

## CSC 142

### Introduction to Java [Reading: Appendix A and chapter 1]

CSC 142 A 1

## What to expect in this class

- Background: knowledge of basic programming in any language (VB, C, ...)
  - functions (or subs), loops, conditionals and arrays
- Outcome: how to program in Java (syntax, using objects, java libraries, ...)
- Class organization:
  - Lectures, lab practice with sample code, homework assignments, tests and quizzes
  - class web site  
[seattlecentral.edu/~flepeint/javaclass](http://seattlecentral.edu/~flepeint/javaclass)

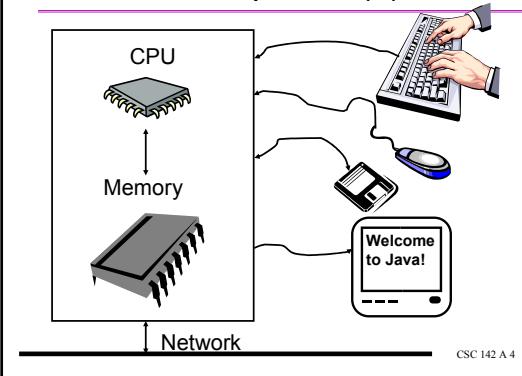
CSC 142 A 2

## What is a computer? (1)

- Programmer's viewpoint
  - CPU (central processing unit): the "brain" of the computer. It can perform simple tasks very fast (e.g. adding...)
  - Memory: where the microprocessor stores data (results of computations,...). Memory comes in many types (RAM, ROM, ...)
  - Input/Output devices: e.g. keyboard, screen, ... They allow the user to interact with the computer.
- Our goal: Make the computer perform complex tasks for us.
- How: Write programs.

CSC 142 A 3

## What is a computer? (2)



CSC 142 A 4

## Programming

- Machine language
  - The CPU set of instructions: coded as a series of 0's and 1's
  - Fast
  - Machine dependent (two different CPU don't use the same machine language)
  - Time consuming to write programs
- Instead, use a high level language
  - Closer to plain English. Programming is easier. (this doesn't mean easy!)
  - Machine independent
  - Can't be understood by the CPU. A translation program is required (compiler, interpreter, ...)

## Programming languages

- Thousands of programming languages (many are just for research purposes)
- Historically (with many omissions!)
  - FORTRAN (scientific) (50's)
  - COBOL (business) (60's)
  - PASCAL, BASIC, C (multi purposes) (70's)
  - C++ (OOP) (80's)
  - Java (OOP + platform independence) (90's)

CSC 142 A 6

## Why are we learning Java?

- A modern language
  - object oriented, portable, secure
  - able to harness the power of networks
- A better C++?
  - some features of C++ are overly complex (e.g. multiple inheritance)
  - Java doesn't include any of these
- A good language to learn programming (the modern way: OOP, ...)

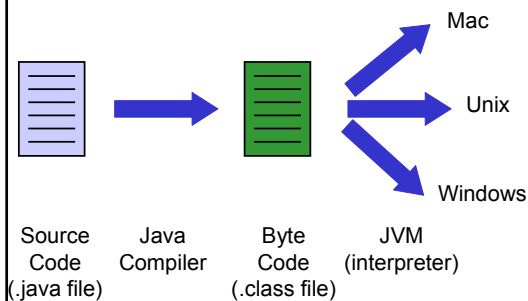
CSC 142 A 7

## A brief history

- Created in the 90's by Sun Microsystems (Bill Joy and James Gosling)
  - first for cell phones and PDA (failure)
  - explosion of the internet rescued the language (small, robust, object oriented, architecture independent, secure)
  - free and open code distribution
- Now
  - used widely on PC, cell phones, palm pilots, ...
  - But there are other languages, e.g. Python, etc...

CSC 142 A 8

## The Java virtual machine



CSC 142 A 9

## Java Applications vs Applets

- Applications: a stand alone java program that is executed locally on your computer
  - Applications can be trusted (in general)
- Applets: a java program that is executed in the context of a web browser (this is how java programs can easily be distributed)
  - Applets should not be trusted

CSC 142 A 10

## Some security features

- An applet can't
  - read or write from or to the computer's file system
  - run programs on the user's computer
  - establish a connection between the user's computer and any other computer except for the server where the applet is stored.
- Java doesn't support pointers (a programmer can't access the memory directly)
- Memory management is done automatically by the JVM (not the programmer's job  $\neq$  C++)

CSC 142 A 11

## The present Java

- Java 6
  - Many libraries
    - Swing (user interface toolkit)
    - Data bases (JDBC)
    - Java security, Servlets ...
- Software: many free products for all platforms (Unix, Solaris, Mac, Windows...)
  - Sun jdk available at [java.sun.com](http://java.sun.com)
  - BlueJ (developed by Monash University, Australia) available at [www.bluej.org](http://www.bluej.org) (what we will use).
  - Java IDE's: e.g. by Borland ([www.inprise.com](http://www.inprise.com))

CSC 142 A 12

## Resources

---

- Text: "Objects first with Java" Prentice Hall
- Other books
  - Niemeyer & Knudsen: "Learning Java" O'Reilly (Good if you already know C/C++)
  - Deitel & Deitel: "Java how to program" (lots of examples)
- Enormous amount of resources on the web
  - [java.sun.com](http://java.sun.com) (excellent on-line tutorials)
  - [cs.washington.edu](http://cs.washington.edu) (CSE 142/143 are now taught in Java). We will use the UW java library.