University of Saskatchewan
Department of Computer Science
176 Thorvaldson Building
110 Science Place, Saskatoon, SK, S7N 5C9, Canada
Telephine: (306) 966-4886, Facimile: (306) 966-4884

Course Syllabus
CMPT 117: Computing II

Catalogue Description

Continues the development of programming skills started in CMPT 116, with an emphasis on object-oriented programming. Data structures for the storage and efficient retrieval of information will be studied and analyzed, in particular stacks, queues, linked lists and simple binary trees. Examples and exercises will be drawn from engineering applications and numerical methods.

Prerequisite(s): CMPT 116; or (CMPT 113 with grade at least 75%).

Course Website: All course relevant information (announcements, course materials, assignments, exam schedules, etc.) will be on the Moodle website: http://moodle.cs.usask.ca. Each student is responsible for checking this website regularly. It is also your responsibility to check your PAWs email account regularly.

Class & Instructor Information

This course is taught in parallel with two sections of CMPT 115 with common lecture material, assignments, laboratories, and exams. Please attend the section you are registered in, as classroom space is limited and registered students have priority seating.

Section 02

Instructor: *Craig Thompson*

Email: craig.thompson@usask.ca

Time: MWF: 11:30 am - 12:20 pm

Location: GEOL 261

Office Hours: Wednesday 2:30pm - 4:30pm

Course Overview

This course introduces the basic concepts of computer science and object oriented software development. You will learn about fundamental *data structures* for organizing data, including lists, stacks, queues, trees, and hash tables, and associated algorithms, as well as their *time and space efficiency*. The course will emphasize *abstract data types* for the design of data storage mechanisms that can be reused and revised. You will learn the basics of object oriented programming, as a natural technological extension of abstract data types. As the practical part of the course, you will develop a familiarity with memory management, including static and dynamic memory allocation, and pointers, through hands-on implementations.

An underlying theme of the course is for students to gain programming and debugging skills. The conceptual material covered in the course is actually fairly straightforward, and can be mastered with a moderate amount of study. The real challenge in this course is to develop the programming skills needed to complete the homework. Students should practice time management, problem solving strategies, and critical, analytical, and scientific thinking.

Learning Outcomes

By the end of this course, your are expected to be able to:

- be proficient in fundamentals of procedural programming, specifically programming in the procedural subset of C++
- design algorithms using pseudocode, and analyze algorithms written in pseudocode
- analyze time and space complexity of algorithms, and to compare and evaluate algorithms and data structures
- understand and use dynamic memory and static memory in procedural programming
- describe and apply the techniques associated with references, pointers and addresses
- explain the concepts behind the use of data structures, and determine the appropriateness of different data structures for various purposes
- design, implement, and apply specific data types: linked lists, arrays, trees, binary search trees and hash table data structures
- explain the concept of abstract data types in terms of interface and encapsulation
- design, implement, and apply abstract data types for: linked lists, arrays, trees, binary search trees and hash table data structures
- apply recursion to computational tasks involving data structures
- describe and apply the fundamentals of object-oriented programming in C++, specifically as an extension of the ADT concept

Resources

Textbook Information

There are no required textbooks. However, we provide the following recommended references:

Richard F. Gilberg and Behrouz A. Forouzan *Data Structures: a Pseudocode Approach in C*, 2$^{ed.}$, Course Technology (Thompson), 2005.

K. N. King *C Programming: a Modern Approach*, 2$^{ed.}$, W. W. Norton & Company, 2008.

Substantial lecture notes will be provided on the course webpage. However, students should not rely solely on the lecture notes for the course.

Grading Scheme

The grading scheme for this course appears in the following table:

Assignments (ten @ 3%)	30%
Tutorial Exercises (ten @ 1%)	10%
Midterm Exam (February 25, evening)	25%
Final Exam (April 11-30)	35%
Total	100%

- Assignments (see the schedule on page 5 for the due date of each assignment)

  There will be ten assignments in this course, one approximately every week. Assignments will consist of two portions:

  1. written questions that require written answers;
  2. programming questions which require you to write computer programs.

  Submission instructions will be included with each assignment description. Generally, you will upload your solutions as files to Moodle, unless you are instructed otherwise. Generally, text files are preferred to documents that include formatting (e.g., MSWord documents). A document that cannot be opened will receive a grade of zero; do not assume the markers will take the time to open your file if it is in a file format that is not standard.

  Note: All computer programs must be written in C++ and must compile using the GNU C++ compiler (`g++`) under Linux: the standard will be the `tuxworld.usask.ca` cluster of machines (which is the same version of GNU C++ as found on the Linux desktops in the lab).

- Tutorials

  Tutorials have associated exercises, which we expect you to do for the skills you learn by doing them. You are expected to submit your tutorial exercises weekly, along with your assignment solutions. These exercises will be graded, and will make up 10% of your total course grade. You should expect to complete these exercises in the time allotted for tutorials, and should not require any time outside the tutorials.

- Mid-term Examination

  The midterm exam is scheduled for the evening of February 25 (location: TBA). It will be held in common with CMPT 117 and both lecture sections of CMPT 115. There are two seatings that students can choose from to meet their schedule:

Time	Note
4:30PM–6:00PM	students cannot leave early (before 5:30PM)
5:30PM–7:00PM	students will not be allowed to enter late (after 6PM)

  We'll use Moodle to sign up for the midterms starting about 2 weeks before. The rules about leaving early and arriving late are to ensure fairness in the examination.

  The mid-term examination is written, closed-book; only bring water, your student card, and writing instruments. The mid-term examination is intended to provide practice for the final exam, and to provide feedback to students regarding their current performance.

  Please see the section on Policies for important information concerning missed midterms and final exams. In the case of a missed midterm, the instructor, in consultation with the student, will determine how the missed work will be compensated for; one potential alternative is transferring the weight of the midterm onto the final examination.

- Final Examination

  The final examination, common to CMPT 117 and both lecture sections of CMPT 115, will be scheduled by central timetabling to occur during the usual final examination interval. It will be three hours long, written, closed-book; bring only water, your student card, and writing instruments.

  Please see the section on Policies for important information concerning missed midterms and final exams.

Attendance expectations

We expect 100% attendance in lecture and in tutorial, with reasonable allowances for illness and unforeseen life-events. In other words, treat CMPT 117 as if you were an intern at a real company; your "bosses" (the instructors and teaching assistants) expect you to show up on time to all lectures and tutorials, master your skills, do your share of the work, and behave professionally. There are almost no consequences for missing class or tutorial, apart from the opportunity cost of paying tuition and not being present to discuss course material with instructors.

Tutorials and Help Sessions

Tutorials

Tutorials in a laboratory setting are mandatory and include new material not presented in class. Lectures emphasize the data organization concepts using pseudocode; tutorials focus on how to implement, in C++, the concepts studied in lecture. Material presented in tutorial is examinable. If you miss a tutorial section, you may try to attend another section during that week; but there is considerable risk you will not be able to find a seat.

Section	Day	Time	Location	Leader
T02	Monday	2:30PM – 3:50PM	SPINKS 311	Matthew Miller
T04	Tuesday	2:30PM – 3:50PM	SPINKS 311	Matthew Miller

Help Sessions

Obstacles to progress and completion of assignments can sometimes be part of the homework (i.e., something we want you to think about carefully), and sometimes beyond the scope of the course (i.e., a problem that you can't really be expected to manage in first year), and it can be nearly impossible to tell the difference without some advice from a TA or instructor.

There are several help sessions in the Spinks Computer Lab, that are specifically for CMPT 115/117 students. The TAs are all prepared for the assignments and lab questions. We highly recommend you to work in the computer lab, because it is very helpful if you can get help when you have difficulties. The schedule will be announced in the first two weeks of the term on Moodle.

UNIVERSITY OF SASKATCHEWAN
Department of Computer Science
176 Thorvaldson Building
110 Science Place, Saskatoon, SK, S7N 5C9, Canada
Telephine: (306) 966-4886, Facimile: (306) 966-4884

Lecture & Tutorial Schedule and Topics

The following schedule is subject to minor adjustments. The topics may shift a bit, but due dates will not.

	Date	Lecture Topic	Tutorial Topic
Week 1	Jan. 5	*First day of classes*	
	5-9	**Topic 0: Introduction (1hr)** **Topic 1: Software Design (1hr)** **Topic 2.1: Algorithm Analysis (1hr)**	(no tutorials this week)
Week 2	12	*CMPT 117 tutorials start this week*	
	12-16	**Topic 2.2: Algorithm Analysis (1.5hr)** **Topic 3.1: Pointers and References (1.5hr)**	Tut. 1: The UNIX command line, and the compiler
		Assignment 1 due on Jan. 16 at 6pm (pseudocode, software design, algorithm complexity)	
	16	*Last day for changing registration*	
Week 3	19-23	**Topic 3.2: Pointers and References (1.5hr)** **Topic 4: Memory (1.5hr)**	Tut. 2: Dynamic memory (pointers & refs), arrays and records
		Assignment 2 due on Jan. 23 at 6pm (pointers & references, C strings & arrays)	
Week 4	26-30	**Topic 5: Abstract Data Types (1hr)** **Topic 6.1: Lists (2hr)**	Tut. 3: Multiple file compilation
		Assignment 3 due on Jan. 30 at 6pm (ADTs, memory, references)	
Week 5	Feb. 2-6	**Topic 6.2: Lists (2hr)** **Topic 6.3: List Traversal (1hr)**	Tut. 4: C Strings and Arrays (as pointers)
		Assignment 4 due on Feb. 6 at 6pm (Lists, dynamic memory)	
Week 6	9-13	**Topic 7: Stacks (1hr)** **Topic 8: Recursion (1.5hr)** **Topic 9.2: Queues (0.5hr)**	Tut. 5: Singly Linked List Implementation
		Assignment 5 due on Feb. 13 at 6pm (stacks, recursion)	
	16-26	Mid-term break	
Week 7	Feb. 23-27	**Topic 9.2: Queues (0.5hr)** **Topic 10.1: Trees (2.5hr)**	(no tutorial work this week – tutorials open for consulting for midterm)
	Feb. 25	*Mid-term Exam (4:30pm-7pm – see section on Midterm examination)*	
		No assignment due this week	
Week 8	Mar. 2-5	**Topic 10.2: Trees (0.5hr)** **Topic 11.1: Binary Search Trees (2hr)** **Topic 12.1: Objects (0.5hr)**	Tut. 6: Array-Based Stack ADT
		Assignment 6 due on Mar. 7 at 6pm (recursion, trees)	
Week 9	9-13	**Topic 12.2: Objects (2.5hr)** **Topic 12.3: Recap: Lists,stacks, queues as Objects (0.5hr)**	Tut. 7: Binary Tree ADT Implementation
		Assignment 7 due on Mar. 14 at 6pm (BSTs, queues)	
	15	*Last day for withdrawing without penalty*	
Week 10	16-20	**Topic 12.4: Recap: Lists,stacks, queues as Objects (2.5hr)** **Topic 13.1: Hashing (0.5hr)**	Tut. 8: Object Oriented Programming
		Assignment 8 due on Mar. 21 at 6pm (Object oriented programming)	
Week 11	23-27	**Topic 13.2: Hashing (1hr)** **Topic 14.1: Topics (2hr)**	Tut. 9: TBA
		Assignment 9 due on Mar. 28 at 6pm (Object oriented programming)	
Week 12	Mar. 30-Apr. 3	**Topic 14.2: Topics (2hr)**	Tut. 10: TBA
		Assignment 10 due on Apr. 8 at 6pm (TBA)	
	Apr. 3	*Good Friday: University closed.*	
Week 13	Apr. 6-8	**Topic : Course Review (1hr)**	(no tutorials this week)
	8	*Last day of classes*	
Final exam	11-30	*(centrally scheduled)*	

Policies

Late Assignment Policy

Due to the aggressive assignment schedule, late submissions cannot be accepted. Be sure to start your assignments early, and hand in partial solutions for partial credit. We understand that legitimate, exceptional circumstances sometimes prevent a deadline from being met. If you feel you cannot submit an assignment on time, please talk to the course instructor at least a day before the assignment is due. Extensions on assignments will be granted only by the course instructor. All extension requests will require suitable documentation.

Missed Examinations

1. "Students who have missed an exam or assignment must contact their instructor as soon as possible. Arrangements to make up the exam may be arranged with the instructor. Missed exams throughout the year are left up to the discretion of the instructor if a student may make up the exam or write at a different time. If a student knows prior to the exam that she/he will not be able to attend, they should let the instructor know before the exam."

2. "Final exams - a student who is absent from a final examination through no fault of his or her own, for medical or other valid reasons, may apply to the College of Arts and Science Dean's office. The application must be made within three days of the missed examination along with supporting documentary evidence. Deferred exams are written during the February mid-term break for Term 1 courses and in early June for Term 2 and full year courses."

   (2007/08. http://www.arts.usask.ca/students/transition/tips.php)

Incomplete Course Work and Final Grades

When a student has not completed the required course work, which includes any assignment or examination including the final examination, by the time of submission of the final grades, they may be granted an extension to permit completion of an assignment, or granted a deferred examination in the case of absence from a final examination. Extensions for the completion of assignments must be approved by the Department Head, or Dean in non-departmentalized Colleges, and may exceed thirty days only in unusual circumstances. The student must apply to the instructor for such an extension and furnish satisfactory reasons for the deficiency. Deferred final examinations are granted as per College policy.

In the interim, the instructor will submit a computed percentile grade for the course which factors in the incomplete course work as a zero, along with a grade comment of INF (Incomplete Failure) if a failing grade. In the case where the instructor has indicated in the course outline that failure to complete the required course work will result in failure in the course, and the student has a computed passing percentile grade, a final grade of 49% will be submitted along with a grade comment of INF (Incomplete Failure).

If an extension is granted and the required assignment is submitted within the allotted time, or if a deferred examination is granted and written in the case of absence from the final examination, the instructor will submit a revised computed final percentage grade. The grade change will replace the previous grade and any grade comment of INF (Incomplete Failure) will be removed. For provisions governing examinations and grading, students are referred to the University Council Regulations on Examinations section of the Calendar.

(2011 University of Saskatchewan Calendar/Academic Courses Policy)

Academic Honesty

The University of Saskatchewan is committed to the highest standards of academic integrity and honesty. Students are expected to be familiar with these standards regarding academic honesty and to uphold the policies of the University in this respect. Students are particularly urged to familiarize themselves with the provisions of the Student Conduct & Appeals section of the University Secretary Website and avoid any behaviour that could potentially result in suspicions of cheating, plagiarism, misrepresentation of facts and/or participation in an offence. Academic dishonesty is a serious offence and can result in suspension or expulsion from the University.

All students should read and be familiar with the Regulations on Academic Student Misconduct,

http://www.usask.ca/secretariat/student-conduct-appeals/StudentAcademicMisconduct.pdf,
as well as the Standard of Student Conduct in Non-Academic Matters and Procedures for Resolution of Complaints and Appeals,
http://www.usask.ca/secretariat/student-conduct-appeals/StudentNon-AcademicMisconduct.pdf.

Academic honesty is also defined and described in the Department of Computer Science Statement on Academic Honesty:
http://www.cs.usask.ca/undergrad/honesty.php.

For more information on what academic integrity means for students see the Student Conduct & Appeals section of the University Secretary Website at:
http://www.usask.ca/university_secretary/pdf/dishonesty_info_sheet.pdf

Examinations with Disability Services for Students (DSS)

Students who have disabilities (learning, medical, physical, or mental health) are strongly encouraged to register with Disability Services for Students (DSS) if they have not already done so. Students who suspect they may have disabilities should contact DSS for advice and referrals. In order to access DSS programs and supports, students must follow DSS policy and procedures. For more information, check http://www.students.usask.ca/disability/, or contact DSS at 966-7273 or `dss@usask.ca`.

Students registered with DSS may request alternative arrangements for mid-term and final examinations. Students must arrange such accommodations through DSS by the stated deadlines. Instructors shall provide the examinations for students who are being accommodated by the deadlines established by DSS.