

# Introduction to Programming

*with Java, for Beginners*

Welcome

## Recitation General Information

- Instructor:
  - Diana Palsetia ([palsetia@seas.upenn.edu](mailto:palsetia@seas.upenn.edu))
  - Location: 174 Moore, Office Hours: TBA
- Assistants:
  - TA: Leon Hermans
    - Email: [leonoh@seas.upenn.edu](mailto:leonoh@seas.upenn.edu)
    - Office Hours & Location: TBA
  - TA & Grader: Freeya Azad
    - Email: [frevaa@seas.upenn.edu](mailto:frevaa@seas.upenn.edu)
    - Office Hours & Location: TBA
- Meeting Times:
  - TR: 10:30-11:30 am in Moore 100 A

## Introduction

- ESE112 involves 2 components:
  - Engineering Lab w/ Theory (.5 cu)
  - Computer Programming Recitation with Java Programming Language (.5 cu)
- You must enroll in both components
  - If you already know Beginner Level Java then come see me after class

## Our Web Site

- This site is our primary communication vehicle:  
<http://www.seas.upenn.edu/~ese112>
- Become familiar with it !!
- For homework/lab work submission (Digital Dropbox on Blackboard)  
<https://courseweb.library.upenn.edu>

## Logistics

- Grades
  - 3 Exams: 55 % (15%, 15%, 25%)
    - Exam 1 (week of 2/11)
    - Exam 2 (week of 3/17)
    - Exam 3 (Final Exam Period)
  - Homeworks: 30%
  - Lab: 15% (attendance grade! – 2 grace days, after which there is penalty for not attending)
- Late Policy for labwork & homework: 10% off per day upto 4 days and then no credit
- No makeup exams are scheduled. Conflict? Schedule in advance

ESE112

4/15

## Java Books

- Required: [Java Backpack Reference Guide](#) (Java 5 edition) by DePasquale, Addison-Wesley
  - This is a good, light, concise, and cheap reference book. It is recommend that you bring it with you to the labs.
- Optional: [Murach's Beginning Java 2 JDK 5](#) by Doug Lowe, Joel Murach, Andrea Steelman
  - This book has detailed explanations of each topic, and good examples to go with them.
- Others
  - **Online:** See the [Java Resources](#) page on course website

ESE112

5/15

## Computer Programming

- Computer Programming is sub-field of Computer Science
- It involves learning how to *translate* a complex problem/simulation/game in to a *computing solution*

ESE112

6/15

## Computing

- Formal Definition:
  - The use of a *computer* process data or perform calculations
- Early computers were *people*
- Advancement in *electronics* enabled computing devices that were faster than man
  - A computer no longer tied to the desk anymore!!
  - E.g. PDA, Cell phones



Source: <http://cannon.sfsu.edu/~gmarcy/cswa/history/pick.html>

ESE112

7/15

## Computing Examples



User information Assurance



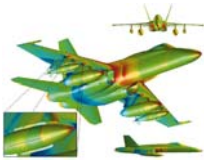
Mars  
Rover

Source: <http://www.physics.arizona.edu/~thews/reu/granados.htm>

Digital Special  
Effects



Information Security



Performance Simulation

ESE112

Other Sources: <http://coweb.cc.gatech.edu/ice-gt/274>

8/15

## Computer Science in General

- Computer Science is the study of
  - What we can do with computers?
  - Automate a known solution
  - How we can best do it
  - Investigate things we don't know
    - Example: The *Blue Brain Project* by EPFL & IBM started in July 2005

ESE112

9/15

## How does CS fit with Engineering?

- Computer Science *partners* well with many fields: finance, engineering, graphics, linguistics, genetics, multimedia, etc.
- Its theoretical foundation lies in
  - Mathematics
  - Electrical engineering
- Where does Programming Craft fit in with ESE112?
  - Serve to engineer robotic system



Edubot

ESE112

10/15

## Programming Languages

- Computer Programming
  - Is telling the computer how to do something
  - Wikipedia Definition: Applies specific *programming languages* to solve specific computational problems with solutions
- Programming Languages
  - Unlike human languages
    - Designed for *instructing* computers to solve problems
    - The listener (the compiler) is exacting & unforgiving (grr!)
  - Like human languages
    - They have a *grammar*
    - We will be learning grammar for *Java* Programming Language

ESE112

11/15

## Programming Language Syntax

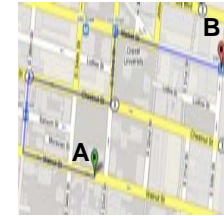
- Syntax is the *grammar* of the language
  - The Listener a.k.a Compiler will point out every syntax error
    - Analogous to rules in English Language:
      - Missing a period after sentence
      - Rules using verbs, nouns etc..
  - Error messages may be helpful
    - Often, they are not
    - You gain experience with error messages after a while

ESE112

12/15

## Program

- Writing a computational solution in a programming language
  - Known as writing a *program*
- When the computer carries out your solution
  - **Running** or **Executing** a program
- Computer Software/Application
  - Is made of one or more programs



Direction from A to B



Baking a cake

13/15

ESE112

## Programming Language Semantics

- Semantics is the *meaning* of the program
  - We learn the semantics after we run or execute the program
  - Basically we observe the output
- After the **executing** program, the semantics of the program may or may be correct
- Semantic errors cause your answers to be wrong
  - E.g. Add the juice of three onions to a cake recipe
  - You may or may not get error messages
    - E.g. Error Message – Dividing a number by zero
  - If your program is not doing what you want it to do, though it runs, the error is *semantic*

ESE112

14/15

## Programming -The Craft

- People have different tastes in programming, but many values are held in common
- Programming is an *art* as well as a *craft*
- There are *concepts* fundamental to all programming languages
  - We will practice the fundamentals using Programming Language called *Java*
  - Java also has additional feature called Object Oriented Programming (OOP) model
    - Design problems/programs such that they correspond to real world entities

ESE112

15/15