CMSC 240: SOFTWARE SYSTEMS DEVELOPMENT  
SUNAY 2018

Lecture: M 13:30–14:45, Jepson G20  
Lab: W 13:30–14:20, Jepson G20

Office Hours: T 14:00–15:00, W 15:00–16:00, or by appointment

Instructor: Barry Lawson  
223 Jepson Hall  
blawson@richmond.edu

Web Page: http://www.mathcs.richmond.edu/~blawson/cmsc240/

Texts: The following texts are available via the UR Library's Safari Books Online subscription — you may access them via links on the course Web page.

- A Tour of C++, B. Stroustrup  
- The C++ Programming Language (4th ed.), B. Stroustrup  
- Head First Object-Oriented Analysis and Design, B. McLaughlin, G. Pollice and D. West.  
- The Art of Debugging with GDB and DDD, Matloff and Salzman. (suggested reading)

Other texts are available free of charge via Safari Books Online through the library's online search. I encourage you to explore and find books and learning resources that work well for you.

Course Description:
In CMSC 150 and 221, you have developed a solid set of programming skills and a good background in the Java programming language. As you continue with your CS major or minor, you will encounter problems that are larger in scale than the typical programs you have written, and that may have requirements that are not so easily met in the Java language. In this course, you will enhance your skill set by learning the C++ programming language (useful in CMSC 301 and several other upper level courses as well as for many real-world problems) and get an introduction to a number of tools and strategies that will prove useful as you tackle larger and more complex problems. We will talk about design strategies and object-oriented techniques that ease the construction of larger scale programs, and you will reinforce your skills in systematic testing and debugging of software. In the latter half of the semester, you will work in a group on a sizeable project, going from design through implementation. The course is focused on developing useful, professional-quality software, a worthwhile goal whether you eventually write software for a living or use programming as a tool in pursuit of some other goal.

Assignments: This course is “about doing”. Accordingly, the assignments will consist of weekly lab/homework assignments, a few individual programming projects, and one substantial group programming project.

Notes on Program Submissions:

- Programs that do not compile will by default receive no credit.  
- Programs with no or minimal commenting will automatically receive a letter grade deduction.
- Follow specifications carefully and precisely. This will require you to read the assignment carefully.  
- Give meaningful usage/error messages.

"Incorrect number of files." is not very helpful.

"Usage: ./problem1 [input file name] [output file name]" is clear and helpful.
- Be consistent with indentation.
- Be generous with comments and with whitespace.
- When I mention example programs in the specifications, it is in your best interests to go investigate those programs, rather than just rely on something you might find on Stack Overflow. For example, there is an approach I want you to use for reading from files, which will be available in a provided example program. Follow that as a guide.
Web Page and Email: I will use the course Web page and email for assignment-related information. It is your responsibility to check both frequently.

http://www.mathcs.richmond.edu/~blawson/cmsc240/

Grading Policy: Final letter grades will be assigned per the traditional 10-point scale (≥ 90% is at least an A--; < 90% but ≥ 80% is at least a B--; etc.), according to the following percentages:

<table>
<thead>
<tr>
<th>Assignment Type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labs &amp; Homework Assignments</td>
<td>40%</td>
</tr>
<tr>
<td>Individual programming assignments</td>
<td>30%</td>
</tr>
<tr>
<td>Group programming assignment</td>
<td>30%</td>
</tr>
</tbody>
</table>

Attendance Policy:

- You are expected to attend each class period and each lab session for its duration. If you must miss a class or lab, you are responsible for any associated material. If there is a class or lab that you must miss, please inform me in advance.
- Any student with an excessive number of absences is subject to a failing grade of V.

Honor Code:

- Unless provided to you by the instructor of this course, you are not permitted to view or use in any way existing assignments, tests, or solutions in any form, whether they be from a previous offering of this or another course or Internet-available.
- Lab assignments and homework assignments may be discussed with others, but are subject to the empty hands policy: You may freely discuss ideas with other students, but each student must leave the discussion without any written or otherwise recorded material.
- You may not work directly with any other student on the completion of individual programming assignments. Any manifestation of copying another student’s work for your own (whether digital, hand-written, oral, etc.) or working together on an assignment is not permitted — this includes, for example, looking at another student’s implementation and then writing “your own” version of that implementation.
- You may not in any way share or sell any of the assignments or materials for this course. The course materials are my creation and belong to me, and on certain assignments I have borrowed (with permission) materials created by other instructors, with appropriate attribution.
- Failure to comply with any of these policies will be treated as an Honor Code violation.

Special Notes:

If you are allowed academic accommodations, or if your desire to observe a religious holiday presents a conflict with class activities, please contact me as soon as possible.

Course Outline: Following is a basic list of topics to be covered this semester, not necessarily in the order presented below. Additional topics may be added as necessary.

- C++ language introduction
- Generic programming
- Working with standard libraries
- Systematic testing and debugging
- Software design concepts
- Object-oriented design
- Software development tools
- Group project

Important Dates:

At this point, I do not intend to have tests or a final exam in this course, focusing instead on the project work. I reserve the right to modify this decision.

Spring Break: Sat 10 Mar – Sun 18 Mar