SCMP 118: Introduction to Programming (with C++) Spring 2018
T & R 9:40-11 PRCL 109

Learning to program teaches you how to think. Computer science is a liberal art. (Steve Jobs)

Professor: Noah Aydin **Office:** RBH 319 **Phone:** 5674 **E-mail**: aydinn@kenyon.edu

Class web page: http://www2.kenyon.edu/depts/math/aydin/teach/118

Office Hours: See course web site for regular office hours and my weekly schedule. Also by appointment.

Textbook: Problem Solving with C++, 8th ed, by W. Savitch, Addison Wesley, ISBN: 978-0-13-216273-9

Software: We will be using the IDE CodeBlocks. You may also use Dev-C++ or NetBeans. See course web page for more details.

Content: This course presents an introduction to computer programming intended both for those who plan to take further courses in which a strong background in computation is desirable and for those who are interested in learning basic programming principles. The course will expose the student to a variety of applications where an algorithmic approach is natural and will include both numerical and non-numerical computation. The principles of program structure and style will be emphasized. Major concepts of chapters 1-8, 10-12, 14 and 15 in the textbook will be discussed.

Grades: Final grades will be determined based on the performance in the following components.

Component	**Points**
Weekly Programming Assignments	350
CodeLab	150
Quizzes	100
Midterm Exam	100
Attendance/Participation/Enthusiasm	25
History Essays	50
Final Exam	250
Total	**1025**

Daily Reading and Quizzes: You are expected to read the sections to be discussed for each class BEFORE the class. The format of the class will be mainly discussion, clarification and hands-on practice rather than lecturing. Not only should you read the material but also you should try out some of the programming concepts or examples on the computer (if you have a personal computer you can install the free software that we are using). If you come to class having read the material, you will be able to participate in class discussions which is part of your grade. To make sure you do the readings, we will have frequent, unannounced quizzes. They will be based on readings and the “Self Test Exercises”. Make sure you do these exercises as you read. Quizzes may include some basic questions that are not yet discussed in class that anybody who read the section should be able to do. On average, you should expect approximately one quiz per week. Some of lowest quiz scores will be dropped. See the course calendar page for reading assignments for each day.

CodeLab: CodeLab is a tool_ that develops coding skills and understanding through online coding exercises with immediate feedback. Most of the exercises are very short (one line). There will be more CodeLab exercises in the early part of the semester.

Weekly Programming Assignments (“Labs”): The only way to learn to speak a language is to speak it. The only way to learn programming is to program. Without any doubt, the programming assignments are the most important element of this course. There will be 11-12 programming projects over the course of the semester. I will talk in class about the particulars of turning them in electronically. Programming assignments must be submitted on time to be considered for credit. Make sure that whatever you submit for a grade is your own work that truly reflects your own understanding. We will watch for codes that are suspiciously similar and report such cases to the Academic Infractions Board.

Computer History Assignments: About 10 times during the semester, you will turn in a brief essay on some computer history fact. See course web page for more information.

Exams: Midterm- Tuesday, Feb 27 (week 7)
Final- Tuesday, May 8, 1:30 p.m., in the usual classroom.
Final exam will be comprehensive and 3 hours long (though you should not need all 3 hours)

Program Grading: All programs will be graded according to the following components.

Correctness: Each program should conform to specifications stated in the problem statement. A program should demonstrate correct handling of ordinary input, special cases and error conditions.

Design: Your programs should be modularized into coherent independent functions or classes with strong cohesion.

Style and Documentation: Your program should be easy to read and understand. This involves program indentation, modular design, variable names, user interface and comments.

NOTE: If a submitted program **fails to compile** it will be graded out of 50% of the total point value. If a submitted program has a **run-time error**, then it will be graded out of 75% of the original point value.

Late Policy: *No work will be accepted late*. You need to submit your labs in your folder in *P drive*. Be aware that the time and date is attached to your work once it is submitted. If the file is modified anytime after the submission, the date and time change accordingly. **Do not open your submitted files after the due date**, until graded.

Attendance: Regular attendance and participation in class discussions and activities is expected. No make-ups for exams will be given without justified and documented excuses. No make up for quizzes (except for long-term extenuating circumstances). Each unexcused absence will hurt your participation grade. Not missing any classes without a valid excuse is a necessary but not sufficient condition for a full credit on this component. You need to enthusiastically participate in class discussions and activities as well.

Academic Honesty: At Kenyon we expect all students, at all times, to submit work that represents the highest standards of academic integrity. It is the responsibility of each student to learn and practice the proper ways of documenting and acknowledging those whose ideas and words they have drawn upon (see Academic Honesty and Questions of Plagiarism in the Course Catalog). Ignorance and carelessness are not excuses for academic dishonesty. You may discuss lab projects with others, but *the final submitted work must be done independently and should reflect your own understanding*. That is, after discussing the problems with others (or the tutor), sit down and write your own code. Violation of this rule will result in a report to AIB (Academic Infractions Board) .If you are uncertain about the expectations for this class, please talk to me for clarification.

Disabilities: If you have a disability that requires an accommodation in this class, please feel free to discuss your concerns with me as soon as possible. Also, you are required to register for support services with the Office of Disability Services in the Olin Library, Center for Innovative Pedagogy. Please contact Erin Salva at 5453 or email salvae@kenyon.edu. Though I am happy to help you in any way I can, I cannot make any accommodations for learning (or other) disabilities without proper authorization from Ms. Salva**.**

Statement on Title XI

Kenyon College seeks to provide an environment that is free of gender bias, discrimination, and harassment. If you have been the victim of sexual harassment/misconduct/assault, interpersonal violence, or stalking we encourage you to report this. If you report this to a faculty member, she or he must notify Kenyon's Title IX coordinator of any information about the incident that you provide. Kenyon College's Title IX and VAWA Policy is available at: http://www.kenyon.edu/directories/offices-services/title-ix/policy/

How to Start Using CodeLab

The first 10 exercises in CodeLab are free so you can start doing the CodeLab exercises before paying. But first you must register to get a username and password and put yourself on the class roster.

REGISTRATION:

1) Go to www.tcgo1.com OR www.tcgo2.com
2) Click "Register for CodeLab"
3) When prompted enter the Section Access Code: KENY-26127-PHBE-35

LOGIN:

1) Go to www.tcgo1.com OR www.tcgo2.com
2) Click "Login to CodeLab"

GETTING FULL ACCESS:

log in to CodeLab, click LOBBY, click the button "Get Full Access", follow the directions