CSCI 105 Introduction to Computer Science

Fall 2019

MW 8:00-9:15am at Busn 209

Course Description

Introduction to computer hardware and software. Problem solving methods. Elementary concepts of algorithm development. C++ programming. Offered every year.

Instructor Information

Name: Shieu-Hong Lin
E-mail: shieu-hong.lin@biola.edu
Phone or extension: 5146
Office Hours: MW 1:00-3:00pm, T Th 11:00am-1:00pm
Office Location: Lim 137

Required Texts and Study Resources

Tony Gaddis. *Starting Out With C++ From Control Structures through Object.* Pearson; 9 edition.

Biola University Mission Statement

TRUTH. TRANSFORMATION. TESTIMONY. The mission of Biola University is biblically-centered education, scholarship, and service; equipping men and women in mind and character to impact the world for the Lord Jesus Christ.

Disability Services

Disability Services exists to assist any student who thinks he or she may need such assistance. Students desiring accommodations for this class on the basis of physical learning, psychological and/or emotional disabilities are to contact The Learning Center that houses both learning assistance and disability services. The Learning Center is located in the Biola Library, Upper Level, Room U-137, and this department can be reached by calling 562.906.4542 or by dialing extension #4542 if calling from on campus.

Program Learning Outcomes

This course is a core course required of all computer science majors designed to be taken within the first semester of the program. Successful completion of this course will prepare students to demonstrate a beginning proficiency toward the accomplishment of
PLO #1 Demonstrate the ability to analyze, model, and solve computing problems,
PLO #2 Apply and synthesize knowledge of the theory and practice of computing, and
PLO #3 Design and develop software programs integrated into functioning systems.

Course Objectives and Student Learning Outcomes

IDEA Objective #4: Developing specific skills, competencies, and points of view needed by professionals in the field most closely related to this course (Essential emphasis).

After successful completion of the course, students will be able to:

- demonstrate a solid understanding of fundamental concepts of computer programming and problem solving,
- use the Microsoft Visual C++ programming environment to develop C++ programs composed of multiple functions in multiple files,
- write C++ programs for problem solving using basic language features such as built-in data types, variable declaration, arithmetic operators, conditional statements, loops, functions, arrays, and structures, and
- develop a perspective of faith and learning through reflection on the essence of computer programming and its connection to intelligence and God's creation.

Class Policies

Class Attendance

You are expected to attend the class regularly since we will examine details of C++ programs using the computers in the lab. Missing the class may hamper your understanding of key concepts and programming skills critically needed in your programming assignments. Class attendance is counted for points for the weekly reading reports.

Assignments

- Assignments are expected to be electronically submitted under the Canvas system. Due dates are all on Wednesdays. The submission link under Canvas may remain open for 2 more days after the due date as a grace period.
- 1 point will be deducted for late submission within 2 days of the due date while the submission link is still open. You will receive no points after the submission on canvas is closed unless it is something like a serious health issue with statements from the doctor as proof.

Incomplete Grade

A temporary mark of "IN" (Incomplete Grade) will be issued in special cases when approved by the Associate Provost of Academic Administration for undergraduate students or the dean of the respective graduate school. "IN" grades course assignments are normally completed no later than five weeks after the end of the term. In the event of the inability of a student to complete the coursework by the approved deadline, the Office of the Registrar will assign the grade which the student has earned by the end of term. You can read more about Biola's policies and procedures regarding absences in the Undergraduate Student Handbook.

Computation of Final Grade

Provide a short statement or summary/chart listing the factors (tasks/assessments/assignments) that will contribute to the final grade for the course such as the following:

Factors Contributing to the Final Grade:

Assignment	Percentage
Weekly reading reports	10%
Weekly programming assignments	45%
Written tests / quizzes	30%
Programing test	15%
Faith and learning reflection	5%
Total Points:	105%

Grading Scale for the Course:

Grade	Percentage Range
A	93 – 100%
A-	90 – 92.99%
B+	87 – 89.99%
B	83 – 86.99%
B-	80 – 82.99%
C+	77 – 79.99%
C	73 – 76.99%
C-	70 – 72.99%
D+	67 – 69.99%
D	63 – 66.99%
D-	60 – 62.99%
F	0 – 59.99%

University Policies

Academic Honesty
Biola University is committed to ethical practice in teaching, scholarship, and service. As such, plagiarism and other forms of academic dishonesty will not be tolerated. To read more about Biola's policies and procedures regarding academic integrity, view https://studenthub.biola.edu/undergraduate-student-handbook-academic-integrity. (ONLINE ONLY: Replace last sentence with "You can read more about Biola's policies and procedures regarding academic integrity in the Undergraduate Student Handbook.") Another helpful resource is Plagiarism.org.

Confidentiality and Sexual Misconduct
As an instructor, one of my responsibilities is to help create a safe learning environment on our campus. I also have a responsibility in my role as a faculty member to share information I hear regarding sexual harassment, sexual assault, domestic violence, dating violence, stalking, sexual exploitation, and gender/sex-based discrimination with the Title IX Coordinator and/or the Campus Safety Response Team. Confidential resources available to students on campus include the Biola Counseling Center (562-903-4800) and the Student Health Center (562-903-4841). Both the Title IX Coordinator and the Campus Safety Response Team understand the sensitive nature of these situations and can provide

information about available on and off-campus resources, such as counseling and psychological services, medical treatment, academic support, university housing, safety measures and other forms of assistance. Additional resources and detailed information on the Non-Discrimination and Sexual Misconduct Policy is available online.

Non-Discrimination Policy
As Christian scholars we are keenly aware of the power of language, and believe in treating others with dignity. As such, it is important that our language be equitable and prejudice free. Good writing and speech do not make unsubstantiated or irrelevant generalizations about personal qualities such as age, disability, economic class, ethnicity, marital status parentage, political or religious beliefs, race, sex, or sexual orientation.
Avoid the use of stereotypes or terminology that demeans persons or groups based on age, disability, ethnicity, gender, race, language or national origin. Avoid drawing attention to irrelevant identifiers of race or gender. Avoid gender-specific language when referencing people in general. Avoid terms that assume the universality of human experience, and in particular presume the normativity of the socially dominant group.

Additional University and/or Department Policies
All university and departmental policies affecting student work, appeals, and grievances, as outlined in the Undergraduate Catalog and/or Department Handbook will apply, unless otherwise indicated in this syllabus.

Assignments and Tasks

Task 1: Weekly Reading and Progress Reports (15 assignments)
Due Date: Wednesdays
Weighting: 10%
Possible Points: 4 points.
Description: The student needs to report the effort made in the learning process, including (i) an estimate of time spent for the reading, (ii) whether the student Is able to answer 80% of the checkpoint questions, and (iii) whether the student came to the class this week. The student also needs to provide 1 to 2 short paragraphs of reflection on the reading as evidence of learning.
Assessment: For the effort part, the student is expected to (i) have attended the class this week at least once (0.5 point), and (ii) be able to answer at least 80% of the checkpoint questions in assigned reading or have spent at least 3 hours in the reading (1.5 points). For the reflection part, the student is expected to show substantial evidence of understanding or effort of trying to understand the contents in the reading .

Task 2: Weekly Programming Assignments (about 11 assignments)
Due Date: Wednesdays
Weighting: 45%
Possible Points: 6 points.

Description: The programming assignments form the backbone of the course. The student incrementally develops programming skills through these programming assignments using programming concepts learned in the class.
Assessment: The student needs to submit a self-evaluation report together with all the source code files for each assignment. In the self-evaluation report, the student should describe results from test

cases verified by a peer reviewer. The following are the **integrity rules** about programming assignments:

- **Peer discussion is encouraged**: Peer discussion is encouraged to cultivate an open learning environment in the class, but you should carefully read the guidelines below to avoid any dishonest behavior and never step over the guidelines explicitly described in the following.
- **Never use code written by others**: Any copy-and-paste of code from other people's programs or from websites is viewed as cheating and you will get 0 points for the assignment.
- **Never circulate your code**: Peer discussion of code shown on the screen or the board is acceptable for debugging purpose and for explanation of ideas. But you should never pass around your code (electronically or on paper) to others except for the TA and the instructor. Violating this rule is viewed as cheating in the class and the provider will receive 0 points for the assignment.
- **Never provide false or exaggerated results of test cases**: You need to describe results of test cases in the self-evaluation report**.** Providing false or exaggerated results of test cases in the report is viewed as cheating and you will receive 0 points for the assignment.
- **Demonstrate the credibility of your authorship of the work**: When you submit your code as your own work for points, you should make sure that you are able to explain your code and reconstruct your code from scratch without any outside help when requested. If you are not able to do that on your own when requested, you will get 0 points for the assignment and there will be an investigation.
- **Consequence of cheating in the class**: Cheatings end in 0 points for the assignments followed by discipline actions described in the student handbook.

Each programming assignment is graded in a 0-6 scale based on the following rubrics:

0. Nothing done **or missing the self-evaluation report or missing the integrity review** in the report.
1. Source code is completed but the code fails to compile successfully.
2. Source code can compile and do something required, but has serious bugs or miss a couple of key features.
3. Source code can compile and do most of the features required, but has many minor bugs or miss a key required feature.
4. Source code can compile and do all the features required, nearly fully functional, only a couple of minor bugs.
5. Source code can compile and do all the features required, fully functional, no bugs.
6. In addition to the points received above, get one more point if

   a. **the self-evaluation report provides description of test cases used (0.25 point)**, and
   b. **the test cases were verified by a peer reviewer (0.25 point)**, and
   c. **the source code is well indented and commented to make it visually very readable (0.5 point)**.

Task 3: Tests (Written tests and a programming test; may include quizzes)
Due Date: See the tentative course calendar below.
Weighting: 45% (30% for written tests/quizzes, 15% for the programming test)
Possible Points: 30-50 points for written tests.

Description: A written tests examines the understanding of the basic syntax and semantics of C++, while a programming test examines basic programming skills in C++.
Assessment A written test will be graded based on the written answers provided while a programming test will be graded based on rubrics comparable to those for the weekly programming assignments.

Task 4: Faith and learning reflection
Due Date: See the tentative course calendar below.
Weighting: 5%
Description: The student submits essays after reflection on faith and learning in computer science in the context of their programming experiences this semester.
Assessment: The essays will be graded according to the rubrics given for each reflection assignment.

Tentative Course Calendar

Date(s)	Topics Presented	Assignments Due
Weeks 1-2	Types, operators, expressions	Lab 1 Programming 0, 1 Reading 1, 2
Weeks 3-4	Logic and conditional statements	Programming 2, 3 Reading 3, 4
Weeks 5-6	Loops and iterations	Programming 4A, 4B Reading 5, 6
Weeks 7-8	Review + functions	Programming 5A, Midterm Reading 7, 8
Weeks 9-10	More on functions	Programming 5B, 5C Reading 9, 10
Weeks 11-13	Arrays, searching, and sorting	Programming 6A, 6B, 6C Reading 11, 12, 13
Weeks 14-15	Review + recursion, pointers	Nim game project as a programming test Reading 14, 15 Faith and Learning Integration Assignment 1, 2