CSE 115/503

January 18-22, 2010

Announcements

- If you have not picked up a syllabus, please do so
- Sign and return form on last page of syllabus – will be part of Lab 1 assignment.
- If you are interested in changing to another recitation and there are open seats, please change your registration through MyUB. If there are not open seats, please fill out form.
- If you are experiencing any kind of registration issues, please see me after class.
- Recitations began this week

Programming a Computer

- We can write instructions to the computer in machine language (the native language of that particular machine), but this is difficult and time-consuming.
- So, early in computer programming history, we created assembly language.

ADD r1 r2 3+4
STOR r3 r6

Assembly Language

- Simple mnemonics that indicate the type of action to be performed.
- Low-level language
- There is a one-to-one correspondence between the lines of assembly language and the machine language for the particular machine.

Higher-level Languages

- Assembly language works well, but a need was recognized to make programming languages more like human languages, and higher-level languages were developed.
- Higher-level languages are more complicated than assembly language
- There is no one-to-one correspondence between one line of a higher-level language and machine language

Higher-Level Languages

Java → object-oriented language

↓ strongly typed language

↓ compiled language

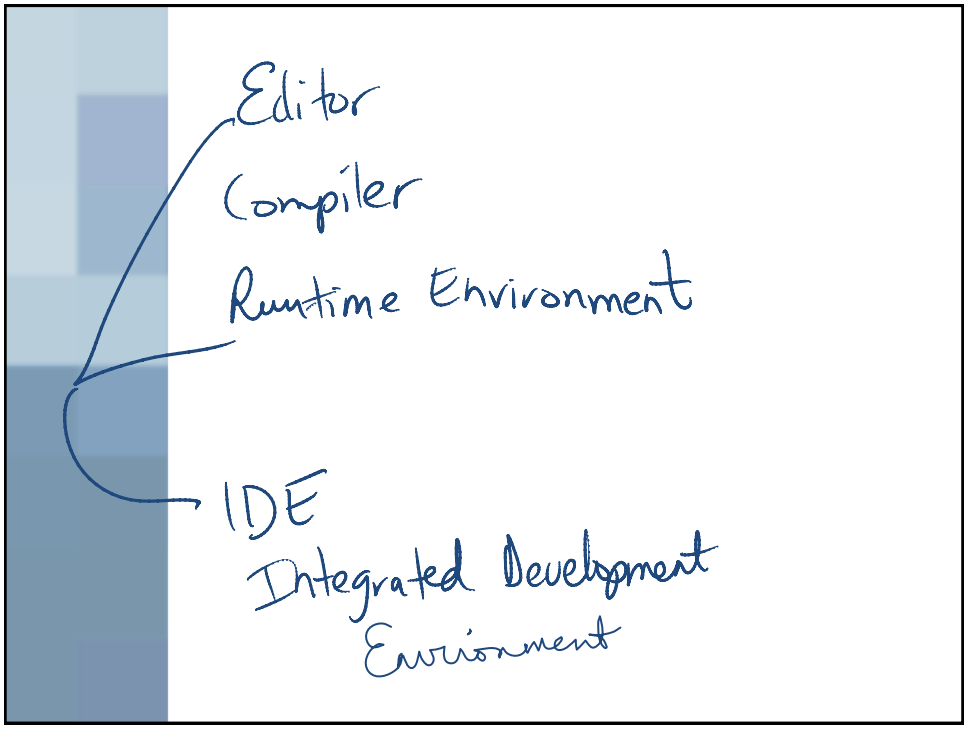
The Boehm-Jacopini Theorem

- If interested in more information, feel free to check out Bill Rapaport's webpage on the Boehm-Jacopini Theorem:
  http://www.cse.buffalo.edu/~rapaport/111F04/greatidea3.html
- In order to compute, we need the following three things:

- Sequencing – define the order things will be executed in
- Selection – choice
- Repetition

To program

- In order to create programs, we need several software tools to help us



Our IDE: Eclipse

- Check out the Resources/Course-Specific Resources section of the course website for information about downloading Eclipse on your own machine

Object-oriented programs

System of objects that work together to solve some problem

Objects

- can do things
- have properties & state

Recall our demonstration

If I tell you…	…object 1, you should…	…object 2, you should…	…object 3, you should…
START	Raise and lower your arms repeatedly	Start counting out loud, from zero (somewhat slowly) 0…1…2…3	Do nothing
STOP	Put arms down and remain still	Stop counting, but remember where you left off	Do nothing
CONTINUE	Do nothing	Keep counting from where you left off	Do nothing
ANYTHING ELSE	Do nothing	Do nothing	Do nothing