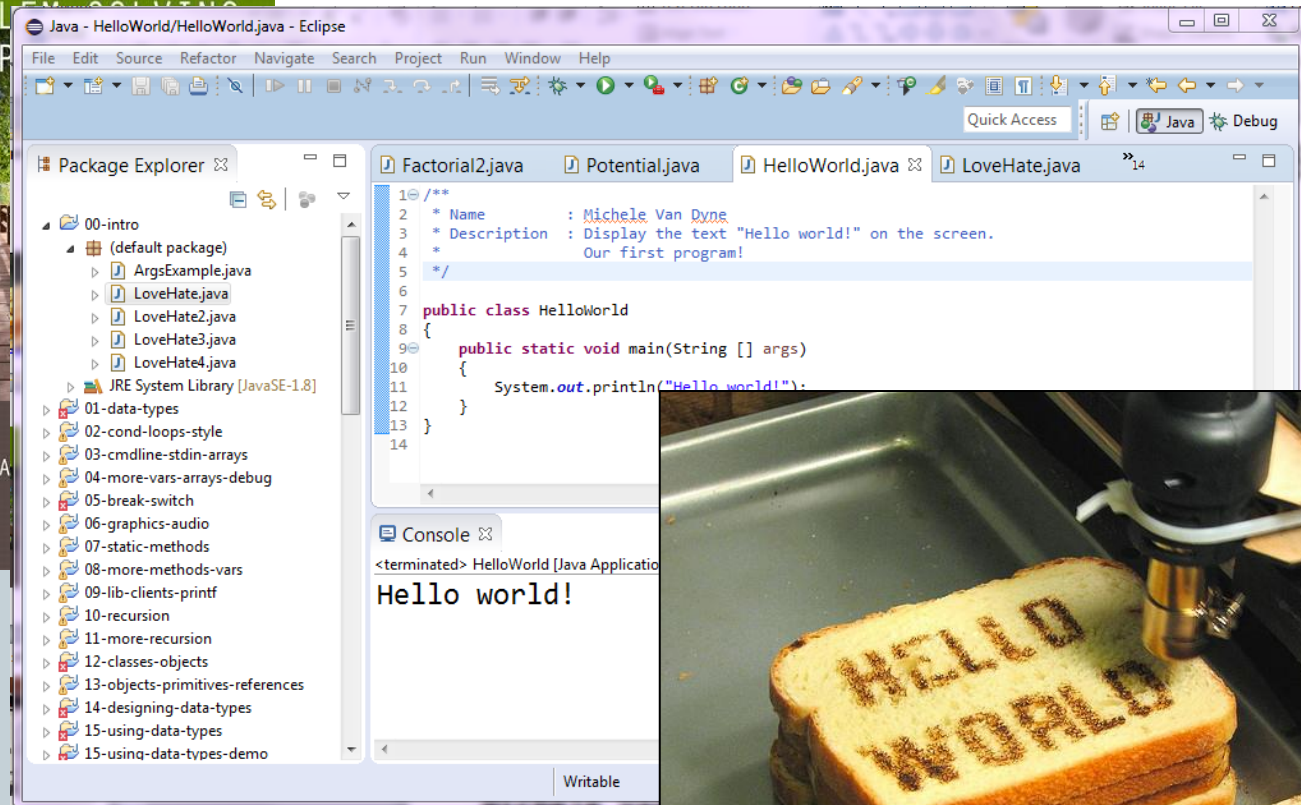
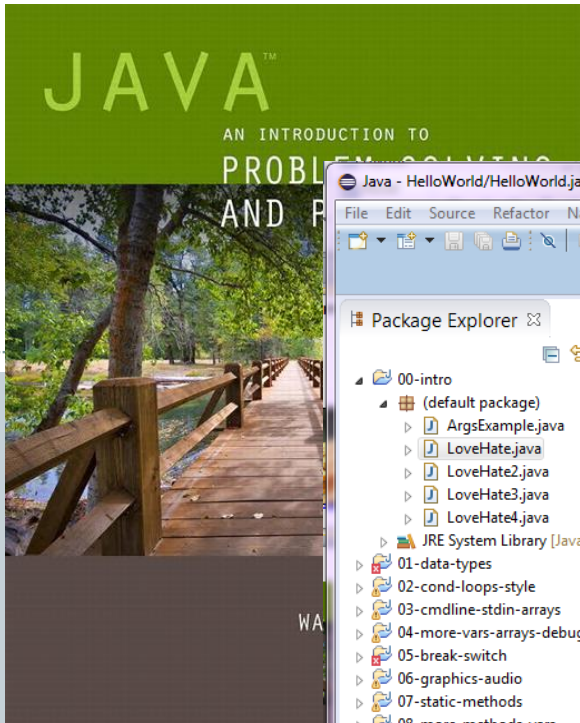


Introduction



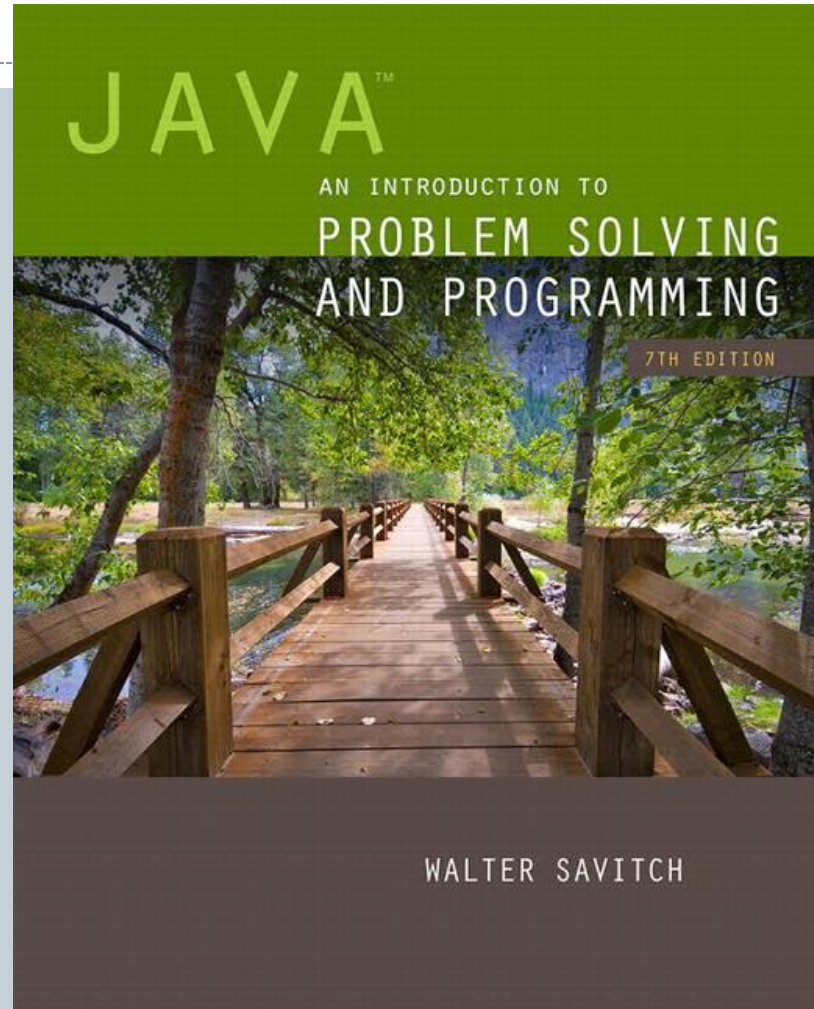
<http://www.flickr.com/photos/oskay/472097903/>

Michele Van Dyne
Museum 204B
mvandyne@mtech.edu

Outline

- Course Overview
- Computer Basics
- Programs and Languages
- Our First Program
 - Java Compilation
 - Eclipse IDE
 - Anatomy of a Java Program
 - Command Line Arguments
- Algorithms

The Book



JAVA™

AN INTRODUCTION TO
PROBLEM SOLVING
AND PROGRAMMING

WALTER SAVITCH

7TH EDITION

Home



Welcome to the companion website to accompany *Java: An Introduction to Problem Solving and Programming, 7th Edition*, by Walter Savitch.

Please select from the following links:

Student Resources:

- [VideoNotes](#)
- [Source Codes](#)
- [Chapter 13: Window Interfaces Using Swing](#)
- [Chapter 14: Applets and HTML](#)
- [Chapter 15: More Swing](#)
- [Appendix 9: The Iterator Interface](#)
- [Appendix 10: Cloning](#)
- [Appendix 11: Java Reserved](#)
- [Keywords](#)

Instructor Resources:

[Instructor Resources:](#) Instructor materials are available on our Instructor Resource Center. Separate registration is required.

Course Staff and Web Site



Michele



We will have teaching assistants also – I'll announce these when we know who.

<http://cs.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 499,996 more and that's just iPhone apps

Reasons to Program

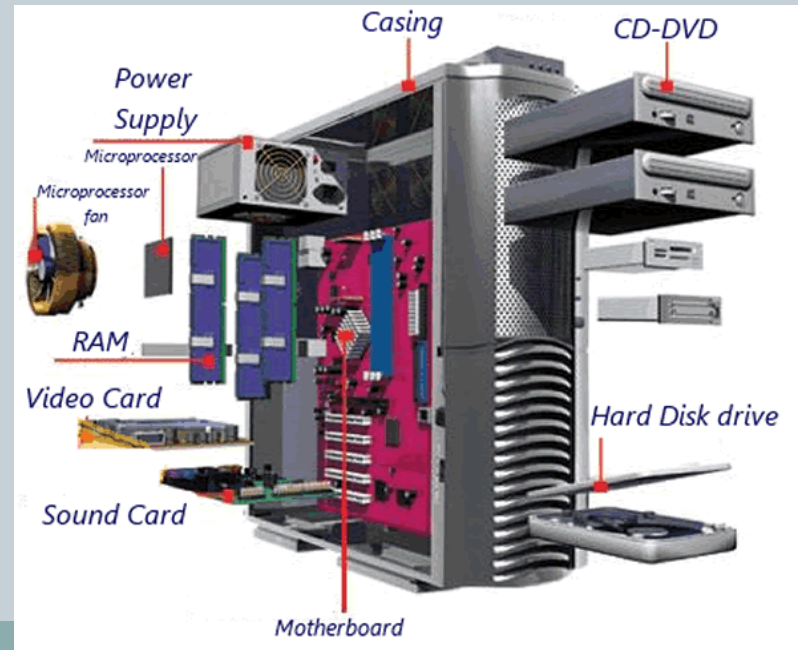
Well...

- 1 Someone had to program all those iPhone apps.
(and rake in the sweet sweet profits)
- 2 Many problems are very 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.



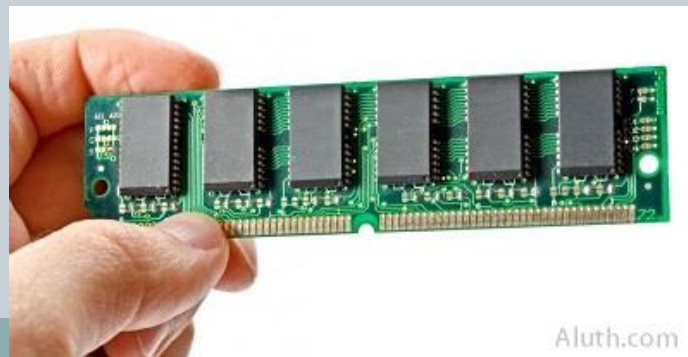
Hardware and Memory

- Most modern computers have similar components including
 - Input devices (keyboard, mouse, etc.)
 - Output devices (display screen, printer, etc.)
 - A processor
 - Two kinds of memory (main memory and auxiliary memory).



Main memory

- Working memory used to store
 - The current program
 - The data the program is using
 - The results of intermediate calculations
- Usually measured in megabytes or gigabytes (e.g. 8 gigabytes of RAM)
 - RAM is short for *random access memory*
 - A *byte* is a quantity of memory



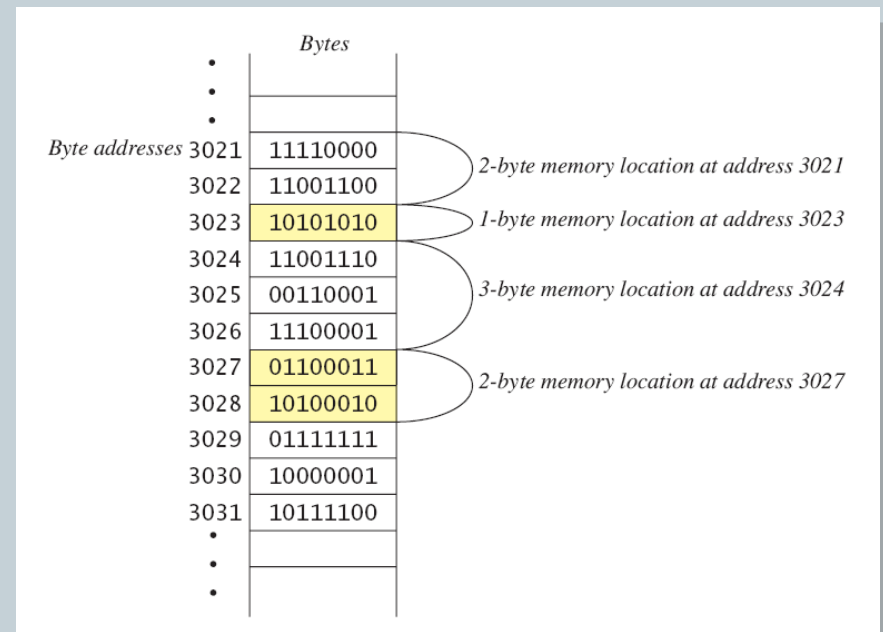
Bits, Bytes, and Addresses

- A *bit* is a digit with a value of either 0 or 1.
- A *byte* consists of 8 bits.
- Each byte in main memory resides at a numbered location called its *address*.



Main Memory

- Data of all kinds (numbers, letters, strings of characters, audio, video, even programs) are encoded and stored using 1s and 0s.
- When more than a single byte is needed, several adjacent bytes are used.
 - The address of the first byte is the address of the unit of bytes.



Programs

- A *program* is a set of instructions for a computer to follow.
- We use programs almost daily (email, word processors, video games, bank ATMs, etc.).
- When the computer follows the instructions it is *running* or *executing* the program.

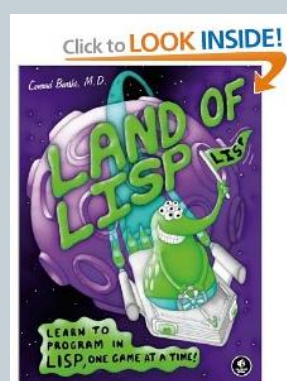
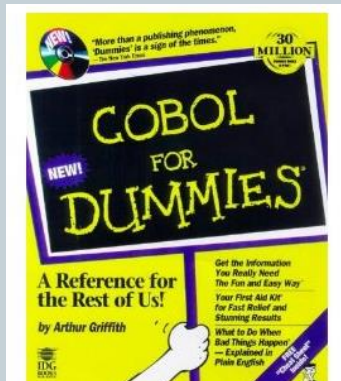
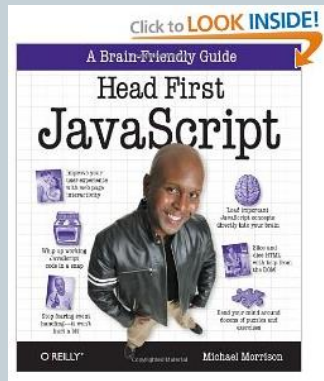
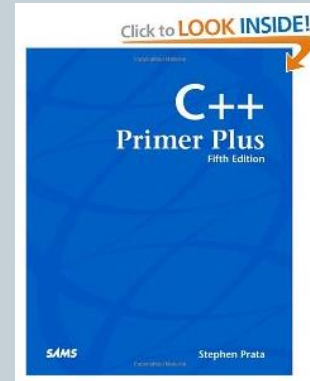
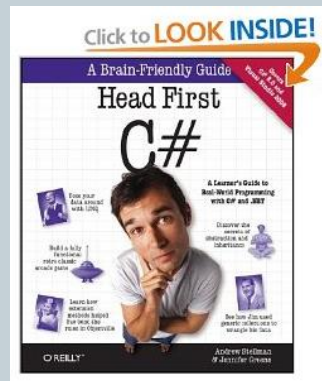
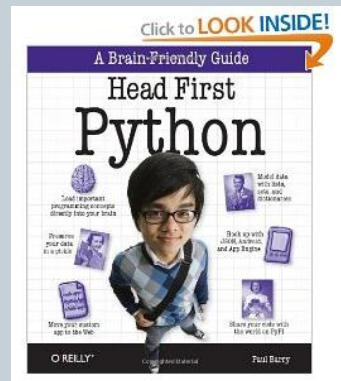
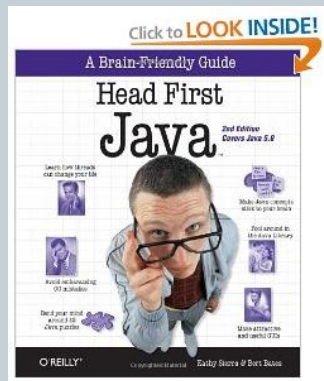


Languages

- 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...

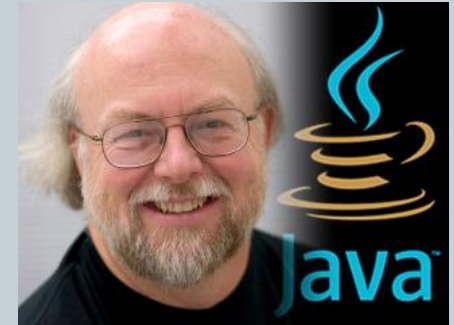
Our Choice: Java

- Advantages

- Widely used, modern
- Freely available, cross-platform
- 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

“compiling”

% 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

“running”

% java HelloWorld

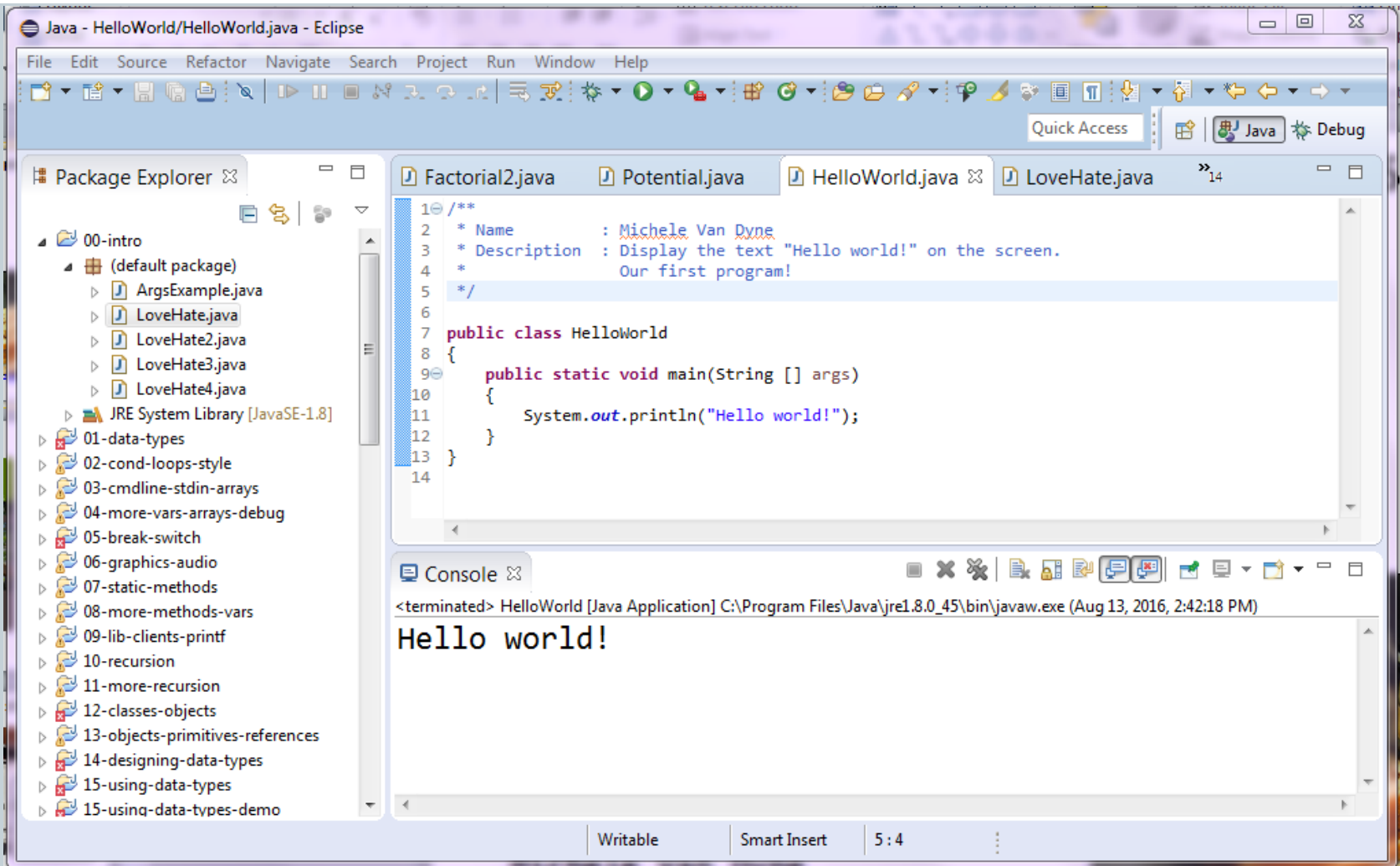


```
Command Prompt
C:\Users\Mich\Desktop\CSCI 135 Fall 2016\Workspace\HelloWorld>java HelloWorld
Hello world!
C:\Users\Mich\Desktop\CSCI 135 Fall 2016\Workspace\HelloWorld>
```

OUR FIRST PROGRAM

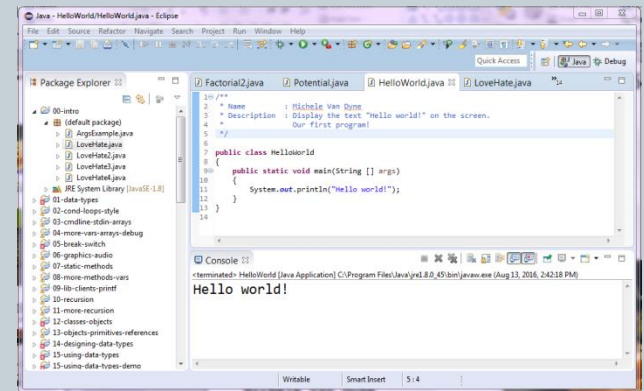
Eclipse

- Eclipse IDE (Integrated Development Environment)

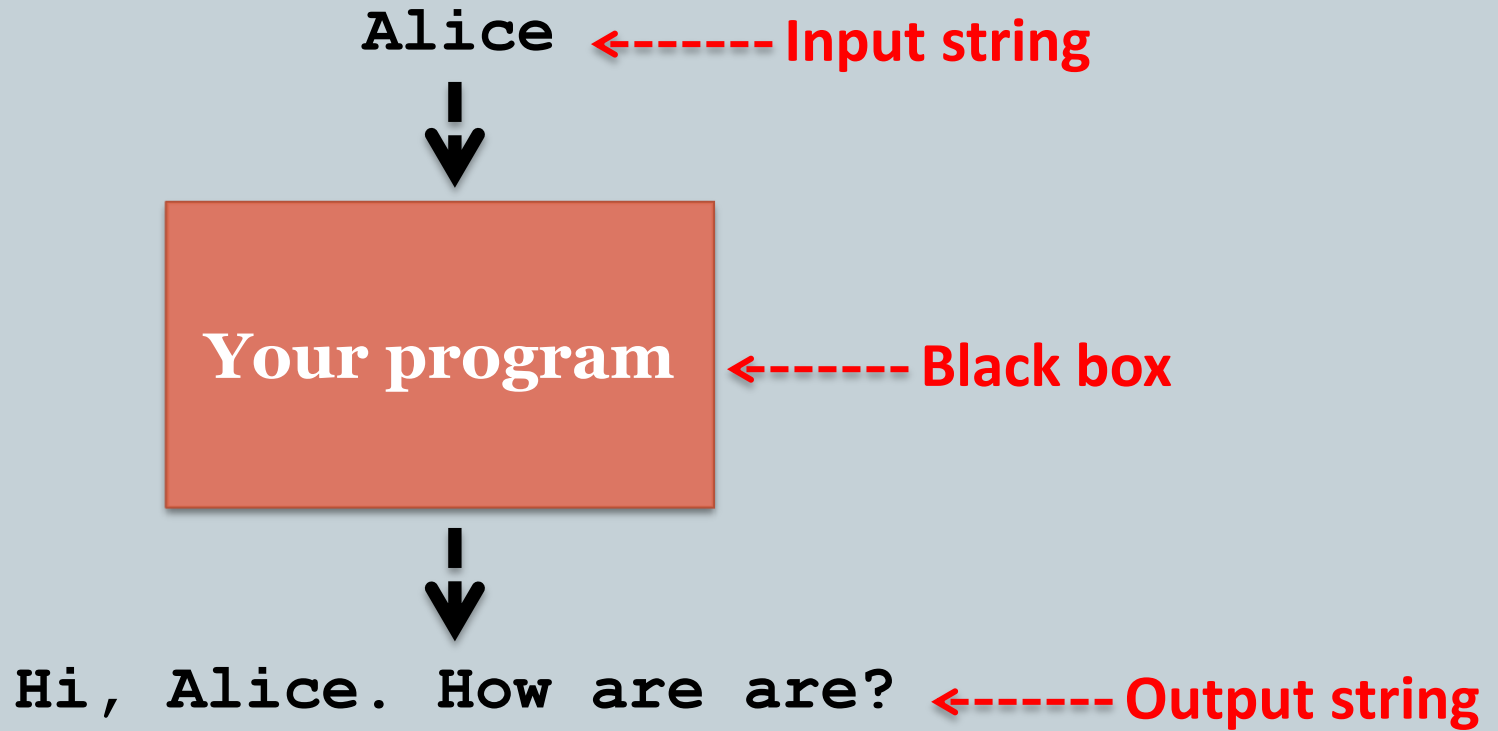


Eclipse

- **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

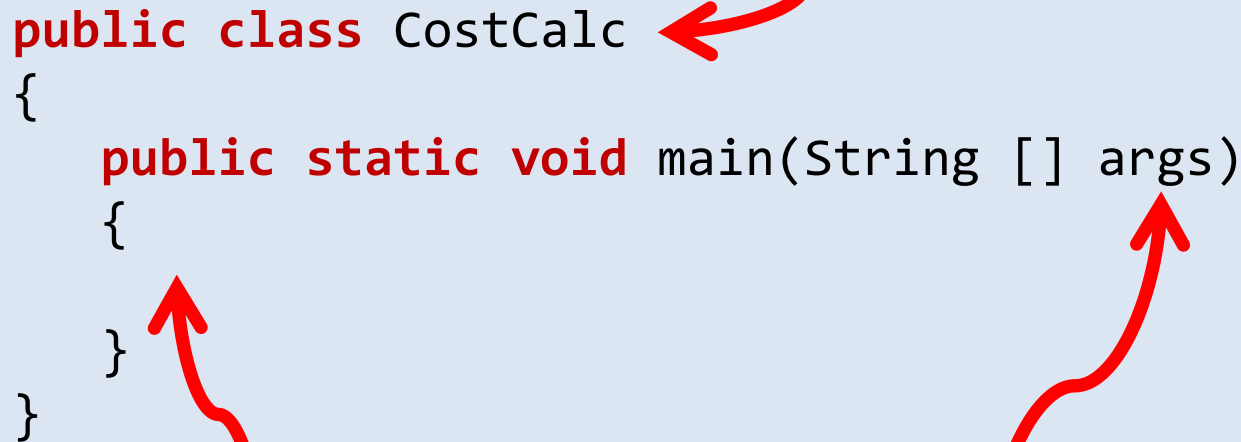


OUR FIRST PROGRAM

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

The **input** that allows the program to produce variable **output**

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

Compiling and Running

- A Java program can involve any number of classes.
- The class to run will contain the words:

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

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

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

OUR FIRST PROGRAM

Command line args in Eclipse

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project named 'Assignment0' with a source folder 'src' containing files 'ArgsExample.java', 'HelloWorld.java', and 'RaceTime.java'. The main editor shows the code for 'HelloWorld.java', which includes a 'main' method. The Run menu is open, showing options like 'Run', 'Debug', 'Run History', 'Run As', 'Run Configurations...', 'Debug History', 'Debug As', and 'Debug Configurations...'. The 'Run Configurations' dialog is open, showing a list of configurations. The 'Main' configuration is selected, and the 'Program arguments' field contains 'apples 6 foo'. The 'Working directory' is set to the default workspace location.

Java - Assignment0/src/ArgsExample.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- Assignment0
 - src
 - (default package)
 - ArgsExample.java
 - HelloWorld.java
 - RaceTime.java

Console

```
<terminated> ArgsExample
Exception in thread "main"
at sun.reflect.NativeMethodAccessorImpl.invoke0(NativeMethodAccessorImpl.java:62)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at ArgsExample.main(ArgsExample.java:17)
```

Run Configurations

Create, manage, and run configurations

Run a Java application

Name: ArgsExample

Main Arguments

Program arguments:

apples 6 foo

VM arguments:

Working directory:

Default: \${workspace_loc:Assignment0}

Other:

Workspace... File System... Variables...

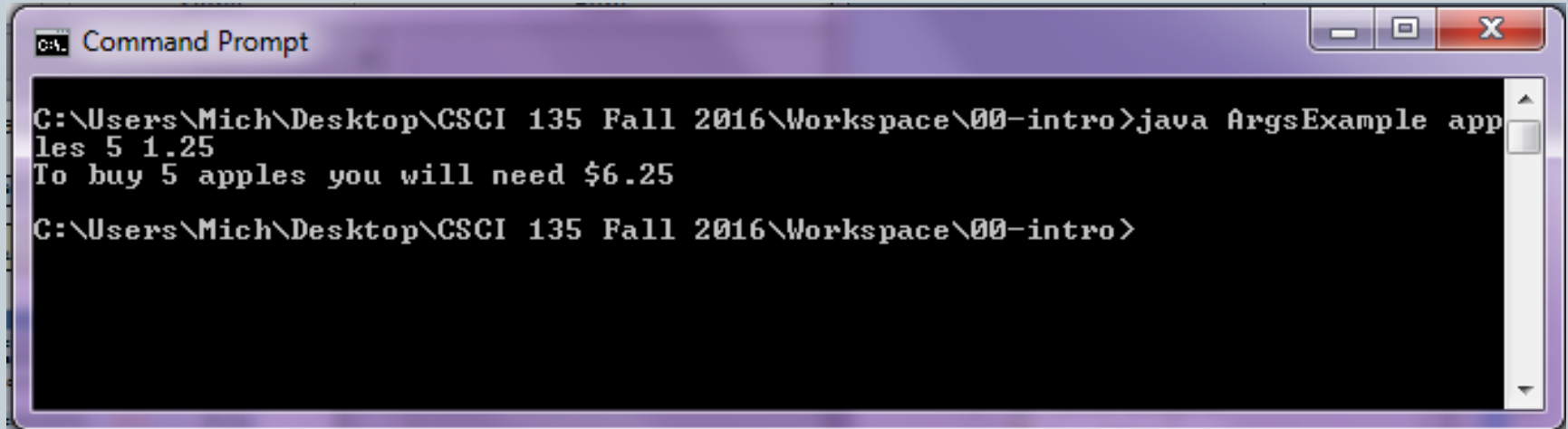
Apply Revert

Run Close

Filter matched 22 of 25 items

OUR FIRST PROGRAM

Command Line args in Command Shell



```
CA: Command Prompt
C:\Users\Mich\Desktop\CSCI 135 Fall 2016\Workspace\00-intro>java ArgsExample apples 5 1.25
To buy 5 apples you will need $6.25
C:\Users\Mich\Desktop\CSCI 135 Fall 2016\Workspace\00-intro>
```

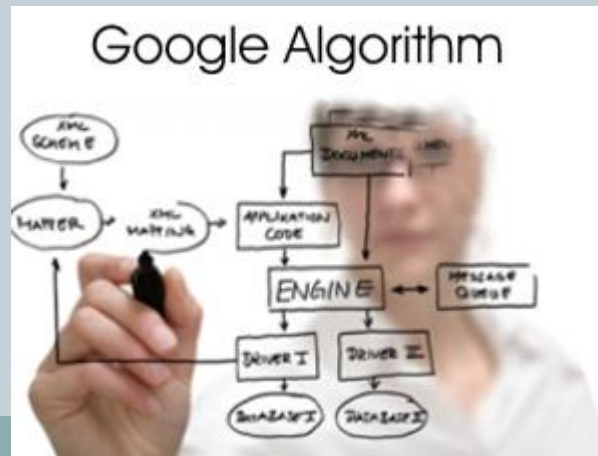

Some Terminology

OFFICIAL**DEFINITION**

- ***Arguments*** – the items inside the parentheses that provide data needed by methods
- ***Method*** – the code that is executed when called
- ***Variable*** – something that can store data
- ***Statement*** – an instruction to the computer; in Java it ends with a semicolon
- ***Syntax*** – the grammar rules for a programming language

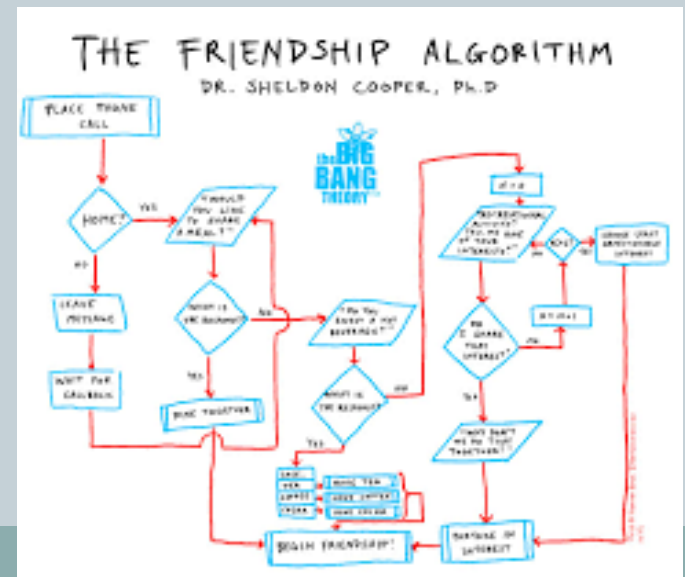
Algorithms

- By designing methods, programmers provide actions for objects to perform.
- An *algorithm* describes a means of performing an action.
- Once an algorithm is defined, expressing it in Java (or in another programming language) usually is easy.



Algorithms

- An algorithm is a set of instructions for solving a problem.
- An algorithm must be expressed completely and precisely.
- Algorithms usually are expressed in English or in *pseudocode*.



Example: Total Cost of All Items

- Write the number 0 on the whiteboard.
- For each item on the list
 - Add the cost of the item to the number on the whiteboard
 - Replace the number on the whiteboard with the result of this addition.
- Announce that the answer is the number written on the whiteboard.



Summary

- Course Overview
- Computer Basics
- Programs and Languages
- Our First Program
 - Java Compilation
 - Eclipse IDE
 - Anatomy of a Java Program
 - Command Line Arguments
- Algorithms

