1.1 Your First Program
Why Programming?

Why programming? Need to tell computer what to do.

“Please simulate the motion of N heavenly bodies, subject to Newton’s laws of motion and gravity.”

Prepackaged software solutions. Great, they do exactly what you want.

Programming. Enables you to make a computer do anything you want.

Ada Lovelace

Analytic Engine

well, almost anything [stay tuned]
Languages

**Machine languages.** Tedious and error-prone.

**Natural languages.** Ambiguous and hard for computer to parse.

- *Kids Make Nutritious Snacks.*
- *Red Tape Holds Up New Bridge.*
- *Police Squad Helps Dog Bite Victim.*
- *Local High School Dropouts Cut in Half.*

[ real newspaper headlines, compiled by Rich Pattis ]

**High-level programming languages.** Acceptable tradeoff.

“Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.” — Donald Knuth
Why Program?

Why program?
- A natural, satisfying and creative experience.
- Enables accomplishments not otherwise possible.
- Opens new world of intellectual endeavor.

First challenge. Learn a programming language.

Next question. Which one?

Naive ideal. A single programming language.
Our Choice: Java

Java features.
- Widely used.
- Widely available.
- Embraces full set of modern abstractions.
- Variety of automatic checks for mistakes in programs.

Java economy.
- Mars rover.
- Cell phones.
- Blu-ray Disc.
- Web servers.
- Medical devices.
- Supercomputing.
- ...

$100 billion, 5 million developers

James Gosling
http://java.net/jag
Why Java?

**Java features.**
- Widely used.
- Widely available.
- Embraces full set of modern abstractions.
- Variety of automatic checks for mistakes in programs.

**Facts of life.**
- No perfect language.
- We need to choose some language.

**Our approach.**
- Minimal subset of Java.
- Develop general programming skills that are applicable to many languages.

It’s not about the language!
# A Rich Subset of the Java Language

## Built-In Types

<table>
<thead>
<tr>
<th>int</th>
<th>double</th>
</tr>
</thead>
<tbody>
<tr>
<td>long</td>
<td>String</td>
</tr>
<tr>
<td>char</td>
<td>boolean</td>
</tr>
</tbody>
</table>

## System

<table>
<thead>
<tr>
<th>System.out.println()</th>
<th>System.out.print()</th>
<th>System.out.printf()</th>
</tr>
</thead>
</table>

## Math Library

<table>
<thead>
<tr>
<th>Math.sin()</th>
<th>Math.cos()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math.log()</td>
<td>Math.exp()</td>
</tr>
<tr>
<td>Math.sqrt()</td>
<td>Math.pow()</td>
</tr>
<tr>
<td>Math.min()</td>
<td>Math.max()</td>
</tr>
<tr>
<td>Math.abs()</td>
<td>Math.PI</td>
</tr>
</tbody>
</table>

## Parsing

<table>
<thead>
<tr>
<th>Integer.parseInt()</th>
<th>Double.parseDouble()</th>
</tr>
</thead>
</table>

## Flow Control

<table>
<thead>
<tr>
<th>if</th>
<th>else</th>
</tr>
</thead>
<tbody>
<tr>
<td>for</td>
<td>while</td>
</tr>
</tbody>
</table>

## Boolean

<table>
<thead>
<tr>
<th>true</th>
<th>false</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
</tr>
</tbody>
</table>

## Punctuation

<table>
<thead>
<tr>
<th>{}</th>
<th>()</th>
<th>,</th>
<th>;</th>
</tr>
</thead>
</table>

## Assignment

<table>
<thead>
<tr>
<th>=</th>
<th>+</th>
<th>-</th>
<th>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>%</td>
<td>++</td>
<td></td>
</tr>
<tr>
<td>--</td>
<td>&gt;</td>
<td>&lt;</td>
<td></td>
</tr>
<tr>
<td>&lt;=</td>
<td>&gt;=</td>
<td>==</td>
<td></td>
</tr>
<tr>
<td>!=</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## String

<table>
<thead>
<tr>
<th>+</th>
<th>&quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>length()</td>
<td>compareTo()</td>
</tr>
<tr>
<td>charAt()</td>
<td>matches()</td>
</tr>
</tbody>
</table>

## Arrays

<table>
<thead>
<tr>
<th>a[i]</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
</tr>
<tr>
<td>a.length</td>
</tr>
</tbody>
</table>

## Objects

<table>
<thead>
<tr>
<th>class</th>
<th>static</th>
</tr>
</thead>
<tbody>
<tr>
<td>public</td>
<td>private</td>
</tr>
<tr>
<td>final</td>
<td>toString()</td>
</tr>
<tr>
<td>new</td>
<td>main()</td>
</tr>
</tbody>
</table>

## Primitive Numeric Types

<table>
<thead>
<tr>
<th>+</th>
<th>-</th>
<th>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>%</td>
<td>++</td>
</tr>
<tr>
<td>--</td>
<td>&gt;</td>
<td>&lt;</td>
</tr>
<tr>
<td>&lt;=</td>
<td>&gt;=</td>
<td>==</td>
</tr>
<tr>
<td>!=</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Hello, World
Create the program by typing it into a text editor, and save it as HelloWorld.java.

```java
/**************************
 * Prints "Hello, World"
 * Everyone's first Java program.
 **************************/

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

HelloWorld.java
Programming in Java

Create the program by typing it into a text editor, and save it as HelloWorld.java.

Compile it by typing at the command-line:

```
javac HelloWorld.java
```

(or click the Compile button in DrJava)

This creates a Java bytecode file named: HelloWorld.class.
Programming in Java

Programming in Java.

- Create the program by typing it into a text editor, and save it as HelloWorld.java.
- Compile it by typing at the command-line:
  javac HelloWorld.java.
- Execute it by typing at the command-line:
  java HelloWorld.

```
% javac HelloWorld.java

% java HelloWorld
Hello, World
```
Dr. Java

http://drjava.org
Dr. Java

```java
/**
 * Compilation:  javac UseArgument.java
 * Execution:    java UseArgument yourname
 * Prints "Hi, Bob. How are you?" where "Bob" is replaced by
 * the command-line argument.
 */

public class UseArgument {
    public static void main(String[] args) {
        System.out.print("Hi, ");
        System.out.print(args[0]);
        System.out.println(" How are you?");
    }
}
```

compiler output:
```
javac 1.5.0 compiler ready.
```

Hi, Bob. How are you?
Hi, Alice. How are you?
Dr. Java

File: /Volumes/WAYNE/java/UseArgument.java

```java
/**
 * Compilation:  javac UseArgument.java
 * Execution:    java UseArgument yourname
 * Prints "Hi, Bob. How are you?" where "Bob" is replaced by
 * the command-line argument.
 * @param args
 */

public class UseArgument {
    public static void main(String[] args) {
        System.out.print("Hi, ");
        System.out.print(args[0]);
        System.out.println(". How are you?");
    }
}
```

Welcome to DrJava. Working directory is /Volumes/WAYNE/java

> java UseArgument Kevin
Hi, Kevin. How are you?
> java UseArgument Bob
Hi, Bob. How are you?
> 

command-line argument