1. Basic Course Information

1.1 Objectives:

This course covers intermediate-level C++ programming and the software development issues that arise in practical programming projects. Topics include C++ syntax and semantics, principles of design, and basic software engineering skills. A heavy emphasis is placed on the development of abstract data types as the basic modules or building blocks of a well-designed program.

1.2 Meeting Times

Lectures

- MW 3:00-4:15 (ocnps 0100)
- MW 4:20-5:35 (ocnps 0100)

Labs

- M 19:10-21:50 (dragas 1115)
- T 13:30-16:10 (dragas 1115)
- T 16:30-19:00 (dragas 1105)
- Th 13:30-16:10 (dragas 1105)

Recitations

- M 12:00-12:50 (dragas 1115)
- T 19:10-20:00 (dragas 1105)
- W 17:45-18:35 (dragas 1105)

All students must register for a lecture, lab, and recitation. Students are expected to attend the lecture and lab session for which they have registered, but may attend any recitation sessions.

1.3 Instructor

Dr. Ayman Elmesalami

Contact:

Office: Dragas, room 1100B
Email: aelmesal@cs.odu.edu
Office hours:
- Tues : 12:00-1:00 CRN14072
- Thurs: 11:00-12:00 CRN10606

1.4 General Organization of the Course

The course is divided into several topics. Each topic is addressed by lectures and textbook readings and accompanied by a variety of activities including:

- **Labs** - ungraded activities that introduce or practice techniques that you will use in the assignment
- **Assignments** - graded activities, most of which involve programming
• **Unix Assignments** - references back to the co-requisite course CS252. These indicate that you should have completed the CS252 curriculum up to and including the listed assignment, because you will need or, at least, might benefit from, the skills taught in that course.

If you have already taken and passed CS252 in a prior semester, you will not need to (and not be able to) actually repeat these assignments. But it may be a good idea to return to the CS252 website and review the lecture notes related to the listed assignment.

• There will be a **midterm exam** and **final exam**.

• There will also be a semester "term" **project** in which you will apply the techniques of design, coding, testing, and debugging to a larger problem than is tackled in the assignments.

1.5 **Required Text:**


• This book is frequently sold with accompanying CDs. We won’t use them, so don’t worry about it. If you can get a used copy that is missing the CDs, or if you can get a better price on a new one without the CDs, go for it!

2. **Course Pre- and Co-requisites**

The prerequisites for this course are:

• CS 150, Problem Solving and Programming I
• Math 163

or equivalents.

The co-requisite for this course is:

• CS 252, Introduction to Unix for Programmers

(A co-requisite is a course that must be taken before or during the same semester as this course.)

3. **Assignments**

Assignments for this course will include "weekly" assignments and a term project. Both may involve programming in C++

3.1 **C++ compiler**

The “official” compiler for this course is the Free Software Foundation’s **g++** (also known as **gcc** or **GNU CC**), **version 4.8.1** or higher. This is the compiler that the instructor and/or grader will use in evaluating and grading projects. If you have access to other compilers, you may use them, but you are responsible for making sure that their projects can be compiled by the instructor and/or the course’s grader using the official compiler.

You may want to develop your programs on the most convenient compiler and then port it over to the official environment. Please don’t underestimate the amount of time that may be involved in coping with subtle differences among compilers.

You can do all work in this course using **g++** on the CS Dept Unix servers via ssh/X or via the CS Dept’s Virtual PC Lab. If you like, however, you can obtain the g++ compiler for free from a variety sources.
3.2 Computer Access:

Students will need an account on the CS Dept. Unix network to participate in this class. This account is unrelated to any University-wide account you may have from the ODU’s computing services (OCCS). If you have had a CS Unix account in the recent past, you should find it still active with your login name, password, and files unchanged. If you have had an account and it has not been restored, contact the CS Dept systems staff in the lab in Dragas Hall, Room 1111K or email root@cs.odu.edu requesting that it be restored. If you do not yet have such an account, follow the directions provided in the pdf file (Account Setup) to get set up.

Please note that, new account creation for students enrolled in a future semester becomes available about one week before the start of that semester.

4. Exams

- **Midterm**
  - will be available on Blackboard in **week 9** (*October 19th – October 23rd*) of the semester
  - You will write the exam **in the lab during your regular lab time of this week**

- **Final exam**
  - will be available on Blackboard from **December 5th** to **December 11th**
  - You will write the exam **as in the final exam dates, which is posted on Blackboard**.
  - Please note that the final exam is cumulative

5. Course Policies

5.1 Assignments and Grading

Assignments will be turned in through the Blackboard. Late submissions of assignments or of the portions of the semester project and make-up exams will not normally be permitted.

Exceptions will be made only in situations of unusual and unforeseeable circumstances beyond the student’s control, and such arrangements must be made prior to the due date in any situations where the conflict is foreseeable.

“I’ve fallen behind and can’t catch up”, “I’m having a busier semester than I expected”, or “I registered for too many classes this semester” are not grounds for an extension. Extensions to due dates will not be granted simply to allow "porting" from one system to another. "But I had it working on my home PC!" is not an acceptable excuse also.

5.2 Academic Honesty

Everything turned in for grading in this course must be your own work. The instructor reserves the right to question a student orally or in writing and to use his evaluation of the student’s understanding of the assignment and of the submitted solution as evidence of cheating. Violations will be reported to the Office of Student Conduct & Academic Integrity for consideration for possible punitive action.

Students who contribute to violations by sharing their code/designs with others may be subject to the same penalties. Students are expected to use standard UNIX protection mechanisms (chmod) to keep their assignments from being read by classmates. Failure to do so will result in grade penalties, at least.

This policy is not intended to prevent students from providing legitimate assistance to one another. Students are encouraged to seek/provide one another aid in learning to use the operating system, in issues pertaining to the programming language, or to general issues relating to the course subject matter.

Students should avoid, however, explicit discussion of approaches to solving a particular programming assignment, and under no circumstances should students show one another their code for an ongoing assignment, nor discuss such code in detail.
5.3 Grading:

Assignments:  40%
Semester Project:  20%
Midterm Exam:  15%
Final Exam:  25%

---------------------------------------
Total:  100%

Accessibility information: The accessibility information can be found at: https://www.odu.edu/educationalaccessibility