CS 3005: Programming in C++

Spring 2019 Syllabus

Course Description

For students pursuing degrees in Computer Science and Computer and Information Technologies, or any student with a strong interest in computer programming. Covers syntax and semantics of the C++ programming language through completion of hands-on projects. The student must already be fluent in some other programming language.

Prerequisites

CS 1410 (Grade C- or higher)

Course fees

Course fee: $25, used to assist in maintaining CIT infrastructure.

Disability Statement

If you suspect or are aware that you have a disability that may affect your success in the course you are strongly encouraged to contact the Disability Resource Center (DRC) located in the North Plaza Building. The disability will be evaluated and eligible students will receive assistance in obtaining reasonable accommodations. Phone # 435-652-7516.

Sections

1. MWF 8:00 - 8:50 am in Smith 116
   Final exam Apr 29 at 07:00 - 08:50 am

Instructor

Curtis Larsen

Objectives

The student learning outcomes for this course are:

- Construct computer programs in a statically typed language.
- Construct computer programs in C++, using functions, classes and STL elements.
- Construct computer programs using stack, heap and static memory.
- Use and construct unit tests.
- Use version control to manage code.
- Use memory checking and debugging tools.
- Create programs with more complexity than in previous course work.

The following C++ language constructs will be discussed:

- C++ syntax
- Typed variables and values
- Pointers and references
- Dynamic, stack and static memory
- C++ functions
- C++ classes
- Polymorphism
- Namespaces
- Making a class robust
- const, in all its flavors
- Template functions and classes
- Standard template library (STL)
- Recursion
- Exceptions
Resources

Textbook

There is no required text for the course. However, reference sources for C++, the STL and C will be very helpful.

1. *Online References* [C++ Reference](#) [C++ Tutorial](#)


Other Resources

- [Notes for installing g++](#)
- [Notes for installing Google Test](#)
- [Notes for editing in Windows and using build tools in bash](#)
- [Notes for getting started with git](#)
- [Linux: Create, Compile, Link, Run Simple Program](#) (3:23)
- [Linux: Create, Compile, Link, Run Multiple File Program](#) (9:45)

Computer Labs

You may use the computers and software in the Smith Computer Center. Some lab assistants may be able to help with assignments and pass off homework assignments for introductory courses.

Course Web Site

Assignment submissions and grades will be managed in the [Canvas System](#).

Assignments and Exams

Reading

While there is no required text book. Students are expected to find and read relevant references and example programs.

Assignments

A series of programming assignments will be required. These assignments are due on the date and time listed in the schedule, or as stated by the instructor during class. It cannot be over emphasized that it is important to start early and get all of your assignments done before their due dates. Programs that do not run correctly will receive no credit.

Parts of some programming assignments will be completed inside Code Grinder. Parts or all of most assignments will be completed using the GNU tool chain. All non-Code Grinder assignment submissions will require compatibility with [g++](#), the 2011 C++ standard and a `Makefile` submission.

Some students will use other development environments such as XCode, Visual Studio or C-Lion. Submissions will still be required to build and run using the GNU tool chain.

The source code for programming assignments may be submitted to the [Canvas](#) assignment page, or through commits to a git repository, depending on the assignment.

Assignment submissions must be correct to receive credit. Feedback will be given on assignments submitted by the due date. Assignments submitted near the submission-cutoff date, may not received feedback.

Participation

You will be required to give positive contributions to class discussions.

Exams
There will be a midterm and a final. Test tasks will be extensions of homework assignments and small new programs. Be sure to understand your assignments and keep backups of all your work.

**Grading**

Letter grades will be assigned based on the percentage of total possible points attained. The point totals will be approximately: assignments 140, participation 20, midterm exam 100 and final exam 200. The following chart will be used for letter grades.

<table>
<thead>
<tr>
<th>Minimum Percentage</th>
<th>Letter Grade</th>
<th>Minimum Percentage</th>
<th>Letter Grade</th>
<th>Minimum Percentage</th>
<th>Letter Grade</th>
<th>Minimum Percentage</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>94</td>
<td>A</td>
<td>84</td>
<td>B</td>
<td>74</td>
<td>C</td>
<td>64</td>
<td>D</td>
</tr>
<tr>
<td>90</td>
<td>A-</td>
<td>80</td>
<td>B-</td>
<td>70</td>
<td>C-</td>
<td>60</td>
<td>D-</td>
</tr>
<tr>
<td>87</td>
<td>B+</td>
<td>77</td>
<td>C+</td>
<td>67</td>
<td>D+</td>
<td>0</td>
<td>F</td>
</tr>
</tbody>
</table>

**Course Policies**

**Attendance**

Students are responsible for material covered and announcements made in class. School-related absences may be made up only if prior arrangements are made. The class schedule presented is approximate. The instructor reserves the right to modify the schedule according to class needs. Changes will be announced in class and posted to the website. Exams and quizzes cannot be made up unless arrangements are made prior to the scheduled time.

Occasional absences are acceptable as long as the student keeps up with assignment work. Students who miss more than two consecutive weeks of class or who miss more than 20% of scheduled classes during the semester without making prior arrangements will receive a failing grade. Students who miss any scheduled exam (including midterm exams and the final exam) or fail to complete a final project without making prior arrangements will receive a failing grade.

**Time Commitment**

Courses should require about 45 hours of work per credit hour of class. This class will require about 135 hours of work on the part of the student to achieve a passing grade, which is approximately 9 hours per week. If you do not have the time to spend on this course, you should probably rethink your schedule.

**Late Policy**

Each assignment has a due date and a submission-cutoff date. The due date is the required date. The submission-cutoff date is to allow students to correct small problems discovered during pass-off, or to allow for minor interruptions to personal schedules. Late work will not be accepted after the submission-cutoff date.

**Collaboration**

Limited collaboration with other students in the course is permitted. Students may seek help learning concepts and developing programming skills from whatever sources they have available, and are encouraged to do so. Collaboration on assignments, however, must be confined to course instructors, lab assistants, and other students in the course. Students are free to discuss strategies for solving programming assignments with each other, but this must not extend to the level of programming code. Each student must code his/her own solution to each assignment. See the section on cheating.

**Cheating**

Cheating will not be tolerated, and will result in a failing grade for the students involved as well as possible disciplinary action from the college. Cheating includes, but is not limited to, turning in homework assignments that are not the student’s own work. It is okay to seek help from others and from reference materials, but only if you learn the material. As a general rule, if you cannot delete your assignment, start over, and re-create it successfully without further help, then your homework is not considered your own work.

You are encouraged to work in groups while studying for tests, discussing class lectures, discussing algorithms for homework solutions, and helping each other identify errors in your homework solutions. If you are unsure if collaboration is appropriate, contact the instructor. Also, note exactly what you did. If your
actions are determined to be inappropriate, the response will be much more favorable if you are honest and complete in your disclosure.

Where collaboration is permitted, each student must still create and type in his/her own solution. Any kind of copying and pasting is *not* okay. If you need help understanding concepts, get it from the instructor or fellow classmates, but never copy another’s code or written work, either electronically or visually. The line between collaborating and cheating is generally one of language: talking about solutions in English or other natural languages is usually okay, while discussions that take place in programming languages are usually not okay. It is a good idea to wait at least 30 minutes after any discussion to start your independent write-up. This will help you commit what you have learned to long-term memory as well as help to avoid crossing the line to cheating.

**College Policies**

Click on this link: [https://academics.dixie.edu/syllabus/](https://academics.dixie.edu/syllabus/) for comprehensive information on the Semester Dates, the Final Exam Schedule, University resources such as the library, Disability Resource Center, IT Student Help Desk, Online Writing Lab, Testing Center, Tutoring Center, Wellness Center and Writing Center. In addition, please review DSU policies and statements with regards to Academic Integrity, Disruptive Behavior and Absences related to university functions.

If you are a student with a medical, psychological, or learning disability or think you might have a disability and would like accommodations, contact the Disability Resource Center (652-7516) in the North Plaza. The Disability Resource Center ([http://dixie.edu/drcenter/](http://dixie.edu/drcenter/)) will determine eligibility of the student requesting special services and determine the appropriate accommodations related to their disability.