Scarborough College
University of Toronto
Fall 1997

CSC A06F: Introduction to Computer Programming
Lecture notes 1: Introduction

Reading:

- Course information handouts,
- Lewis and Loftus, chapters 1 and 2.

What is computer science?

The study of computation by machine.

The study of the design and structure of computer systems.

- Problem solving using algorithms and abstract data types.
- Theory and design of programming languages.
- Theory of computation.
- Software engineering: the design, construction, and management of very large programs.
- Software verification: testing and proving programs to be correct.
- Applications: e.g., artificial intelligence, databases, user interfaces, etc.

What is a computer system?

- **Hardware:** The physical, tangible parts.
  - CPU
  - main memory (e.g., RAM)
  - secondary memory (e.g., hard drives, diskettes, ...)
  - input/output devices (e.g., monitor, keyboard, mouse, ...)
- **Software:** The programs and data make the hardware do things. Two main kinds:
  - **System software:** The program that runs the computer itself, and provides an interface (e.g., Windows 95, Mac OS).
  - **Application software:** Any program that applies the computer to a task (e.g., Word processors, games, Mars Rover control software).

Everything the computer does requires a program; even creating, loading, and running *other* programs.

What is a program?

A **program** is a sequence of instructions to the hardware to *execute* a set of actions.

The hardware can perform many kinds of actions:

- Arithmetic calculations.
- Decide if a logical situation is true or not.
- Store or retrieve a value from memory.
- Store or retrieve data from a disk drive.
- Open a connection to another computer on the Internet.
- Draw pictures on the screen.
- ...

A program in Java

```
// Filename is "Max.java"
// Program to compute the maximum number in a set of positive
// numbers entered by the user

import java.io.*;

public class Max {
   public static void main (String args[]) throws IOException {
      DataInputStream stdin = new DataInputStream (System.in);

      int max = 0;

      // Read first number
      String token = stdin.readLine();
      int number = Integer.parseInt(token);

      while (number > 0) { // stop when we read 0
         // if the number is bigger than max, then it's the new max
         if (number > max) {
            max = number;
         }
         // read the next number
         token = stdin.readLine();
         number = Integer.parseInt(token);
      }
      System.out.println ("The maximum is "+max);
   }
}
```


What is a programming language?

Can't use English to tell the computer what to do. It's too vague, imprecise, and ambiguous. Also includes more than the computer can do.

Programming languages are precise, unambiguous, and limited to what the computer can do.

Different programming languages are designed for describing different kinds of programs:

> Pascal, Turing, C, C++, Visual Basic, SmallTalk, Fortran, Lisp, Prolog, Ada, PL/1, …

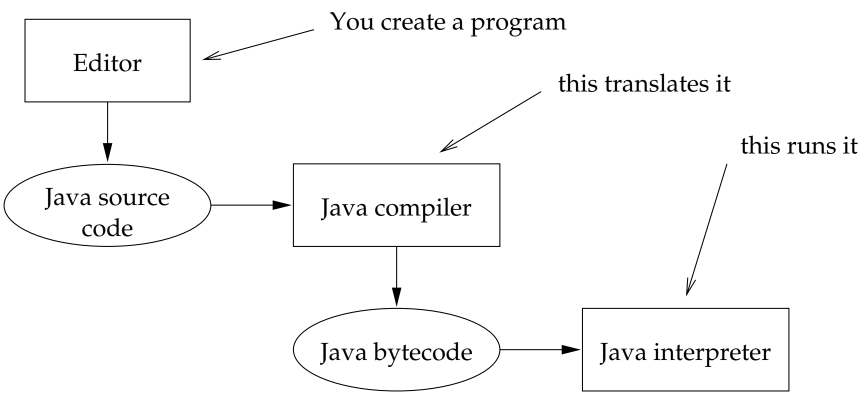
Java

- A simple, robust, object-oriented programming language (OOPL).
- A set of standardized packages of pre-written code.

Very suitable for network applications, e.g., on the Internet.

But also good for any small-scale application.

Using Java



The ***Java Development Kit*** (JDK) comes with a *compiler*, `javac`, and an *interpreter*, `java`.

```
> javac Max.java        // compile the maximum program
> java Max              // runs it
5                       // input some numbers
8
4
0
The maximum is 8        // here is the output
>
```

Another way to use Java

An **integrated development environment** (IDE) combines the editor, compiler, interpreter, and other useful programs into one application.

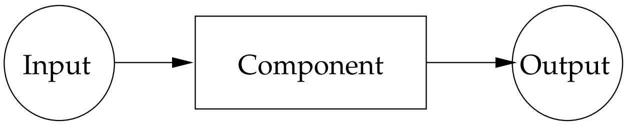
VisualAge for Java is an IDE.

Just write your program in one or several editor windows, then press the "Run" button. Up pops a window for input and output.

Software systems

Software systems are built out of **software components** :

- Performs a specific task.
- Transforms input to output.
- Can be combined with other components.



Building software

Will focus on the art and science of building software, which involves *problem-solving* at its heart.

- Analyze the problem.
- Design a solution:
  - What components are necessary?
  - How do they interact?
  - What kind of *data* does the component need?
  - What *algorithms* transform the data?
- Write programs for each component.
- Test and fix the programs.
- Once in use, maintain the system.

What is an algorithm?

Algorithm: A precisely defined, finite sequence of steps that yields the answer to a well-defined class of problems.

In other words, an algorithm is a *method* for doing something.

E.g., The algorithm for finding the maximum of a set of numbers:

> If we have a set of numbers, then to find the maximum, look at each number in the set in turn, and if it is greater than the maximum seen so far, set the maximum to that number.

An algorithm is a mathematical concept.

Writing down an algorithm in a programming language is called **implementing** the algorithm and results in a program.

A large part of computer science is:

- The study and design of algorithms
- Turning algorithms into usable computer programs

Classes and objects

When we write an algorithm down in Java, and give it a name, we call it a **method** .

A method operates on some data (e.g., a set of numbers), and can return a result (e.g., the maximum).

When we have several different methods that operate on the same data, we can group them into an single **object** with their data.

We define an object in Java using a **class** .