1.1 Your First Program
Why Programming?

**Idealized computer.** "Please simulate the motion of a system of $N$ heavenly bodies, subject to Newton's laws of motion and gravity."

**Prepackaged software solutions.** Great, if it does exactly what you need.

**Computer programming.** Art of making a computer do what *you* want.
Languages

“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

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.
Why 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.

Caveat.

“There are only two kinds of programming languages: those people always [gripe] about and those nobody uses.” – Bjarne Stroustrup
Why Java?

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

Caveat. No perfect language.

Our approach.
- Minimal subset of Java.
- Develop general programming skills that are applicable to:
  - C, C++, C#, Perl, Python, Ruby, Matlab, Fortran, Fortress, ...
A Rich Subset of the Java Language

### Built-In Types
- int
- long
- char
- double
- String
- boolean

### System
- System.out.println()
- System.out.print()
- System.out.printf()

### Math Library
- Math.sin()
- Math.cos()
- Math.log()
- Math.exp()
- Math.sqrt()
- Math.pow()
- Math.min()
- Math.max()
- Math.abs()
- Math.PI

### Parsing
- Integer.parseInt()
- Double.parseDouble()

### Flow Control
- if
- else
- for
- while

### Boolean
- true
- false
- ||
- &
- !

### Punctuation
- {
- }
- (   )
- ,   ;

### Assignment
- =

### String
- +
- 
- length()
- compareTo()
- charAt()
- matches()

### Arrays
- a[i]
- new
- a.length

### Objects
- class
- static
- public
- private
- toString()
- equals()
- new
- main()
Create, Compile, Execute
Programming in Java

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

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
```

(or click the Run button in DrJava)
Dr. Java

http://drjava.org
Dr. Java

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

Java is:

- Object oriented.
- Statically typed.
- Architecture neutral.
- Multi-threaded.
- Garbage-collecting.
- Robust.
- Small.
- Simple.
- Fast.
- Secure.
- Extensible.
- Well-understood.
- Fun.

Java is:

- Not new.
- Designed for toasters.
- Not done yet.
- Not as useful as C, C++, FORTRAN.
- Slow.
- Unsafe.
- Huge.
- Complex.

Don't believe anything on this slide! Make up your own mind.
HelloWorld.class