Introduction to Computers and Programming

Lecture 1: administrative details and an introduction to computers

Chris Conway
New York University
Road Map for Today

• Welcome to Introduction to Computers and Programming!
• Course Description
  – What material will we cover?
  – What am I getting myself into?
• Administrative Issues
  – Course Web Page, Text Book, Exams, Office Hours, Homework, Grading, Cheating Policy, etc.
• Syllabus
• History of Computers
• Intro. to Programming Languages and Java
• Reading
  – Liang 6 and Liang 7: chapter 1.1 – 1.4, 1.6 - 1.7
  – (most of today’s material is not in the book)
Course Prerequisites

• Prerequisites:
  – No prior programming experience required (Really)

• Who should be taking this course:
  – students who want to switch to a computer science major
  – students who want a computer science minor or a computer applications minor
  – students who are just interested in programming.

• Who should NOT be taking this course
  – Students trying to get out of taking a math requirement. This class may be more difficult than the math you are trying to avoid.

• You must get a C or better in this class to take further computer science classes.
Course Description

- **Official Description:** Elementary introduction to programming. The characteristics of computers are discussed and students design, code, and debug programs using a high level programming language.
What the class is really about

There are three main goals of this course:
1. Basics of Java
2. Core Concepts of Programming Languages
1. Learn the Basics of Java Programming

- Java is a popular programming language, widely used in industry.
- We will learn all the specifics of how to program in Java.
- This includes all the peculiar rules that are specific to Java.
- We will cover the fundamentals: Variables, Arithmetic, If / Else, For Loops, While Loops, Arrays, Methods, etc.
This program counts from 1 to 10. In a few weeks, it will all make sense (I promise!)
2. Learn the Core Concepts of all Programming Languages

- There are many programming languages available: Pascal, c, Java, Ada, Perl.
- All of these languages share core concepts.
- By focusing on these concepts, you are better able to learn *any* programming language.
- Hence, by learning Java, you are poised to learn other languages, such as C++ or Perl.
- By learning the core concepts, you are also much more marketable as you are able to learn new technologies quicker.
- **Note:** Java is an object oriented programming language. However, we will not touch upon the concepts which categorize it as one.
An Example: For Loops

- Java has a construct called a `for` loop that enables a program to repeat actions over and over.
- Most other languages also have a `for` loop.
- Hence, by learning about for loops in Java, you can easily learn for loops in C or Ada.
3. Learn the Principles of Software Development

- Building high quality software is *very* difficult.
- The course presents the syntax and concepts of programming, and also presents strategies for building real software that addresses real problems.
- I will also try to bring my real-world industry experience to class.
Administrative Matters
Course Web Site

• Course web site is available at:
  http://cs.nyu.edu/courses/summer09/V22.0002-001/
Web site contains the following information:
  – Administrative information
  – Course Syllabus
  – Homework assignments
  – Class notes
  – Class programs
  – Sample exams
  – Compiler instructions
  – Link to the class mailing list
Class mailing list

• First assignment is to join it. Do it today!
• Go to:
  http://cs.nyu.edu/mailman/listinfo/v22_0002_003_sp09
  and follow the instructions
• All assignments and news will be sent to the class list
• Homework questions should be sent to the list and answered by students when possible.
Course Text Book

• Introduction to Java Programming (7th Edition) Brief Version
• Available at the NYU Bookstore
• Book includes a CD-ROM with Java programs and other supplemental materials.
• Lecture notes will follow the book.
• Please keep up with the reading!
Software

• For the course, you may use any IDE you are comfortable using. I will use one or more of the following in the classroom:
  – Eclipse
  – JCreator (Note: this IDE has no debugger)
  – Netbeans

• All these products can be downloaded from the web for free.

• The class website has instructions on downloading and installing these programs.

• If you do not have your own computer, the computer labs on campus have the software.
Grading

• Your grade will be determined as follows:
  – Homework (30%)
  – Midterm (30%)
  – Final Exam (40%)

• Class participation will help your grade!
homework

Ten points will be deducted for each class day late, with a possible maximum of 30 points being deducted.

Home works will not be accepted after the third class following its due date.

For each assignment that you do not hand in within the time limit, your final grade will be lowered by one letter grade (i.e., if you are averaging a B+, but you have missed 2 home works, your final grade will be B-).

Submit the program via email to the grader (more on this later)

**Back up your work.** Computer crashes or lost programs are not valid excuses for not handing in an assignment.
A Word About Cheating

• For the purposes of this class, cheating is defined as by the CS Department’s academic integrity policy – Discussing homework concepts is fine, but you must submit your own work.

• If you are caught cheating, you will receive an immediate FAILURE for the course.
Student Civility

• In an effort to make this class enjoyable for everybody...
  – Please be on time to class!
  – Please do not talk to your friends and neighbors in class! It disturbs everyone, and makes it hard to concentrate. If you have a question, just ask me!
  – Please turn your pagers and cell-phones off!
Help is always available

• Option 1: Come to my Office Hours
  – Monday and Wednesday 11:15 - 12:30
  – Location: Room 404 Warren Weaver Hall
  – I get bored when nobody visits!
  – If you cannot make my office hours, I will be happy to make an appointment with you. Please try to give me advance warning when you need an appointment.

• Option 2: Write to the class mailing list. Please do not send homework code to the list.
The purpose of this course is to teach you about computing, but particularly, programming in Java (a powerful, widely-used programming language).

Why care about computers and programming?

- Enabling technology
- Growing field with great opportunity (read: $)
- Creative outlet
What Is a Computer?

• Computer
  – Performs computations and makes logical decisions
  – Millions / billions times faster than human beings

• Computer programs
  – Sets of instructions by which a computer processes data

• Hardware
  – Physical devices of computer system

• Software
  – Programs that run on computers
Computer Organization

● Six logical units of computer system
  – Input unit
    • Mouse, keyboard
  – Output unit
    • Printer, monitor, audio speakers
  – Memory unit
    • Retains input and processed information
  – Arithmetic and logic unit (ALU)
    • Performs calculations
  – Central processing unit (CPU)
    • Supervises operation of other devices
  – Secondary storage unit
    • Hard drives, floppy drives
Evolution of Operating Systems

- **Batch processing**
  - One job (task) at a time
  - Operating systems developed
    - Programs to make computers more convenient to use
    - Switch jobs easier

- **Multiprogramming**
  - “Simultaneous” jobs
  - Timesharing operating systems
Personal Computing, Distributed Computing, and Client/Server Computing

• Personal computers
  – Economical enough for individual
  – In 1981 IBM introduced the IBM personal computer using “off the shelf” components.
The boot process

- The process by which a machine comes up from rest state to the state that is usable is known as booting
- When the power is turned on
  - The CPU runs the BIOS (Basic Input / Output System)
    - Usually located on a chip on the motherboard
    - Runs POST (Power On Self Test) of various hardware components
    - Loads the boot sector program
The boot process (continued)

– Boot Sector Program
  • Located in the first sector of the hard disk or floppy disk
  • Is responsible for loading the rest of the operation system into the RAM

– Operating System
  • Once it is loaded, it configures the various hardware components
  • Then it waits for the user to issue commands
  • Then you can run your applications
Three types of programming languages

- Machine languages
  - Strings of numbers giving machine specific instructions
  - Example:
    
    | 100011 | 00011 | 01000 | 00000 | 00001 | 000100 |
    | 000000 | 00010 | 00011 | 00001 | 00000 | 100000 |
    | 101011 | 00011 | 01000 | 00000 | 00001 | 000100 |

- Assembly languages
  - English-like abbreviations representing elementary computer operations (translated via assemblers)
  - Example:
    
    | LOAD   | BASEPAY |
    | ADD    | OVERPAY |
    | STORE  | GROSSPAY |
High-level languages

- Instructions closer to everyday English
  - English is a natural language. Although high level programming languages are closer to natural languages, it is difficult to get too close due to the ambiguities in natural languages (a statement in English can mean different things to different people – obviously that is unacceptable for computer programming).

- Use mathematical notations (translated via compilers)
- Example:

  \[ \text{grossPay} = \text{basePay} + \text{overTimePay} \]

- Interpreter – Executes high level language programs without compilation.
Some Procedural High-level Languages

- Other high-level languages
  - FORTRAN
  - COBOL
  - Pascal
  - C
  - C++
  - Python
  - Javascript
The Key Software Trend: Object Technology

• Objects
  – Reusable software components that model items in the real world
  – Meaningful software units
    • Date objects, time objects, paycheck objects, invoice objects, audio objects, video objects, file objects, record objects, etc.
    • Any noun can be represented as an object
  – Very reusable
  – More understandable, better organized, and easier to maintain than procedural programming
  – Favor modularity
Characteristics of Java

- Java is simple
- Java is object-oriented
- Java is distributed
- Java is interpreted
- Java is robust
- Java is secure
- Java is architecture-neutral
- Java is portable
- Java is fast
- Java is multithreaded
- Java is dynamic
Basics of a Typical Java Environment

• Java programs normally undergo five phases
  – Edit
    • Programmer writes program (and stores program on disk)
  – Compile
    • Compiler creates bytecodes from program
  – Load
    • Class loader stores bytecodes in memory
  – Verify
    • Verifier ensures bytecodes do not violate security requirements
  – Execute
    • Interpreter translates bytecodes into machine language
Typical Java environment

Program is created in an editor and stored on disk in a file ending with .java.

Compiler creates bytecodes and stores them on disk in a file ending with .class.

Class loader reads .class files containing bytecodes from disk and puts those bytecodes in memory.

Bytecode verifier confirms that all bytecodes are valid and do not violate Java’s security restrictions.

Interpreter reads bytecodes and translates them into a language that the computer can understand, possibly storing data values as the program executes.
Another Basic Step for Java Programming

- Debugging
  - Check program execution and output to ensure program compiles and runs as expected
  - If it doesn’t, make corrections in the edit phase and repeat the remaining steps
Hardware Trends

• Every year or two the following approximately double:
  – Amount of memory in which to execute programs
  – Amount of secondary storage (such as disk storage)
    • Used to hold programs and data over the longer term
  – Processor speeds
    • The speeds at which computers execute their programs