Computer Science Principles

Spring, 2017

Brad Smith

770-633-2312

bradanna@gmail.com

I. Course Description:

Computer Science Principles is a one-semester course introducing computer science basics and current career options with a focus on the fundamental building blocks of any computer language. This class will teach how to program in an efficient manner with quality, by learning principle programming concepts and putting them into action. This course will also explore and utilize "state of the art" development and collaboration tools used in major businesses specializing in computer science careers. Students will be expected to develop logical thinking and be able to collaborate with each other in a friendly, constructive manner.

Credit: 1/2 credit

II. Learning Objectives:

By the end of this course, the student should be able to:

1. Demonstrate and utilize current software development tools used in industry. (Github and Slack)

2. Develop and participate in collaborative team projects and evaluation of other students work.

3. Understand career options for Computer Science and take a field trip to an actual engineering office to see real world programming in action.

4. Develop logical thinking and viewing everyday tasks from a programming point of view.

5. Exhibit and learn computer programming basics, such as decisions, looping, data structures, hardware interaction, and basic logic operations.

6. Learn to be efficient in code and project design.

IV. Course Requirements:

- Github and Slack Collaboration accounts and tools, possible Pivotal Tracker as well
- Class Participation and review of classmates work
- Class programming Project
- Pair Programming Projects
- Weekly Topic Sheets
- Engineering Site Visit

Instructor Information:

Mr. Bradley Smith

2522 Shoals Bluff Ct

Buford, GA 30519

770-633-2312 (cell)

bradanna@gmail.com

Tentative Syllabus - Subject to Change at Instructor's Discretion and Class Pace

Date	Class Work
January 6 (1)	Introduction; Go over Github and Slack tools. Review Pre-class assignments. Talks about Hardware interface. Code guidlelines introduced Discuss function format. Source code management importance.
January 13 (2)	Introduce Agile and Waterfall Project Mangement Strategies. Introduce Class project, and decide management style. Continue Github/Slack. Begin True/False. Work on Logic And/Or and Number Systems.
January 20 (3)	Continue Logic, True/False, Number Systems. Introduce Decision code styles. Review Class project progress.
January 27 (4)	Review of all topics covered, continue decision code, introduce loops and repetition. Continue Class Project review.
February 3 (5)	Introduce Recursion. Assign Pair Programming project. Hardware impacts of recursion vs elegance.
February 10 (6)	Test Driven Development introduced, continue review of all concepts learned. Pair programming project time and Class project time. Introduce Arrays
February 17 (7)	Continue Array work, introduce sorting algorithms
February 24	NO CLASS (Winter Break)
March 3 (8)	Pair Programming Project due. Review all aspects done so far. Introduce Operating Systems and Interrupts.
March 10 (9)	Programming career options discussed. Continue repetition on fundamentals learned so far. Individual project assigned
March 17 (10)	Class Project Work time, review of all concepts. Individual project due.
March 24 (11)	Class project presentation and test. Review of all material, final evaluation.
March 31 (12)	Field Trip – Career Day

VI. Semester Assignments:

A. Github/Slack: We will use these common collaboration tools to communicate within the class, and turn in assignments. Because we will work as a software "team" each student's work will be available for review by all other students, and they are encouraged to kindly make suggestions and help out whenever a "teammate" is struggling. This is "real world" and is NOT meant to be judgmental, or ugly. Participation is required and will be the primary basis of grading the class.
B. Class programming project: A large scale project, designed to incorporate all students. The students will decide how to manage the project as a team, and how to break the project up into small objectives, which combined will complete the project.
C. Pair Programming Project: Students will be working with a partner on a smaller scale project, as is also common in real world programming environments.
D. Weekly Topic sheets: Worksheets with review items and exercises to complete for next class.
E. Engineering Site Visit: Visit to Brad's office, and exposure to a variety of software engineers and disciplines, in an effort to show what programming as a career looks like.