CS 110 Computational Thinking I

Shah Jamal Alam, Jibran Rashid and Waqar Saleem

December 8, 2014

> "Begin at the beginning," the King said gravely, "and go on till you come to the end: then stop."
>
> — Lewis Carroll, Alice in Wonderland

This course will introduce you to the theory and practice of some of the fundamental ideas underpinning Computer Science. In the process, you will develop an appreciation of how Computer Science can be applied to solve everyday problems.

Outcomes

Outcomes are measured at one of three *levels of mastery* (adapted from [1]).

Familiarity Basic awareness of a concept. The student understands what the concept is or what it means.

Usage Ability to use or apply a concept in a concrete way, e.g. use it appropriately in a program, use a particular proof technique.

Assessment Ability to select an appropriate approach from understood alternatives. The student is able to consider a concept from multiple viewpoints and/or justify the selection of a particular approach to solve a problem.

On successful completion of the course, you will have the following abilities at the specified levels of mastery.

- To write programs of moderate compexity to solve problems. [Usage]
- Be familiar with common "types" and their supported operations. [Assessment]
- Understand the relation between variables and memory. [Usage]
- Be familiar with some common data structures. [Usage]

Lecture

Computer science is the study of computation-what can be computed and how to compute it. Computational thinking thus has the following characteristics[2]:

- *Conceptualizing, not programming.* Thinking like a computer scientist means thinking at multiple levels of abstraction.
- *Fundamental, not rote skill.* A fundamental skill is something every human being must know to function in modern society.
- *A way that humans, not computers, think.* Computational thinking is a way humans solve problems; it is not trying to get humans to think like computers.
- *Complements and combines mathematical and engineering thinking.* Computer Science is rooted in mathematics, which itself is established on the foundations of logic and set theory. Being free to build virtual worlds enables us to engineer systems beyond the physical world.
- *Ideas, not artifacts.* The computational concepts we use to approach and solve problems, manage our daily lives, and communicate and interact with other people.

Lab

For the lab component of the course, you will work on practical problems related to the theory covered in the lectures. Python will be used as the implementation language.

Table 1 shows the plan of activities. For each Lab activity, the instructor will first present the problem to be solved. In some cases, some supporting material and/or exercises may be assigned the previous week as a "Pre-lab" in order to aid the presentation. Feedback will be provided for submitted lab assignments in terms of a summary of class performance and common problems faced by the class. You will perform Lab assignments in groups and each group will have to make a presentation to the rest of the class for every assignments. The presentation will cover the group's progress on the assignment so far, any problems encountered, and solutions employed.

Grading

Labs:	30%	$(5 \times 10\%)$
Exams:	50%	$(2 \times 25\%)$
Assignments and Quizzes:	20%	

Self-Paced Assessment

The course allows the retaking of a previous exam when the student feels better prepared. There are two exams, both of which must be initially attempted within the semester on their respective dates and times. As a consequence, all students will complete course exam requirements by end of the semester.

If a student chooses to apply for exam retakes, then the student is awarded an incomplete at the end of the semester. One exam retake per month may be attempted during the subsequent semester. Retakes are conducted individually and are set up by mutual consent of the course faculty. For each month a retake schedule will be finalized at least one week before the exam retake date. Since CT-I is offered in the Fall semester, this option is available to students up to the end of summer break (based on faculty availability). The latest score on an exam replaces the previous score. Note that exam retakes do not influence scores for assignments, labs and quizzes taken during the semester.

The grades are finalized based on the retakes at the end of summer break. Students may request at any point after the end of CT-I to be assigned a final grade. The grade will be computed using the most recent exam scores and will be final. No retakes for any exam from CT-I are allowed afterwards.

W	Dates	Activity	Pre-lab (for next week)	Lecture
1	25 Aug to 29 Aug	Lab A: Sample Python programs Biggest number	Python installation Introduction to LaTeX	History of Computing Logic
2	1 Sep to 5 Sep	Lab B: Sample Python programs Gaussian elimination	L-systems	Proofs Assignment 1
3	8 Sep to 12 Sep	Lab 1: Fractals	–	Sets, functions and relations *Assignment 1 due*
4	15 Sep to 19 Sep	Progress presentations on Lab 1	Context Free Grammars	Abstraction and modularity Assignment 2
5	22 Sep to 26 Sep	*Lab 1 due* Lab 2: Expression Parser	–	Boolean logic Recursion *Assignment 2 due*
6	29 Sep to 3 Oct	*Feedback on Lab 1* Progress presentations on Lab 2	Entropy	Information Theory Assignment 3
7	6 Oct to 10 Oct	- - - - - - - - - - - - - - - - Eid-ul-Adha - - - - - - - - - - - - - - - -		
8	13 Oct to 17 Oct	*Lab 2 due* Lab 3: Compression	–	Data Analysis *Assignment 3 due*
9	20 Oct to 24 Oct	*Feedback on Lab 2* Progress presentations on Lab 3	Complex Networks	Data Structures Assignment 4
10	27 Oct to 31 Oct	*Lab 3 due* Lab 4: Data Analysis	–	Data Structures *Assignment 4 due*
11	3 Nov to 7 Nov	*Feedback on Lab 3* Progress presentations on Lab 4	The Game of Scrabble	Graphs
12	10 Nov to 14 Nov	*Lab 4 due* Lab 5: Single Player Scrabble	Citations in LaTeX	Graphs
13	17 Nov to 21 Nov	*Feedback on Lab 4* Progress presentations on Lab 5	–	–
14	24 Nov to 28 Nov	Progress presentations on Lab 5	–	Complexity
15	1 Dec to 4 Dec	Scrabble tournament	–	Complexity
16	8 Dec to 13 Dec	- - - - - - - - - - - - - - - - Exam Week - - - - - - - - - - - - - - - -		

Table 1: Original Course Plan.

W	Dates	Lecture	Lab	Assignments
1	25 Aug to 29 Aug	History of Computing Activity: Yohsin Beads The CS Interesting Test	Introduction to the course Lab A	Homework A
2	1 Sep to 5 Sep	Logic Activity: The Zebra Puzzle	Lab B: Functions and Conditionals	Homework B
3	8 Sep to 12 Sep	First Order Logic	Lab C: Conditionals and Strings	Homework C *Assignment 1*
4	15 Sep to 19 Sep	Induction Activity: Tower of Hanoi	Problem Set 1: Functions	Homework 1
5	22 Sep to 26 Sep	*Lab 1 due* Binary search Quiz 1	Lab 2: Binary Search Tree	*Assignment 1 due* *Problem Set 1 due* Homework 2: Fractals
6	29 Sep to 3 Oct	Induction Quiz 1a	Lab 3: Expression Evaluation	Lab 2 due Expression Parser
7	6 Oct to 10 Oct	- - - - - - - - - - - - - - - Eid-ul-Adha - - - - - - - - - - - - - - -		
8	13 Oct to 17 Oct	Networks	Network libraries	Lab 3 due
9	20 Oct to 24 Oct	Networks	Lab 4: Networks and Data Structures	Exam 1
10	27 Oct to 31 Oct	Compression	Feedback on Labs 1 and 2	
11	3 Nov to 7 Nov	Huffman Coding Activity: Monty Hall Problem	Progress on Labs 3 and 4	Lab 4 due
12	10 Nov to 14 Nov	Huffman Coding	Lab 5: Compression	*Assignment 2*
13	17 Nov to 21 Nov	Huffman Correctness Proof	Progress on Lab 5	Lab 5 due *Assignment 2 due*
14	24 Nov to 28 Nov	Bipartite Graphs Graph Isomorphism	Graph problems	Exam 2
15	1 Dec to 4 Dec	Review Activity: Kendo	Feedback on Labs 3 and 4	
16	8 Dec to 13 Dec	- - - - - - - - - - - - - - - Exam Week - - - - - - - - - - - - - - -		

Table 2: Actual Course Plan.

References

[1] ACM/IEEE-CS Joint Task Force, *Computer Science Curricula 2013*. Online. Accessed: 11-Dec-2013: `http://ai.stanford.edu/users/sahami/CS2013/`

[2] J. Wing, *A Vision for the 21st Century: Computational Thinking*. CACM Vol. 49, No. 3, March 2006. `http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf`