Java Chapter 1

* Programming and Java
* CIS 255 • Shelby-Hoover Campus

Computer Basics

* Hardware
  + Central Processing Unit (CPU)
  + Main Memory (RAM)
  + Secondary Storage: Hard Disks, Flash Memory, Optical Discs
  + Input Devices: Keyboard, Mouse
  + Output Devices: Monitor, Printer
* Software
  + Program: a set of instructions for the computer to perform, usually on a set of data
  + Most application software communicates with the computer through the operating system

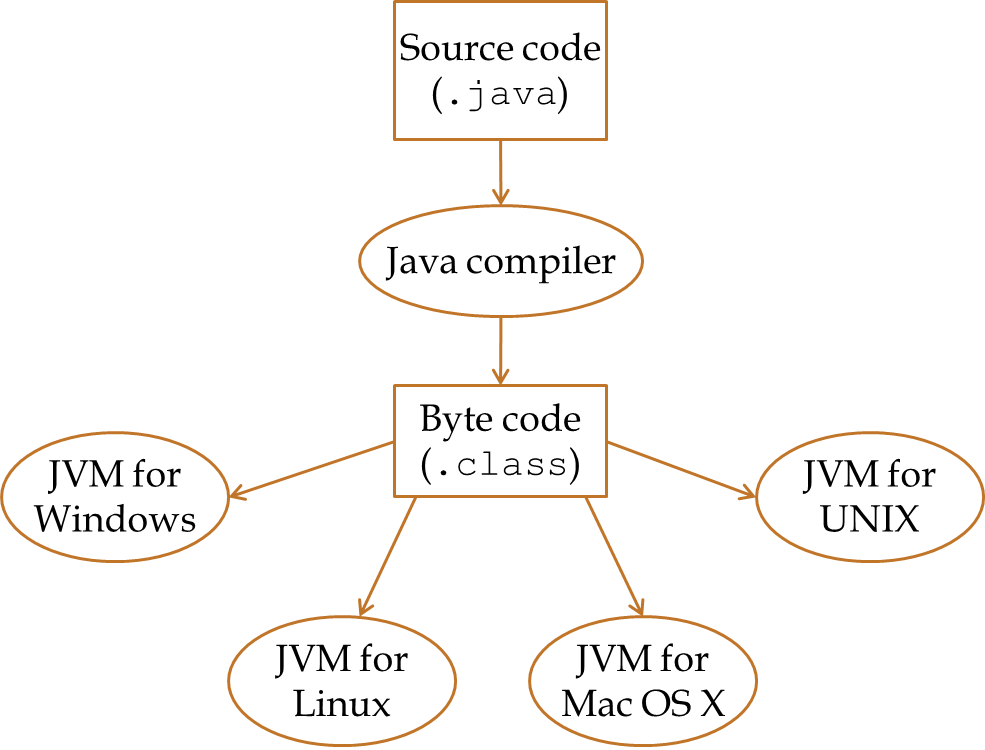
Programming Basics

* Underlying information is stored / manipulated in **binary** (0 for off, 1 for on)
* Processors understand rudimentary memory operations written in **machine language** (binary)
* Programmers usually write programs in a programming language (Table 1-1)
  + C++, C#, Java, Basic, Pascal, FORTRAN, COBOL
  + Easier for humans to write, understand

High Level to Low Level

* Code written in a high-level language must be converted to a low-level language form before the instructions can be executed
* A **compiler** converts a set of high-level language instructions into a low-level language all at once
  + The high-level language code is known as **source code**
  + The low-level language code that results is **object code**
* In Java, the code generated by the compiler does not target a specific machine architecture but rather an intermediate form called **byte code**
  + This byte code can be executed via the **Java Virtual Machine (JVM)**, a hypothetical computer that interprets the byte code to generate machine language specific to a computer
  + Byte code is **portable**: the instructions in a single byte code file can be executed in the JVM on a variety of platforms, eliminating the need to compile a separate file for each target platform

Source Code and Byte Code



Language Elements

* **Key words / Reserved Words**: words with predefined meaning (cannot be used for other purposes)
* **Operators**: symbols or words that perform operations
* **Punctuation**: characters that mark the beginning or end of a statement or block, separate items in a list, etc.
* **Programmer-Defined Names / Identifiers**: identify memory locations, code units, new data types
* **Syntax**: rules that govern how the various language elements are combined to form statements and programs

The Programming Process

* Remember: the point at which you enter code in a specific language should not be the beginning point of the programming process
* There are several different methods that can be applied to a problem to generate a solution
  1. Clearly define what the program is to do (as an algorithm)
  2. Visualize the program running on the computer
  3. Use design tools (flowcharts, pseudocode) to create a model of the program
  4. Check the model for logical errors (tracing)
  5. Enter the code and compile it
  6. Correct errors found during compilation; repeat 5 & 6
  7. Run the program with test data for input
  8. Correct errors found during execution; repeat 5 through 8
  9. Validate the results of the program

Programming Approaches

* **Procedural Programming** separates the data from the instructions performed on the data; the focus is on the sets of instructions (procedures, functions, methods)
* **Object-Oriented Programming** unites individual data elements into units with predefined interaction
  + **Object**: a group of variables representing some entity
  + **Method**: a procedure that can be used to access or modify the data stored in the object
  + **Class**: the code that defines the makeup of a particular type of object, including the data each object contains and what operations can be performed on that object

An Introduction to Java

* Developed by Sun Microsystems in the early 1990s
  + Designed to allow programs to run on a variety of home entertainment devices
  + Originally known as Oak
  + Eventually applied to the Internet, incorporated into Netscape Navigator
* Java programs may be applications or applets
  + Applications are standalone programs than run locally on a computer
  + Applets are designed for embedding in web pages

Java Structure

* Your initial programs will take this form:

🡨 import statements for optional functionality go here

public class ClassName

{

public static void main(String[] args)

{

🡨 Your code goes here

}

}

* The identifier ClassName in the first line after the import statements should match the name of your Java file
* All Java programs require a “class”, but a Java program that contains only static methods (like main) is procedural

Some Basic Java Syntax

* In Java, a statement (a single instruction) ends with a semicolon
* Declaration
  + Listing the data type and name of a variable to be used
  + Example: int studentCount; (int is used for whole numbers)
* Output
  + The System variable out represents output to the console (a text-based screen)
  + The methods print and println send output to the console via System.out
    - System.out.print("Hello!"); displays Hello!
    - System.out.println(studentCount); displays the value of the variable studentCount and starts a new line of output below it
* Assignment
  + Storing a predetermined value in a variable
  + The variable must be on the left side of the equal sign; the value goes on the right
  + studentCount = 15; assigns 15 to the variable
  + studentCount = studentCount + 1; adds 1 to the existing value of the variable and assigns the sum as the new value