CSC 142

Introduction to Java

[Reading: chapters 1 & 2]

1

What to expect in this class

- Background: knowledge of basic programming in any language (Python, 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

    http://flepeint.ccfaculty.org/java142

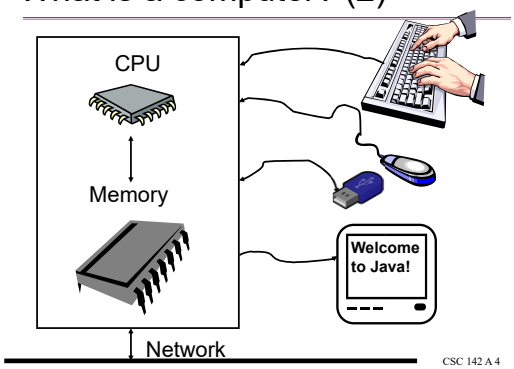
2

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.

3

What is a computer? (2)



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, …)

5

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)

6

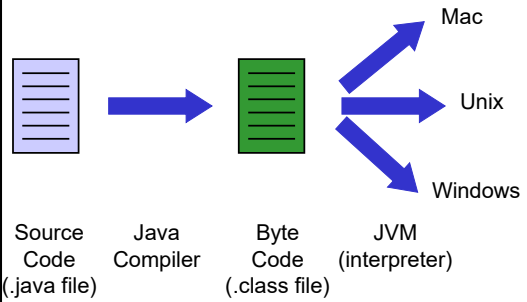
Why are we learning Java?

- 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, …)

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, …
  - But there are other languages, e.g. Python, etc…

The Java virtual machine



The present Java

- Java 15
  - Many libraries
    Swing (user interface toolkit), JavaFX
    Data bases (JDBC)
    Java security, Servlets …
- Software: many free products for all platforms (Unix, Solaris, Mac, Windows…)
  - Java jdk available at oracle.com
  - Eclipse at eclipse.org (what we will use).
  - Other Java IDE's available.

Resources

- Text: “An introduction to OOP with Java" McGraw-Hill by C. Thomas Wu
- Other books
  - “Building Java programs” Pearson by Reges & Stepp
- Enormous amount of resources on the web
  - docs.oracle.com (excellent on-line tutorials)
  - Check www.cs.washington.edu. We will use the UW java library.