CS 246 Object-Oriented Software Development Fall 2015

Course Description

Introduction to object-oriented programming and to tools and techniques for software development. Designing, coding, debugging, testing, and documenting medium-sized programs: reading specifications and designing software to implement them; selecting appropriate data structures and control structures; writing reusable code; reusing existing code; basic performance issues; debuggers; test suites.

Course Objectives

- Design, implement, test, and debug C++ programs to solve problems requiring hundreds of lines of code, making appropriate use of
  - types, variables, arrays, strings, and dynamic memory; loops, conditionals, and other control structures; structures, unions, and enumerations; procedures and functions; the preprocessor; formatted and unformatted I/O;
  - classes, objects, overloading, and single inheritance;
  - a subset of the STL, including vector, list and map;
  - assertions and exceptions;
  - basic software development tools, including makefiles, a shell, a revision control system, and a debugger;
  - test suites for unit testing, white and black box testing;
  - structured programming, incremental development;
  - interface design, abstractions, information hiding, cohesion and coupling;
  - a subset of UML to specify classes, objects and relationships between them; and
  - a selection of design patterns.
- Explain the following properties of the memory model used in C++, including their impact on time and space efficiency when designing code: bytes vs. words, memory as an array, run-time stack and stack frames, memory allocation on the heap vs. automatic allocation on the stack, pointers as memory addresses, the representation of objects in memory.
- Define and explain at an elementary level basic software-engineering concepts, including the waterfall model and other development methodologies.

Schedule

Subject	Catalog#	Units	Title
CS	246	0.5	Object-Oriented Software Devel

Notes: Choose TUT section for Related 1.

Class	Comp Sec	Camp Loc	Assoc. Class	Rel 1	Rel 2	Enrl Cap	Enrl Tot	Wait Cap	Wait Tot	Time Days/Date	Bldg Room	Instructor
6044	LEC 001	UW U	1		201	96	96	0	0	02:30-03:50TTh	MC 2035	Naeem,Nomair Ahmed
6403	LEC 002	UW U	2		201	96	95	0	0	08:30-09:50TTh	MC 2034	Zhang,Cheng
6451	LEC 003	UW U	3		201	104	106	0	0	01:00-02:20TTh	MC 2017	Naeem,Nomair Ahmed
6579	LEC 004	UW U	4		201	96	96	0	0	11:30-12:50TTh	MC 1056	Zhang,Cheng
7762	LEC 005	UW U	5		201	105	105	0	0	10:00-11:20TTh	RCH 110	Naeem,Nomair Ahmed
6081	TUT 101	UW U	99	99		72	56	0	0	08:30-09:20F	DWE 3516	
6082	TUT 102	UW U	99	99		65	65	0	0	09:30-10:20F	RCH 204	
6404	TUT 103	UW U	99	99		84	86	0	0	10:30-11:20F	RCH 309	
6452	TUT 104	UW U	99	99		72	73	0	0	11:30-12:20F	RCH 207	
6453	TUT 105	UW U	99	99		72	72	0	0	11:30-12:20F	DWE 3516	
6577	TUT 106	UW U	99	99		72	73	0	0	12:30-01:20F	RCH 207	
6598	TUT 107	UW U	99	99		72	73	0	0	02:30-03:20F	DWE 3516	
6116	TST 201	UW U	99	99		497	498	0	0	04:30-06:20Th 10/29-10/29		

Contact Information

Role	Name	**Location**	**Email**	Office Hours
Instructor	Cheng Zhang	DC 3334C	cheng.zhang@uwaterloo.ca	M 130-230pm W 2-3pm
Instructor	Nomair Naeem	DC 3111	nanaeem@uwaterloo.ca	M 12-130pm Monitor Google Calendar
ISA	Sean Harrap	MC 4065	cs246@uwaterloo.ca	M,T,W,TH 11-12pm
ISA	Yang Tian Zi	MC 4065	ytzi@uwaterloo.ca	M,T,W,TH, 3-4pm
ISA	Richard Wallace	—	rbwallace@uwaterloo.ca	—
IA	Adam Domurad	MC 4065	adomurad@uwaterloo.ca	T,TH, 11:30-1pm
ISC	Gang Lu	MC 4008	glu@uwaterloo.ca	—

You can also contact the course staff via email for a scheduled appointment.

Website: http://www.student.cs.uwaterloo.ca/~cs246

Newsgroup

CS 246 will be using Piazza.
We encourage you to use the newsgroup. It is a handy and timely way for students to learn from each other and the course staff. Essential announcements will also be posted here.

Make sure you read the rules for using Piazza as posted here:
`https://piazza.com/class/ie1ho5ifovl7ma?cid=8`

TextBook

There is no required textbook for this course. Course notes, available under the Resources section of the course website, are an excellent reference. However, if you want to purchase a reference textbook, we recommend *Savitch, Walter. Absolute C++, 5th edition, Addison Wesley.*.

Course Topics

- The shell (4 hours)
  - file system, pattern matching, quoting, shell/system commands, file permission, redirection, shell programming.
- C++ (16 hours)
  - declarations, expressions, control structures, structured programming, preprocessor, 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., git).

- Software engineering (8 hours)
  - development process, design (UML), language selection, patterns, testing

Student Assessment

Assignments	Number	Tentative Due Date	Weight	Total
	0	Due Date: Sept, 23	0	
	1	Due Date 1: Sept, 28	5	
		Due Date 2: Oct, 2		
	2	Due Date 1: Oct, 7	5	
		Due Date 2: Oct, 14		
	3	Due Date 1: Oct, 26	9	
		Due Date 2: Nov, 4		
	4	Due Date 1: Nov, 11	9	
		Due Date 2: Nov, 18		
	5	Due Date 1: Nov, 25	12	
		Due Date 2: Dec, 4		
				40
Tests	Midterm	Oct 29, 2015	20	20
	Final	Set by Registrar's Office	40	40

Important Notes:

1. You cannot get credit for assignments 1-5 until you have achieved 100% on assignment 0.
2. There is NO way to get EXTRA credit in this course.
3. The above mentioned due dates for assignments are subject to change

Passing the course

A passing mark in the weighted test portion of the final grade must be achieved to pass the course. The final grade is calculated using the following formula:

```
if ( testing_average < 50% ) then
    final_grade = MIN( weighted_testing_average , normally_calc_grades )
else
    final_grade = normally_calc_grades
```

In other words, good assignment marks alone cannot get you a passing mark in this course.

Assignments

Assignments will be distributed through the course's `git` repository. To gain access to the repository, follow the directions given in assignment 0 which can be found on Piazza. Future assignments will NOT be posted on the course website and/or Piazza.

All assignments must be done individually, unless explicitly designated as a group assignment.

Students repeating the course must choose a different project from the one they chose when they previously took the course.

Note: The final assignment is a project to be done in pairs. All members of a group receive the same grade (no exceptions). Only one member of a group submits the assignment. The instructors/staff do not arbitrate group disputes; group members must handle any and all problems. A group assignment may be done individually, but it must be understood that the amount of work is significantly greater and no extra marks are given for this additional work. Part of the evaluation of your project will consist of a live demo in front of a TA. This is mandatory.

Late Assignments

- Assignments are due on the due-date at the time specified on the assignment (usually 16:55).
- Late assignments are not accepted.
- An assignment not handed in receives a mark of 0, unless there is a documented reason.
- If a documented reason is available, talk to the instructor within one week of the assignment due date to see if an accommodation can be made.

Assignment Submissions

Assignments must be submitted using the Marmoset Submission and Testing Server
(URL: https://marmoset.student.cs.uwaterloo.ca/)

Marmoset downtime

If Marmoset fails to accept submissions for more than two of the six hours immediately prior to the deadline, or is down at the deadline, a 12-hour extension will be granted. For an extension to be granted, Marmoset must fail to accept submissions; failure or delay in displaying results is not grounds for extension. It is bad practice, and risky, to rely on Marmoset as your primary means of testing. The failure must be due to a problem with Marmoset or a widespread network failure; your home connection is your own responsibility.

Assignment Marking

- An assignment is marked on the following criteria: correctness, documentation, testing, style and efficiency.
- Note that correctly-working code is only one component to an assignment mark.
- Documentation is very important.
- You should perform adequate testing on your code before submitting the final version. If your makefile does not work, or code does not compile, the correctness mark is zero.
- Use only relative paths in includes, not absolute (unless specified otherwise in assignment).
- All assignments must compile and run on one of the linux.student.cs computers.

Assignment Returning and Remarking Policy

- Assignment grades are available through the MarkUs System
- To request a remark, send an email to the ISA clearly stating the questions you want remarked and any supporting evidence for your case.

- Remark requests will not be accepted if more than two weeks have passed since the assignment was handed back.

Note: The entire assignment is examined when remarking; therefore, the assignment grade could receive a lower mark than the current mark.

Tutorials

Multiple tutorial sections are offered per week; each covering the same content. As long as there is adequate space, you may attend any one.

Tutorials offered in certain weeks might be considered mandatory and include testable material. Prior notice will be given in class or through Piazza.

Tutorial material is available through the git repository

Exams

Both the midterm test and the final exam are closed book.

A missed test/exam receives a mark of 0, unless there is a valid documented reason. If a documented reason is provided for missing the midterm, its weight is applied to the final exam. If a documented reason is provided for missing the final exam, a grade of INC MIGHT be given. For a missed final exam, upon the receipt of a valid doctor's note, the student's term work is evaluated to determine if a grade of INC (incomplete) is suitable. If an INC is granted, the student's grade will be calculated using the weightings of the course components from the term the student was registered, the student's term marks from the registered term, and the mark from the final exam when it is written. The final exam should be written with the scheduled exam in the following semester. A copy of the documented reason must be given to the Instructional Support Coordinator for the course.

Unclaimed midterms are retained for one month after term grades become official in Quest. After that time, they are destroyed in compliance with UW's confidential shredding procedures.

Academic Offenses

Students are expected to know what constitutes academic integrity, to avoid committing academic offenses, and to take responsibility for their actions. Students who are unsure whether an action constitutes an offense, or who need help in learning how to avoid offenses (e.g., plagiarism, cheating) or about rules for group work/collaboration should seek guidance from the course professor, tutor, TA, IA, academic advisor, or the Undergraduate Associate Dean. For information on categories of offenses and types of penalties, students should refer to Policy #71, Student Academic Discipline, http://www.adm.uwaterloo.ca/infosec/Policies/policy71.htm. Students who believe that they have been wrongfully or unjustly penalized have the right to grieve; refer to Policy #70, Student Grievance, http://www.adm.uwaterloo.ca/infosec/Policies/policy70.htm.

All assignments in CS246 are to be done individually. You are welcome to discuss general ideas regarding assignments with other students in the class, but no code-level sharing is permitted. You may not look at someone else's code, nor share your code with someone else, either in person or via electronic communication (e.g., instant messaging, course newsgroup). When code is shared, **both** parties are considered to have committed an offence. Marmoset tokens cannot be shared; it is an offence to "borrow" someone else's Marmoset account for the purpose of using extra release tokens for testing, or for any other purpose. If you have taken this course before we strongly recommend that you do each assignment from scratch.

Although each assignment is worth only about 5-9% of your final grade, the penalty for an offence under Policy 71 is a grade of 0 on the assignment and an *additional* 5% deduction from your course grade.

Note: The final assignment (a5) is a project to be done in pairs. Standard plagiarism rules apply. All code submitted must be the work of students in the group (with the exception of any code provided by the course staff). Students repeating the course must choose a different project from what they did last time they took the course. If you are caught cheating in assignment 5, the penalty is the same; 0 on the assignment and an additional 5% deduction from your course grade.

Note for students with disabilities: The Office for Persons with Disabilities (OPD), located in Needles Hall 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.