Java is...

Enthusiasts will tell you Java is...
- Simple
- Object oriented
- Machine-independent
- Multi-threaded
- Garbage collecting
- Secure
- The language of the Web

Skeptics will reply that it is...
- Complex
- Slow
- Designed for toasters
- Not finished yet

Make up your own mind!

Java perspective

Similar to C in many ways
- types: int, float, double, char...
- control: for, if, while, do, switch...

Modern primitives
- Unicode: 16-bit "universal" character set

Higher level of abstraction than C
- Object oriented
- No pointers (just references to objects)
- Inheritance
- Language support for:
  - interfaces, threads, exceptions
- System support for:
  - Software libraries, security, graphics, network

How to learn Java

This is the language of the web, so learn it on-line!
- http://www.javasoft.com/
- http://java.sun.com/docs/books/tutorial/
- http://java.sun.com/jdk/

Start with existing code, read code, read docs

Experiment by making small changes
- Add functionality progressively
- (This is how you'll do your next assignment.)
“Hello World” in C

```c
#include <stdio.h>

int main(int argc, char *argv[]) {
    printf("hello, world\n");
}
```

Compile and execute

- gcc compile code into machine language
- a.out load and execute the program

+ Advantage: good compilers make efficient programs
- Disadvantage: translated code is machine-dependent

```
% gcc hello.c
% a.out
echo, world
```

Unix

“Hello World” in Java

```java
class Hello {
    public static void main(String args[]) {
        System.out.println("hello, world\n");
    }
}
```

Translate and interpret the program

- javac translate the code into byte stream
- java interprets the byte stream (Hello.class)

+ Advantage: machine-independent
- Disadvantage: interpretation is slower than running compiled code

(`Good native code java compilers are inevitable.`)

```
% javac hello.java
% java Hello
echo, world
```

Unix

Object-oriented programming

A different way of thinking about programming

- Some people would say that learning C has already ruined you.
- Direct support of abstract data type (ADT)
  - Encapsulate functions that manipulate data
  - Build levels of abstraction
- CLASS
  - Data type description
  - Functions that operate on data (METHODS)
- A program is a sequence of classes
- OBJECT
  - “instance” of a class
  - An object to a class in Java is as a variable to a type in C
- METHOD
  - instantiate objects
  - invoke other methods
**Class Implementation**

**Rational.java**

class Rational {
    int p, q;
    
    Rational(int p, int q) {
        this.p = p; this.q = q;
    }
    
    public String toString() {
        return Integer.toString(p) + "/" + Integer.toString(q);
    }
    
    void mul(Rational x) {
        p *= x.p; q *= x.q;
    }
    
    void add(Rational x) {
        p = p*x.q + q*x.p;
        q *= x.q;
    }
}

**Example Rational client**

e.java

class eCalc {
    public static void main(String[] args) {
        int j;
        Rational e = new Rational(1, 1);
        Rational t = new Rational(1, 1);
        for (j = 0; j < 6; j++) {
            Rational r = new Rational(1, j+1);
            t.mul(r);
            e.add(t);
            System.out.println(e + " " + t);
        }
    }
}

**Inheritance**

**RationalA.java**

class RationalA extends Rational {
    
    RationalA(int p, int q) {
        super(p, q);
    }
    
    private static int gcd(int m, int n) {
        if (n == 0) return m;
        return gcd(n, m%n);
    }
    
    private static void reduce(Rational x) {
        int t = gcd(x.p, x.q);
        x.p /= t; x.q /= t;
    }
    
    void add(Rational x) {
        p = this.q*x.q + x.p*this.q;
        this.q *= x.q;
        reduce(this);
    }
}

**Why java?**

Many of the ideas were around in older languages
- C++, Modula-3, Smalltalk, Ada, ...

What's new?
- Free, easy to install
- Works on most computers
- Support for large-scale software engineering
- Growing library of software available
- Internet
- Graphics
- Security

What's holding us back?
- Legacy code
- Libraries are complex
- Slow?
- Standards?