**CIS 120 Computer Programming I**

**Trinity College**

Spring 2014

Rolfing Computer Lab

Monday 6:30-9:30 PM

**Dan Fleming and Dr. Paul Bialek**  **Office:** Lew 105A

**Office Phone:** (847) 317-6988 **Hours**: M-F 8:30-4:30

**E-mail:** [dfleming@tiu.edu](mailto:dfleming@tiu.edu)

**Course Description**

CIS 120 introduces the students to computer programming and the modes of thinking necessary therein. This course will equip a student to read and write programs centered on the Python programming language. We will explore topics from basic syntax through algorithms and culminate the course with a large programming project. The students will be exposed to introductory computer science as a field. No prior programming experience is assumed; however, high school level algebra and a facility with computers is necessary.

**Faculty**

Dan Fleming (M. Div, M. A.) has worked as a freelance and contract programmer for various web startups. He is currently employed as a programmer/analyst for Trinity International University.

Dr. Paul Bialek is overseeing Dan in the teaching of this course.

**Objectives**

Upon completion of the course a student will be able to:

* Write functional code in Python.
* Read programs written in Python.
* Understand programming principles.
* Design and implement efficient algorithms.
* Document code properly to be read by third parties.
* Identify errors in code and debug appropriately.
* Design and implement an entire programming project.

**Textbooks**

Shaw, Zed A. *Learn Python the Hard Way.* 3rd Edition. New York: Addison-Wesley, 2014. Available for free (legally!) at learnpythonthehardway.org/book/

Guttag, John V. *Introduction to Computation and Programming using Python.*  (revised edition) Boston: MIT Press, 2013.

**Evaluation**

1. Homework (20%) – Homework will be due on Moodle by the beginning of each class session (6:30PM). Because homework will be discussed in class no late assignments will be accepted. Email and other formats will not be accepted, only what is submitted to Moodle.
2. Quizzes (10%) – Quizzes will be given at the beginning of each class session. 10 minutes (6:30-6:40) will be allowed for the quiz. Quizzes will focus on necessary vocabulary from the previous lecture and the readings due that day, unless otherwise noted by the instructor. Quizzes will also serve as attendance taking. The 2 lowest grades will be discarded. There are no make-ups on quizzes.
3. Class Participation (5%)
4. Midterm (25%) – The midterm will be a short answer exam. Students will show their ability to write and comment code. More information will come during lecture.
5. Final (40%) – The final will be a combination of a project and a presentation on that project. The student will devise a game that the computer can play against a user. Suggestions for the project include: Blackjack, Memory and Connect-Four. We will be doing Battleships as an in-class project. If you have another game you would like to try please talk to the instructor. There will be weekly assignments at the end of the semester to help with the project.

The presentation will be: run the game, play a few turns against the computer, and explain relevant portions of the code to the class. The class and instructor will then offer questions to the presenter. More information on the breakdown of grading for this project will be given in class.

THE RUNNING PROGRAM WILL BE DUE BEFORE THE FINAL CLASS PERIOD ON FRIDAY MAY 9 11:59PM VIA MOODLE. This is to give the instructor time to review the code before in-class presentations. Late projects will be deducted 10% per calendar day late. No projects will be accepted after 6:30PM Monday May 12.

**Integrity**

I have a zero tolerance policy for cheating. Cheating in a computer programming class includes, among other things, using code found on the web without attribution and turning in a classmate’s work as your own. Any instances of cheating will result in a 0 for that assignment, and may result in a failing grade for the course. Instances of cheating will also be referred to the Dean’s Office for action.

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Topics | Reading Due | Assignments Due |
| Jan 15 | Introduction |  |  |
| Jan 27 | Text and Files | Guttag 1, 2.1, | LPTHW 1-10 |
| Feb 3 | Logic | Guttag 2.1-2.4, 4 | LPTHW 11-21 |
| Feb 10 | Loops and Objects | Guttag 3 | LPTHW 22-31 |
| Feb 17 | Testing and Projects | Guttag 5-6 | LPTHW 32-42 |
| Feb 24 | Projects and Review | Guttag 7-8 | LPTHW 43-52 |
| Mar 3 | MIDTERM | MIDTERM | MIDTERM |
| Mar 17 | Understanding Complexity | Guttag 9 | Guttag Finger Exercises in 1-8 |
| Mar 24 | Search and Sort Algorithms | Guttag 10 | Guttag Finger Exercises 9-10 |
| Mar 31 | Math Math Math | Guttag 19 |  |
| Apr 7 | And Yet More Math |  |  |
| Apr 14 | Battleship Major Project Week 1 |  | Any 20 Project Euler problems NOT solved in class. |
| Apr 28 | Battleship Major Project Week 2 |  | Definition and Analysis of final Project |
| May 5 | Battleship Major Project Week 3 |  | Class structure code for Final Project. |
| May 12 | FINAL PRESENTATION |  | Final Project Due  **May 9 11:59PM** |