

CS246: Object-Oriented Software Development

Fall 2012

Prof. Reid Holmes

[Some slides from Mike Godfrey]

I am ...

- Prof. Reid Holmes
- email: rtholmes@cs.uwaterloo.ca
 - URL: <http://www.uwaterloo.ca/~rtholmes>
 - Office: DC3551, hours TBD (send email)
- Please go through the course tutor for questions about assgts, grading, etc
 - He may redirect you to me
 - We are sharing assgts with all four lecture sections of CS246, so his answers will be more “official”
- See me for lecture and admin related questions

Other contact / schedule info

- Tutor is Kaiyu (Kevin) Wu
 - Email: cs246tutor@student.cs.uwaterloo.ca
 - Office: MC 4065 (CS consulting centre)
 - Office Hours: TBD
 - **Kevin is your first point of contact for questions**
- Tutorials (multiple sections, given by course staff)
 - Please attend!
 - NO TUTORIALS IN FIRST WEEK

Other contact / schedule info

- Course web page (check regularly for announcements, etc):
 - <https://www.student.cs.uwaterloo.ca/~cs246/>
- We'll use Piazza for discussions
 - Link from CS246 webpage
- We'll use Marmoset for (most of) the assignment submissions
 - We may use MarkUs too

Expectations

- I expect:
 - You will come to lecture, pay attention, take notes
 - You will go to tutorial
- You can expect:
 - About 5 / 6 assgts (first one trivial, last one bigger)
 - One mid-term, one final exam

Marking breakdown

40%	Exam
35%	Assgts
25%	Mid-term

- You must obtain a passing grade ($\geq 50\%$) for the weighted grade of the midterm and the final exam to pass the course

Lateness, cheating, etc

- Don't be late
 - Late submissions get 0
- Usual UW policies on illness, missed exams/assgts, etc. (see web)
- Usual UW policies on cheating, academic integrity (see web)
 - Remember that the Piazza is semi-public, so don't share secrets
 - But it's OK to quote stuff said by course staff
 - I am an international expert in plagiarism detection of software. Really. Don't mess with me 😊

Working in groups

- Don't work in groups
 - ... unless we tell you to
- All group members receive the same grade, period
 - Course staff will not arbitrate group disputes
- You can do a group assgt on your own if you really want to, but we don't recommend it
 - Working in groups is good for you, and like the real world
 - It's lots of work, and you don't get any special credit

Assignment remarking

- Attach a sheet of paper to your assignment clearly stating the questions that you want remarked plus any supporting evidence
- No more than two weeks after the assignment is handed back, return your assignment with the remarking page to the tutor during a tutorial or an office hour (not in lecture!)
- The entire assignment will be examined:
 - You mark could go down (or up, or stay the same)

CS246: Object-Oriented Software Development

- Course Description:
 - Introduction to object-oriented programming and to tools and techniques for software development.
 - Designing, coding, debugging, testing, and documenting medium-sized programs.
 - Prerequisites:
 - CS 145 taken Fall 2010 or earlier, or CS 146, or
 - a grade of 60% or higher in CS 136 or 138;
 - Honours Mathematics or Software Engineering students only.
 - Anti-requisites: GENE 342, SYDE 322/221.

Students with disabilities

- The Office for Persons with Disabilities (OPD), located in Needles Hall, Room 1132, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum.
- If you require academic accommodations to lessen the impact of your disability, please register with the OPD at the beginning of each academic term.

CS246: Main topics / goals

- Reading requirements specifications and designing software to implement them,
- Selecting appropriate data structures and control structures,
- Writing reusable code,
- Reusing existing code,
- Analyzing basic performance issues,
- Using debuggers, and
- Creating test suites.

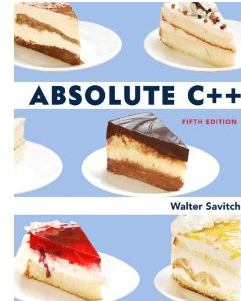
Lecture hours by topic

- **The (Unix) shell (4 hours)**
 - File system, pattern matching, quoting, shell/system commands, file permission, redirection, shell programming.
- **OO programming using C++ (16 hours)**
 - Declarations, expressions, control structures, structured programming, pre-processor, I/O, dynamic allocation, objects, overloading, inheritance, templates, STL, separate compilation.
- **Unix tools (8 hours)**
 - Compiler, debugging and the debugger (e.g., GDB), code management (e.g., make), version control (e.g., SVN).
- **Software engineering (8 hours)**
 - Development process, design (UML), language selection, patterns, testing.

Course lecture notes

- Official lecture notes by Peter Buhr
 - PDF available for download from course web page
 - Slides version is virtually identical content
- I will be using a mix of:
 - Buhr's slides (Unix shell, Unix tools)
 - Savitch's slides (C++ content)
 - My own slides and additions (everything)

Course text



- Walter Savitch, “Absolute C++”, 5th Ed., Addison-Wesley
 - Older editions OK too
- The text is recommended but not required
 - Really, tho, you should own a good intro text for C++, and this is as good as any

Course lecture slides

- I will be maintaining a web page of the extra (non-Buhr) slides I use here:
<https://cs.uwaterloo.ca/~rtholmes/teaching/2012fall/cs246/>
- Savitch's slides can't be put on the open web, so they've been password protected:
 - Login: “cs246”, password:
 - Please don't put the password in a public place

Software

- C++ and Marmoset, usually
 - By default, we assume you use “g++” as your compiler
 - g++ is the C++ compiler from the GCC suite
 - You may recall that their C compiler is called “gcc”
 - Free (beer/speech) download from <http://gcc.gnu.org>
 - Runs (at the command line) on MacOS, Linux, and even Windows
 - But installation is harder
 - You can use whatever you like, as long as you can get it to pass thru Marmoset (which uses g++)
 - If you pick VisualStudio, it’s up to you to make sure it runs correctly under g++ before you submit!

Assgt 1

- ... is out (see web page)
- ... is trivial
- ... should be completed as soon as possible

Some career advice

1. I highly recommend getting used to the Unix command line, if you aren’t already!
 - MacOS has a (real) Unix command line; use “Terminal” in Applications/Utilities
 - Or you can download the cooler iTerm app
 - Windows has a fake Unix command line, called CygWin, which is a free download, but non-trivial to use
 - Virtual Box + Linux is probably the best way for Windows users to go
2. I also strongly recommend not getting hung up on text editors; use what is right for your task
 - Sometimes vim, sometimes emacs, sometimes nano, sometimes an IDE

Quick non-scientific survey

- How many people use Windows / MacOS / Linux as their preferred OS for programming?
 - Have you ever installed / maintained your own Linux box?
- If I asked you to write a binary search, what language would you choose?
- What’s your preferred programming environment?
 - notepad, vim, emacs, MS Developer studio, Eclipse, other?