

University of Calgary
**CPSC 233: Introduction to Computer
Science for Computer Science Majors II**
Winter 2013

Course Information

Course:	CPSC 233, Introduction to Computer Science for Computer Science Majors II
Lecture/Time:	L02, MWF 10:00A.M. — 10:50A.M.
Location:	ST 147
Prerequisites:	Computer Science 231
Antirequisites:	Credit for both Computer Science 233 and any of 219, 235, Electrical Engineering 497 or Computer Engineering 493 will not be allowed.

Instructor Information

Instructor:	James Tam
Office:	ICT 707
Email:	tamj@cpsc.ucalgary.ca
Phone:	(403) 210-9455
Office Hours:	1:00P.M. - 1:50P.M., 3:00P.M. - 3:50P.M. Mondays 11:00A.M. - 11:50A.M. Wednesdays

Textbook & References

The recommended textbook for this course is **Absolute Java**, 5th edition, by Walter Savitch (Pearson Education, 2013). This text should be available in the University of Calgary Bookstore. This course uses the recommended textbook as the primary source, but also covers materials from other sources. Several reference books are recommended by the instructor:

- *Big Java*, 3rd Edition, Cay S. Horstmann, Wiley, 2007.
- *Introduction to Programming in Java*, 1st Edition, Robert Sedgewick, Kevin Wayne, Addison-Wesley, 2007.
- *Thinking in Java*, 3rd Edition, Bruce Eckel, Prentice Hall PTR, 2002.
- Other sources of help includes the (free) online library website: <https://login.ezproxy.lib.ucalgary.ca/login?url=http://proquest.safaribooksonline.com/>

Course Description

“Continuation of Introduction to Computer Science for Computer Science Majors I. Emphasis on object-oriented analysis and design of small-scale computational systems and implementation using an object oriented language. Issues of design, modularization, and programming style will be emphasized.” (calendar, p. 337)

This course is designed to give you an introduction to computer programming and problem solving using Java. The course consists primarily of lectures and assignments. Major topics to be covered in the lectures include: object-oriented programming (encapsulation, inheritance, polymorphism), arrays, linked list, recursion, exception handling, I/O, design patterns, and GUIs. While the lectures will emphasize on the OO and other important concepts, as well as the analysis of the solutions, the assignments will provide you opportunities for hands-on experience and for better grasp of the programming skill. By the end of this course, you should be able to conduct object-oriented analysis and designs and to implement the designs in Java.

Course Policies (supplemental to the U of C calendar)

Email Policy

The best way to reach your instructor is during office hours or by appointments. When contacting the instructor via email, please use “**CPSC233**” as the prefix in the subject line. Otherwise, your emails might not be addressed.

Missed Quizzes, Exams, or Assignments (calendar, p. 42)

Make-up examinations and grade adjustment will not be given except in cases of extreme personal emergencies. For further information see the University Calendar.

Late Assignments

Late assignments will not be accepted except in cases of extreme personal emergencies. Penalty may be applied to late submissions.

Reappraisals and Appeal (calendar, p. 45)

Students seeking reappraisal of a piece of graded term work (including the midterm exam) should discuss their work with the instructor within *15 calendar days* of it being returned to the class. Regraded work stands an equal chance of having a higher or lower score assigned. Any term work that is not returned during regular lecture or tutorial time may be collected from the instructor during office hours.

Academic Dishonesty (calendar, p. 48)

“A single offence of cheating, plagiarism, or other academic misconduct, on term work, tests, or final examinations, etc., may lead to disciplinary probation or a student’s suspension or expulsion from the faculty by the dean, if it is determined that the offence warrants such action.” **More information about academic misconduct can be found in the appendix of this handout.**

Grading

The grading in this course will be based on five *individual* assignments, four quizzes, one midterm exam, and one final exam. The midterm exam will cover all materials up to a point that will be specified in class. The final exam will cover all materials in the course. The composition of the final grade is as follows:

Assignments	32%
Quizzes	8%
Midterm Exam	25%
Final Exam	35%

The 75-minute midterm exam will be held in ST 148 from 7:00P.M. to 8:15P.M. on *March 1, 2013*. All exams are closed book. No aids are allowed for both the midterm exam and the final exam.

Each of the above components will be given a letter grade using the official University grading system. The final grade will be calculated using the grade point equivalents weighted by the percentage given above and then reconverted to a final letter grade using the official University grade point equivalents. In order to obtain a final grade of C- or better, a student must achieve a weighted average of D or better on the midterm and final exams.

Course Web Site

Course information is accessible on the course Web site:

<http://pages.cpsc.ucalgary.ca/~tamj/233/index.html>

Information will be updated on a regular basis, so check the Web site regularly.

Lectures

Lectures are in ST 147 from 10:00A.M. to 10:50A.M. on Mondays, Wednesdays and Fridays. We will cover most materials from, but not limited to, Chapters 1 – 13, 15, 17 – 18 in the textbook. Certain topics will be discussed in more depth and will be more extensive than the textbook. You are responsible for all materials covered in class and for knowing any in-class or online announcements that were made during the term. Since the exams will be based on the materials covered in class, it is strongly recommended that you attend each lecture. Below is a list of topics to be covered in this course.

	Topics
1	Introduction to Java (Chapters 1 and 3)
2	Basic I/O (Chapter 2)
3	OOP-Encapsulation (Chapters 4 and 5)
4	Arrays (Chapter 6)
5	Linked list (Chapter 15)
6	Advanced Java (Chapter 5)
7	Recursion (Chapter 11)
8	OOP-Inheritance (Chapter 7)
9	OOP-Polymorphism and abstract classes (Chapter 8)
10	Interfaces and inner classes (Chapter 13)
11	Exception handling (Chapter 9)
12	File I/O (Chapter 10)
13	Design patterns (Chapter 12)
14	GUIs (Chapters 17 and 18)

Tutorials

The tutorial sessions will provide helps on the assignments and extra programming examples. Your TA is a good resource for getting help in this course. Please attend your registered tutorial section and get to know your TA.

Section	Time	Location	TAs
T05	11:00A.M. - 11:50A.M. TR	MS 176	Md. Moniruzzaman
T06	9:00A.M. - 9:50A.M. TR	MS 176	Md. Moniruzzaman
T07	1:00P.M. - 1:50P.M. TR	MS 176	Dina Said
T08	3:00P.M. - 3:50P.M. MW	MS 176	Hoi Le

There will also be four 15-minute quizzes, each worth 2%, taking place at the end of the tutorial sessions (see the schedule below). As each quiz will be conducted in different tutorial sections, do not exchange with other students knowledge of the questions on the quizzes. These questions should be treated as exam questions. Failure to observe this policy will be treated as cheating.

Below is the tentative schedule for the tutorials and the quizzes, which may be altered slightly according to our actual progress. Any revisions will be announced in class and posted on the course Web site.

Week	Date	Topics
1	Jan 7 - 11	<i>No tutorials</i>
2	Jan 14 - 17	Tutorials
3	Jan 21 - 22 Jan 23 - 24	Tutorials Review and Quiz 1
4	Jan 28 - 29 Jan 30 - 31	Tutorials Help on Assignment 1
5	Feb 4 - 5 Feb 6 - 7	Tutorials Review and Quiz 2
6	Feb 11 - 14	Help on Assignment 2
7	Feb 18 - 22	<i>No tutorials (Reading Week)</i>
8	Feb 25 - 26 Feb 27 - 28	Tutorials <i>Midterm Review</i>
9	Mar 4 - 7	Help on Assignment 3
10	Mar 11 - 12 Mar 13 - 14	Tutorials Review and Quiz 3
11	Mar 18 - 21	Help on Assignment 4
12	Mar 25 - 26 Mar 27 - 28	Tutorials Review and Quiz 4
13	Apr 1 - 4	Tutorials
14	Apr 8 - 11	Help on Assignment 5
15	Apr 15 - 16	<i>Final Review</i>

Assignments

There will be five individual assignments that count 32% toward your final grade. *No late submission is accepted (Please refer to the policy on Late Assignments)*. Assignments will consist of programming questions and will be graded on a percentage scale. Each assignment is due at 4:00P.M. on the due date via the submit program. The table below provides the weight of each assignment and the tentative schedule. Please refer to the course Web site for the latest update.

Assignments	Weight	Due
1	6%	Feb 1
2	6%	Feb 15
3	6%	Mar 8
4	7%	Mar 22
5	7%	Apr 12

The assignments will definitely be time consuming; it is important to start as early as possible and to avoid procrastination. The experience gained in doing the assignments is

one important factor in determining your success in this course. The programming questions require knowledge in UNIX/Linux operating systems. A handout describing each assignment will be available on the course Web site. There will be no hard copies handed out in class, so make sure to check the course Web site for assignment handouts.

For fairness and consistency, all programming questions will be marked on the computers in the computer lab (1st floor MS). You may work on the programming questions on your own computer, but please make sure to test your program under your CPSC account before submitting the assignments. If a program does not work on the lab computers, it is considered incorrect and you will not get full marks. It is important that you do not customize your CPSC account since the TAs will test your program with the default account configuration.

Tips for Success in CPSC 231

- Know the CPSC computer lab on the 1st floor in MS. This will be your primary working environment for this course. Helps on setting up your own computer for this course are available at best effort only.
- Do coursework every day. Practice programming is the key to understand the concepts and retain the skills. Last-minute cramming definitely will not work in this course.
- Plan your time around the assignment deadlines, quizzes, and exams.
- Do not miss lectures and tutorials. Someone else's notes will not be as good as your own notes. Participating in-class discussion will enhance your learning experience and reinforce the memorization of the course materials.
- Read the textbook, study the lecture notes, and do the assignments.

Sources of Help

- For most any questions related to the course (e.g., problems with assignments, questions about lecture material etc.), please feel free to ask your course instructor.
- For questions about the tutorial material or the marking of assignments, please see your TAs.
- There are tutors that work in the Continuous Tutorial (CT for short) that you can go to for more help. Use the entrance to the computer lab on the first floor of MS building that is near the triple elevators (towards ES building rather than the entrance closer to the ST building) and you should see several signs just inside the entrance for each of the Continuous Tutorials. The CT hours will be available on the course Web site soon.

- Please visit the “help desk” located in MS 151 for system problems in UNIX. For example, this is the place to go, if your computer is “locked-up” and it no longer accepts any input (nothing appears to happen when you use the mouse or keyboard) or if you forgot your password and need to be issued a new one.
- If you have questions about the administrative aspects of Computer Science such as what courses that you should take for different Computer Science concentrations or if want to know if courses from another post-secondary institution will transfer over to the U of C, then the main Computer Science office, located in ICT 602 is probably where you should begin your inquiries.
- Please also check out the useful links on the course website for supplementary materials, such as connecting to the CPSC servers from home and *vim* or *emacs* editors.

Appendix: Academic Misconduct (Cheating v.s. Collaboration)

Assignments must reflect individual work. Each student must demonstrate that he or she can complete the assignment on their own so you cannot copy the work of other students nor can students work in groups. Any suspected cases of cheating must be forwarded by me onto the Department Head, which may be passed on further to the Office of the Dean of Science and Student Services resulting in penalties such as failing the course or even expulsion from the university.

A few questions and answers to help clarify things:

- **Q:** What constitutes cheating in this course?

A: It is probably similar to what you have seen in other courses. Cheating has occurred if you hand in someone else's work as if it were your own (without crediting the other person). **Furthermore if a student knowingly provides his or her graded work to another student** then both students are guilty of academic misconduct (the first student helped the second student to cheat).

- **Q:** What happens if you include someone else's code and you do credit the other person properly e.g., You use the code from a text book and you include the following citation: *The function (listed below) for opening and reading from a file was taken in it's entirety from the book "The Tao of file systems" by James Tam.*

A: This will not constitute cheating because you clearly indicated that the work was not your own but since someone else did the work for that section of your assignment you won't get credit for that part of the assignment. You could get marks for the other parts of the assignment that you wrote yourself. The crediting of other's people work must be very specific and clear because your marker needs to be able to unambiguously determine which parts of your program did you complete and which parts came from an outside source. It's probably best to directly document each block of external program code that you've used.

- **Q:** The code given in lectures would be really handy for our assignments, are we allowed to use it and get credit for the work?

A: Yes, unless you are told otherwise you can make use of the sample code and get credit. Just make sure that you indicate where you got it from in your program documentation. Don't just include it in your code without citing the source (in this case it's the instructors) because you will be claiming that this work is yours when it isn't so you will be guilty of academic misconduct.

- **Q:** What is the difference between getting help from someone vs. cheating?

A: If you describe the process to someone using plain English then you should be okay because then the person still must figure out how to implement your generic ideas in

the appropriate programming language used for the course (both of you are handing in separate submissions). If you simply give (or communicate in some form) your code to a friend then this is NOT okay, even if your friend says that he or she will only use your solution only as a *guide* to figure out the answer and *promises* that he or she won't just copy it into their own program.

OK	Not OK
<p>To display the contents of a two-dimensional array onscreen you will need two nested loops and two loop controls: one to keep track of the row that is currently being displayed and one to track the current column value. Start by initializing the outer loop to the value of the lowest row value and the inner loop to the value of the lowest column value and display that array element. The inner loop will travel along the row from the lowest column value to the highest column value. After the last column has been reached, the outer loop value increases by one which allows the inner loop to traverse the second row. The outer loop will repeat until all the rows have been traversed.</p>	<pre data-bbox="808 598 1477 892"> for (int r = 0; r <= 4; r++) { for (int c = 0; c <= 4; c++) { System.out.print(my_array[r][c]); } System.out.println(''); } </pre>

This list of questions only includes things that has arisen before, if you ever unsure if a particular situation constitutes cheating or not then it is up to you to consult with your instructor.