CMPT 120
Introduction To Computing Science And Programming I

Summer 2012

Instructor: Hassan Khosravi

Admin

- See website http://www.cs.sfu.ca/~hkhosrav/personal/py/120-2012.html

Textbook

- Introduction to Computing Science and Programming I by Greg Baker
  - One page per sheet
  - Two page per sheet
  - Four page sheet
- How to Think Like a Computer Scientist -- Learning with Python
  - PDF format
  - HTML format
  - HTML zip format
- Lecture slides

- Reference Books
  - Learn Python The Hard Way, Zed Shaw
    - by downloading it or buying it from its home site

Course Grading

- Lab assignments/quizzes 13%
- Assignments 25%
  - 4 assignments
- Midterm 29th of June in class (20%)
- AEP 2%
- Final (40%)
  - You must be able to attend the final exam to pass the course!

Teaching Style

- **Motivate the students**. I feel it is the duty of the instructor to present the subject in a motivating and engaging manner.
  - **Ask a lot of questions**
  - **Get the students involved.**
  - Use I-clickers
- **Go Over Many Examples**
- **Set clear and realistic goals.** Students respond best to goals that are both challenging and achievable.
- **I would like all of you to be successful**
  - You are all competing with yourselves to do your personal best and I am here to help you with that
- **Final grade:** Normal distribution
- **Always respect the students.**

Questions and policies

- http://www.cs.sfu.ca/~hkhosrav/personal/py/120-2012.html

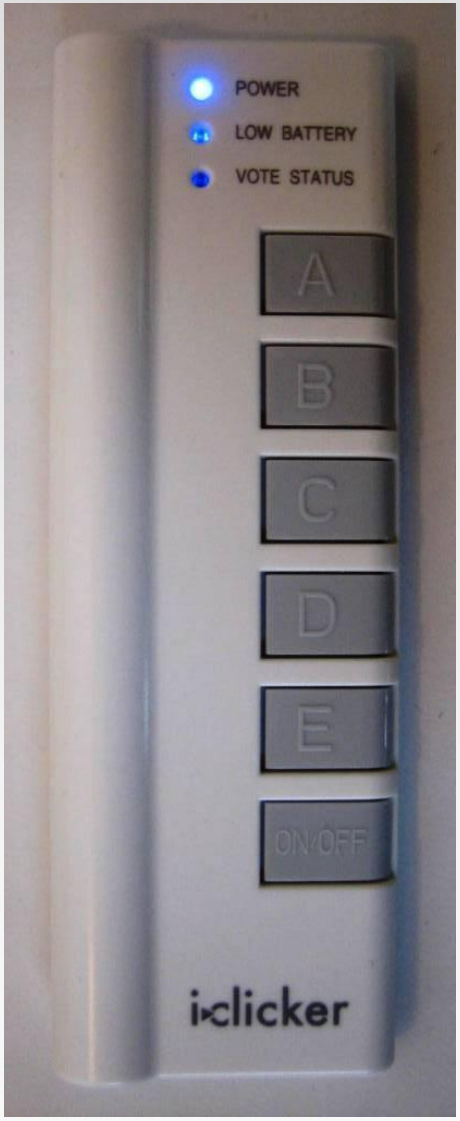
I-clickers

- http://www.iclicker.com/
- Enter your last name, first name, student ID (your
- email ID), and the clicker ID and click “submit.”

How do you vote?

- Turn on the clicker by pressing the “On/Off” button.
- A blue “Power” light will appear at the top of the remote.
- When I ask a question in class (and start the timer), select A, B, C, D, or E as your vote.

- Check your “Vote Status” Light:
- Green light = your vote was sent AND received.
- Red flashing light = you need to vote again.
  - **Not sure you saw the light? Just vote again.
  - **Want to change your vote? You can vote again as long as the timer is still going.



I-clicker Question

- You are

- A: First year student
- B:second year student
- C:Third year student
- D: fourth year student
- E: Other

I-clicker Question

- What is your major?

- A: Computer Science
- B: math or statistics
- C: Physics or Chemistry
- D: Business or engineering
- E: Other

I-clicker Question

- Which of the following best describes your experience with computers?

- A: I have seen computers and know how to turn them on
- B: I mostly use them for browsing internet and Google is my best friend. I have no programming experience.
- C: I have done some programming before, but not much.
- D: I think I'm a decent programmer.

I-clicker Question

- Which of the following best describes your expectations from the course

- A: I’m here to get the credits
- B: I’m here to see how I like programming. I may consider programming as a future career
- C:I want to learn programming professionally. I know that programming is my future career.
- D: I am all set to become the next Bill gates or Steve jobs

Content of CMPT 120

- Course Website is http://www.cs.sfu.ca/~hkhosrav/personal/py/120-2012.html
- About CMPT 120
  - Computer Science and Pseudocode
  - Programming Basics
    - Data types, User inputs
  - Control Structures
    - If statements, loops
  - Problem Solving
    - Data structures
      - Lists, strings, references
    - Algorithms
      - Searching, sorting, recursion
    - Working with Files
      - File input, file outputs

What is an Algorithm?

- The concept of an “algorithm” is fundamental to all of computing science and programming

- An algorithm is a set of instructions that can be used to solve a problem.

An everyday algorithm

- A baking recipe

1. Combine the room-temperature butter and the sugar. Mix until light and fluffy.
2. Add the eggs to the creamed butter and mix to combine.
3. In another bowl, combine the liquid ingredients and mix to combine.
4. Sift together the flour and other dry ingredients.
5. Alternately add the dry and liquid ingredients to the butter-egg mixture.
6. Mix just enough to combine.

What is an Algorithm?

- We are more interested in the kinds of algorithms that can be completed by computers.

- An algorithm definition accepted by most Computer Scientists
  - “An algorithm is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time”

What is an Algorithm?

- "An algorithm is a sequence of **unambiguous** instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time"

- Unambiguous: When you read an algorithm, there should be no question about what should be done.

1. Combine the **room-temperature** butter and the sugar. Mix until **light** and **fluffy**.
2. Add the eggs to the creamed butter and mix to **combine**.
3. In another **bowl**, combine the liquid ingredients and mix to **combine**.
4. **Sift together** the flour and **other dry ingredients**.
5. **Alternately add the dry and liquid ingredients** to the butter-egg mixture.
6. Mix **just enough** to combine.

What is an Algorithm?

- "An algorithm is a sequence of unambiguous instructions for **solving a problem**, i.e., for obtaining a required output for any **legitimate input** in a finite amount of time"

- Solving a Problem: An algorithm should always present a solution to a particular problem.
  - Our example: Using these ingredients, make muffins."

- Legitimate input:
  - An algorithm might need some kind of input to do its job.
  - In addition to having the inputs, they have to be "legitimate"
    - 1 can of baby corn, 1 cup orange juice; 1 telephone. We aren't going to get very far.
    - "legitimate" ingredients include sugar, eggs, flour and butter.

What is an Algorithm?

- "An algorithm is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a **finite amount of time**"

- Finite amount of time: The algorithm should finish eventually
  - A recipe that leaves us in the kitchen until the end of time isn't much good.
  - Stir with a fork until the mixture turns into Beef Wellington.

Data Structures

- A data structure describes how a program stores the data it's working with
  - To carry on with the cooking example
    - Most people have their recipes in cookbooks on a shelf
    - Recipes on index cards in a box (you might have to shuffle through the whole pile to find the one you want)
    - If pile is in some kind of order, e.g. alphabetical by the name of the dish it makes, you might be able to find the recipe much faster.
  - The way you choose to store information can have a big effect on the algorithm you need to work with it

What is Computing Science?

- Computing science is often defined as the study of algorithms, including
  - 1. Their formal and mathematical properties.
  - 2. Their hardware realizations.
  - 3. Their linguistic realizations.
  - 4. Their applications.

What is Computing Science?

- Their formal and mathematical properties:
  - What problems can be solved with algorithms
  - For what problems can we find solutions in a reasonable amount of time
  - Is it possible to build computers with different properties that would be able to solve more problems?”

- Their hardware realizations:
  - One of the goals when building computers is to make them fast.
    - Able to execute algorithms specified by the programmer quickly.
  - Make good use of their memory and be able to access other systems (disks, networks, printers, and so on).

What is Computing Science?

- Their linguistic realizations:
  - There are many ways to express algorithms so a computer can understand them.
  - Finding languages that are written by people and followed by computers.
  - Some “language” that can be understood by both people and computers.
- Their applications:
  - what actual useful things can be done algorithmically.
  - Is it possible for a computer to understand a conversation?
  - Can it drive a car?
  - Can the small computers in cell phones be made more useful?