

UNIVERSITY OF WATERLOO  
**DEPARTMENT OF MANAGEMENT SCIENCES**  
FACULTY OF ENGINEERING  
COURSE INFORMATION SHEET  
September 7, 2016

## Course Information

---

MSCI 240, Algorithms and Data Structures<sup>1</sup>

**Lecture:** MWF 13:30-14:20, E2 1303A

**Tutorial:** W 16:30-17:20, E2 1303A

**Lab:** MTTh 8:30-11:20, as announced in class, CPH 4335

*Prereq: GENE 121 or MSCI 121; Antireq: CS 240, ECE 250, MTE 140 and SYDE 223*

## Instructor

---

Mark Hancock, PhD

Assistant Professor

Office Hours: F 11:30-12:30, starting Sep 9 (CPH 3633)

E-mail: [mark.hancock@uwaterloo.ca](mailto:mark.hancock@uwaterloo.ca)

## Teaching Assistants

---

Akanksha Mahajan

Office Hours: TBA

E-mail: [a7mahaja@uwaterloo.ca](mailto:a7mahaja@uwaterloo.ca)

Ishita Kathuria

Office Hours: TBA

E-mail: [ikathuri@uwaterloo.ca](mailto:ikathuri@uwaterloo.ca)

## UW Learn

---

We will use UW-Learn for email messaging to the class, distribution of materials, and other uses. You are responsible for email messages sent to you at your official UW email address.

## Textbooks

---

Stuart Reges and Marty Stepp (2013). *Building Java Programs: A Back to Basics Approach*, 3<sup>rd</sup> edition, Pearson Education. ISBN: 978-0-13-336090-5 (required)

Robert Sedgewick and Kevin Wayne (2011). *Algorithms*, 4<sup>th</sup> edition, Addison-Wesley Professional. ISBN: 978-0-321-57351-3 (recommended)

---

<sup>1</sup> Portions of this document are copied with permission from Carol Hulls (GENE 121) and Mark Smucker (MSCI 240)

## Calendar Description

---

Design and analysis of data structures and algorithms. Mathematical and conceptual analysis of algorithms for set operations, sorting, graphs, and priority queues. Comparison of algorithms on different data structures. Algorithmic and data structural solutions to common engineering problems in computer science. Mathematical analysis of space and time complexity as well as other forms of computational complexity in algorithms.

## Course Objectives

---

By the end of this course, students should be able to:

1. Understand and use abstraction to separate interface from implementation as demonstrated by implementing abstract data types as Java objects. Implementations include both array based and link/pointer based techniques. Implementation requires skill with a modern integrated development environment and the ability to use a modern symbolic debugger.
2. Analyze the performance of solutions:
  - a. Empirically measure and analyze the efficiency of a computer program using both internal timings and a professional quality code profiler. Analysis involves understanding variance of timings and the resolution of the timer. Variance is largely caused by varying inputs to the algorithms and to a lesser extent by the time-sharing nature of modern computers.
  - b. Analyze algorithms and predict their ability to scale with increasing input sizes. Requires understanding of big-Oh notation.
3. Select appropriate data structures and algorithms for programming tasks. To make good selections requires knowledge of the algorithmic complexity of operations given the data structure and algorithm and knowledge of the tradeoffs in time, space, implementation and maintenance.

## Lectures

---

You should come to all lectures. The lecture will be your primary source for this course with the text (and the multitude of other texts and the web) as your supplementary sources. Lecture notes will not be provided. You must obtain lecture notes from fellow students if you miss lecture. The instructor's notes are for his use and are not suitable for students.

You are responsible for all announcements made in lecture. **Announcements made in lecture may not be made in duplicate on UW-Learn. In other words, do not rely on UW-Learn for all announcements.**

On two occasions, lecture will be swapped with tutorial.

The following tutorials will be **lectures**:

1. Wednesday, September 14 @ 16:30-17:20
2. Friday, October 14 @ 16:30-17:20 (Wednesday schedule after Thanksgiving)

The following lectures will be **tutorials**:

1. Monday, November 7 @ 13:30-14:20
2. Wednesday, November 9 @ 13:30-14:20

## Tentative Lecture Schedule

---

1. Java classes, generics. (BetterArray/SimpleListOfInt and List<T>)
2. Abstract data types. Examples: queues and stacks implemented with arrays and linked lists. Java interfaces, implements.
3. Empirical analysis of algorithms (examples: linear search, insertion sort, selection sort, and growing storage of SimpleListOfInt)
4. Analytical analysis of algorithms, algorithm complexity and asymptotic behavior, big Oh. (Same examples as used for empirical analysis)
5. Understanding the key built-in Java collections and algorithms:
  - a. List based: List<T>, Queue<T>, Stack<T>, LinkedList<T>. (Details largely covered earlier)
  - b. Hashtables: HashSet<T>, Map<TKey, TValue>, Hashtable<TKey, TValue>. How do hashtables work? Most common implementation with slots and chaining.
  - c. Binary Tree: SortedMap<TKey, TValue>. How do binary trees work? Balanced trees?
  - d. Algorithms: indexOf, binarySearch, sort. indexOf is linear search and already covered. The algorithm behind binary search, and then the algorithms behind sort:
    - i. Insertion sort
    - ii. Recursion (needed to understand quicksort and mergesort)
    - iii. N-log-n sorting: Quicksort, heapsort. (Touches on priority queues implemented with heaps.)
6. Mergesort and its importance for out of memory sorting.
7. Graphs. Basic properties and breadth-first and depth-first search.
8. Priority queues implemented with heaps
9. String searching (if time permits)

## Grading

---

### Clicker/REEF

You are required to have a clicker/REEF and bring it to all lectures, tutorials, and labs. You can purchase a clicker and REEF licenses from the bookstore. You must register your class via Learn.

You will earn participation points for the use of your clicker. Each clicker question answered is worth 1 point. Expect every lecture to be a chance to earn clicker participation points.

### Exercises, Projects, Exams, Etc.

There will be near weekly homework (exercises or projects) assigned. Homework not handed in on time will be late and receive a zero. From each set of exercises, only a few (approximately 3) exercises will be graded. The tutorial will utilize in class work and "sports day." Details regarding sports day will be provided later. There will be 4-5 programming assignments.

There will be a midterm and a final exam.

**You are expected to be available for all exams. Any unexcused absence from an exam will result in a zero. All excuses must be approved in advance of the scheduled exam.**

## Grade Breakdown

The breakdown of grade assignments is as follows:

<b>Assignments (35%)</b>	
<b>Labs (4-6 marked labs)</b>	15%
<b>Exercises (4-7 assignments)</b>	
<b>Tutorials/Sports Day/Clicker</b>	
<b>Programming Assignments/Projects (4-5, each worth equal amount)</b>	20%
<b>Exams/Quizzes (65%)</b>	
<b>Midterm</b>	20%
<b>Final Exam</b>	45%

The instructor reserves the right to modify the weighting of all course components in the final computation of the course grade.

In general, all participation points (largely the clicker, tutorials, labs, quizzes, and sports day) are put together and will count as one additional assignment.

All grades are final one week after work has been returned. If you have an issue with graded homework, please see the TA first. If you cannot resolve your problem with the TA, please see the instructor.

All graded material will be returned in class, tutorial, or lab. The final exam is not returned.

## Course Policies

---

### Late Policy/Assignment Extensions

Assignment extensions will **ONLY** be granted for illness or for domestic affliction. Documentation by a health professional is required. If there are extenuating circumstances which prevent you from submitting your assignment by the due date, see your instructor **before** the due date in order to make special arrangements. Requests after the due date will be denied except for extremely serious life events (e.g. you are in the hospital unable to reach the instructor).

Application for an assignment extension **MUST** be made **IN PERSON** with the instructor. Any email requesting an extension to an assignment will be ignored. **TAs do not have the authority to extend assignment deadlines.**

### Labs

Labs do start at 8:30am. Failure to be on time will result in being excused from the lab. Repeated late arrival will result in escalation of the issue to the instructor and/or the Management Sciences Associate Chair of Undergraduate Studies.

If you need to miss a lab, you can switch your place with a student in another lab. You may not attend other lab sections without arranging a switch with another student and informing both the TA and the instructor of the switch in advance via email.

In summary:

1. You must find another student with which to switch. The only exception to this rule would be illness or other situations handled on a case by case basis. If you are seeking an exception, you must receive approval from the instructor. The TA cannot grant the exception.
2. You must email the TA and the instructor in advance with your plan to switch. Please cc the other student as well on the email.

Failure to meet 1 and 2 will result in a zero for the missed lab, and you will not be allowed to attend lab.

### Etiquette / Rules

We have some simple rules that are all based on respect for the educational process, respect for the privilege and honor to attend university, and respect for others and yourself. Failure to follow these classroom rules will result in a 3 stage escalation process:

1. Warning. The instructor will remind you to follow the rule in question.
2. Leave the classroom. The instructor will ask you to leave the classroom for the remainder of the day's lecture or tutorial.
3. Letter to Management Sciences Associate Chair of Undergraduate Studies. The instructor will ask you to leave the classroom and will also send a letter to the Management Sciences Associate Chair of Undergraduate Studies detailing the discipline problem.

The classroom rules are:

Rule 1: Be on time.

Rule 2: Raise your hand to speak.

Rule 3: No distractors. Details: No distracting materials. This includes all non-class materials. For example, working on homework for another class. Reading a newspaper. Laptops, iPods, phones, etc. are all banned. In tutorial, you may not use the computers for anything other than MSCI 240: no Facebook, no web browsing, no checking email, no instant messaging, no games, etc. You cannot do two things at once. You must reserve 50 minutes each lecture for MSCI 240.

Outside the classroom etiquette:

1. Do not contact the instructor by phone except in the case of emergency.
2. Do not "drop in for a quick question". All office visits to the instructor should be via appointment or during office hours.
3. Keep the computer lab clean.
4. Take lengthy conversations outside of the computer lab. Be aware that the room adjacent to the computer lab is used as a classroom and meeting room.
5. Do not leave your computer locked in the computer lab for extended periods of time.

## Academic Integrity

It is expected that within this course, the highest standards of academic integrity will be maintained, in keeping with UW's Policy 71, "Student Academic Discipline Policy." While all aspects of Policy 71 apply to all UW courses, of particular relevance to MSCI 240, which involves various types of student efforts, it is each student's responsibility to avoid:

- **cheating**, which "includes copying from another student's work or allowing another student to copy from one's own work, submitting another person's work as one's own, fabrication of data, consultation with any unauthorized person during an examination or test, and use of unauthorized aids" (UW Policy 71).
- **plagiarism**, "which is the act of presenting the ideas, words or other intellectual property of another as one's own...properly acknowledged use of sources is an accepted and important part of scholarship" (UW Policy 71).

**All referenced work in reports and presentations must be appropriately cited, to include websites, as well as figures and graphs in presentations.** If there are any questions whatsoever, feel free to contact Prof. Hancock about any possible grey areas.

**Students are strongly encouraged to review UW's Academic Integrity "For Students" webpage.** A link can be found on the UW-ACE site homepage and under *Course Resources* → *Academic Integrity*.

Key points:

- All work must be your own.
- All assignments are to be completed by yourself.
- You must acknowledge assistance from the web, books, and others. No material may be copied from any source.
- You may not provide answers to other students.
- Turning in the work of others as your own; failure to acknowledge sources and assistance and discussions; and providing answers to others will be punished.
- We will report all allegations of academic misconduct (cheating) to the Associate Dean of Undergraduate Studies Engineering, who will conduct an investigation following the rules of Policy 71.
- Typical "first time offender" punishment for cheating on an assignment is a zero on the assignment, 5% reduction in final course grade, and a letter of reprimand or academic probation.

You may discuss assignments with other students, but you may not give answers. "Giving answers" includes, but is not limited to, the following:

- Jim asks Betty "What did you get for problem 5?" If Betty tells Jim, both she and Jim have violated these rules.
- Sarah asks Mohamed "How did you solve this part of the program?" If Mohamed tells Sarah, both have violated these rules.

- Fred gives Wilma a copy of his code.
- Betty gives Barney her assignment “to look over.”

If you do not know how to discuss work without giving answers, direct the other person to see the TA or the instructor for help.

If you do discuss work with others, what do you need to do to be academically honest (ethical)?

1. Attempt to do all work by yourself first. If you solve a problem using only your brain and hard work, then you have no one else to acknowledge.
2. Best practice to follow:
  - a. Work at a whiteboard with people.
  - b. No one is allowed to take notes.
  - c. Erase the whiteboard when done talking.
3. Discuss with others, but never give or share answers and never copy. Talking about problems with others and helping others debug their code can be a great way to learn material better and faster. You can also turn to original sources and other books, but you must do ALL of the following (a, b, c, and d):
  - a. You must put away all group notes, books, papers, internet webpages, etc. and write up your own solution. If you cannot write up a solution from start to finish without looking at notes, then the work is not your own, but someone else’s. Go back, study the problem some more, then try to write up your solution. Repeat until you can do it without using the notes or other sources.
  - b. You must acknowledge your sources whether a written source or a person. It is often obvious to your instructor and TAs when you have copied an answer, no matter how obscure you might think your source is. You must acknowledge anyone you discussed a problem or assignment with.
  - c. You must also acknowledge the nature of your source. If you uncovered a solution to a problem, you must acknowledge that!
  - d. Write your acknowledgments either at the beginning or end of your problem write-up.

You will be asked to sign each assignment stating that it is your original work, except for the assistance that is stated in the acknowledgements. As an example of a possible consequence, the penalty for plagiarism on an assignment (assuming no previous academic offences in any course) includes a letter placed in your file describing the offence, zero on the assignment, and 5% subtracted from your overall course grade. (See <http://www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm> for more examples.)

For work involving computer code (e.g. Java, C#, Visual Basic, etc.), your work may not contain the code, or portions of code, of another person or book or website etc. Discussions of computer programs should be at the level of pseudo-code and not at the level of real computer code.

## Examples

NOT ALLOWED UNDER ANY CIRCUMSTANCES:

- Copying code or code fragments. There is no need to copy any code in this course from any source.
- Copying even one line of source code.
- Copying even one line from some published source (web, book, etc.) without quoting it and providing a proper citation.
- Submitting code that has only minor modifications of another's work, e.g. changing variable names.
- Providing code or code fragments (by paper or electronically) to another student (or group). Showing another student your assignment.
- Writing any part of another student's assignment (even one line of code) for them.
- Submitting output (screen captures) that was not generated by the code submitted.
- Failing to acknowledge help received or given in the completion of homework and projects.

## Attendance Policy

Any circumstances that prevent you from studying for more than a few days should be reported. If you miss a deadline for health reasons, a Verification of Illness form (available at [uwaterloo.ca/health-services/student-medical-clinic/services/verification-illness](http://uwaterloo.ca/health-services/student-medical-clinic/services/verification-illness)) should be submitted to the instructor.

You are expected to write the quizzes and final exams as scheduled unless you have contacted the instructor before the exam about a schedule conflict. Failure to write an exam or quiz without sufficient reason (e.g. severe illness) will result a grade of zero for the examination.

## Other Important Notices

Responsibilities: All students must be familiar with their responsibilities as outlined in this syllabus and as outlined by the Faculty of Engineering and the university. Please see the following for more details: [uwaterloo.ca/engineering/current-undergraduate-students/academic-support/course-responsibilities](http://uwaterloo.ca/engineering/current-undergraduate-students/academic-support/course-responsibilities)

Grievance: A student who believes a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70, Section 4, [uwaterloo.ca/secretariat-general-counsel/policies-procedures-guidelines/policy-70](http://uwaterloo.ca/secretariat-general-counsel/policies-procedures-guidelines/policy-70). When in doubt please contact the department's administrative assistant for further assistance.

Discipline: A student is expected to know what constitutes academic integrity (check [uwaterloo.ca/academicintegrity/](http://uwaterloo.ca/academicintegrity/)) to avoid committing an academic offence, and to take responsibility for his/her actions. A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about "rules" for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate Associate Dean. For information on categories of offences and types of penalties, students should refer to Policy 71, Student Discipline, [uwaterloo.ca/secretariat-general-counsel/policies-procedures-guidelines/policy-71](http://uwaterloo.ca/secretariat-general-counsel/policies-procedures-guidelines/policy-71). For typical penalties check Guidelines for the Assessment of Penalties, [www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm](http://www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm).



Appeals: A decision made or penalty imposed under Policy 70 (Student Petitions and Grievances) (other than a petition) or Policy 71 (Student Discipline) may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72 (Student Appeals) [uwaterloo.ca/secretariat-general-counsel/policies-procedures-guidelines/policy-72](http://uwaterloo.ca/secretariat-general-counsel/policies-procedures-guidelines/policy-72).

Note for Students with Disabilities: AccessAbility Services (formerly known as the Office for Persons with Disabilities or 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 AccessAbility Services at the beginning of each academic term [uwaterloo.ca/disability-services/](http://uwaterloo.ca/disability-services/)

Health Services provides physical and mental health care: 519-888-4096 or [uwaterloo.ca/health-services](http://uwaterloo.ca/health-services)

**YOUR SIGNATURE ON YOUR ASSIGNMENT INDICATES YOU UNDERSTAND AND HAVE COMPLIED WITH UW POLICY.**