

UNIVERSITY OF WINDSOR
SCHOOL OF COMPUTER SCIENCE
03-60-141-01 Winter 2014
Introduction to Algorithms and Programming II
Course Outline

INSTRUCTOR:

Dr. Robert D. Kent

Email: rkent@uwindsor.ca

Office: 5100 Lambton Tower

Phone: 519-253-3000 ext. 2993 (Email preferred)

Office Hours: Tuesday, Wednesday, Thursday 01:00pm – 2:20pm (or by appointment)

Note: Only email originating from a valid University of Windsor student account will be accepted from students wishing to contact the instructor. Include your full name and student ID in your correspondence.

PRE-REQUISITES:

You cannot register in this course unless you have taken the following courses:
60-100 (or 62-140) and 60-140 with a minimum passing grade in each course.

LECTURES:

Monday and Wednesday, 2:30pm – 3:50pm – Toldo Health Education Centre 200

LABS:

Students registered in Lecture Section 01 must register in one of the following Laboratory sections:

Section **51** - Tues 01:00pm – 02:20pm West Library 305 TBA

Section **52** - Tues 02:30pm – 03:50pm West Library 305 TBA

Section **53** - Tues 04:00pm – 05:20pm West Library 305 TBA

Section **54** - Tues 05:30pm – 06:50pm West Library 305 TBA

Section **55** - Tues 04:00pm – 05:20pm Erie Hall 3119 TBA

Section **56** - Tues 05:30pm – 06:50pm Erie Hall 3119 TBA

NOTE: Laboratory Instructors and GA staff will be announced in the lecture, labs and on the course website.

Attendance is mandatory. Labs start the week of January 13-17. You must have a working student computer account before coming to the lab. You can obtain a UWINID account from the Computer Centre or call ITS ext. 4440.

Any questions relating to laboratory should be addressed directly with Dr. Kent or your lab instructor.

COURSE

DESCRIPTION:

60-141. Introduction to Algorithms and Programming II

This course is the continuation of 60-140 that introduces students to more advanced algorithm design and programming in a high level language such as C. The main objectives of the course are to develop the ability to identify, understand, and design solutions to a wide variety of problems. Topics covered include: multi-dimensional arrays, pointers, strings, advanced modular programming, records, binary files, recursion, stacks, linked lists and introduction to algorithm analysis. (Prerequisite: 60-100 (or 62-140) and 60-140.) (3 lecture, 1.5 laboratory hours a week; plus unsupervised study and work on individual assignments.)

REQUIRED

TEXTBOOK:

Deitel, H.M. and Deitel P.J. C How To Program, 7th Edition. Prentice-Hall: New Jersey, 2012.

Earlier editions of the above text may be used for reference and study purposes, but all lecture references to material in the text will be made using the listed edition above.

SUPPLEMENTS:

A course website has been established with URL:

<http://rkent.myweb.cs.uwindsor.ca/cs141/>

NOTE: Students may obtain lecture slides and past copies of some examinations. Students must

note that the website is certainly not a substitute for attending lectures, completing assigned readings, programming and general practice. Students must not rely on previous examinations as sole study aids – examinations are created anew for each semester.

GRADING SCHEME:

The University of Windsor uses a percentage marking and grading scale. The following are the university-wide grade descriptors for undergraduate programs and will be printed on the back of student transcripts.

% Score	Grade
90.0-100	A+
85.0-89.9	A
80.0-84.9	A-
77.0-79.9	B+
73.0-76.9	B
70.0-72.9	B-
67.0-79.9	C+

% Score	Grade
63.0-66.9	C
60.0-62.9	C-
57.0-59.9	D+
53.0-56.9	D
50.0-52.9	D-
0-49.9	F

Students must understand that the professor reserves the sole right to adjust the marks of the entire class at the end of the semester and before submitting grades to the Registrar. If such adjustments are made, they are made by adding a constant value to all student total marks before calculating the letter grade.

For a more detailed explanation, students are directed to the webpage: www.uwindsor.ca/calendar and from there, follow the link to **Examinations, Grading and Graduation** and then to the link **MARKS/GRADES DESCRIPTORS**.

COURSE EVALUATION:

20% Midterm #1 (Wed., Feb. 12 – in class)
 20% Midterm #2 (Wed., Mar. 19 – in class)
 4% Final Exam (Friday, April 11, 12:00pm – 3:00pm. Location TBA by Registrar)
 10% Laboratory (Mandatory Attendance, Lab Exercises)
 10% Assignments (5 assignments, equally weighted, every 2 weeks)

NOTE: Consistent with University Senate Bylaws, no marked work will be assigned to students during the last week of classes, nor will any assigned work come due for submission during the last week of classes. It is possible, even likely, that a final assignment may become due for submission immediately following the Final Examination itself.

Course Schedule:

<u>Week</u>	<u>Topic*</u> <small>(*The instructor reserves the right to change the outline to accommodate student pace in understanding of the subject matter)</small>	<u>Assigned Reading</u>
1	Modular Programming: Recursive algorithms, Functions in C (no labs assigned – review 140 labs as self-directed learning)	Chapters 1 to 5
2	Modular Programming: Recursive algorithms, Functions in C Labs Begin	Chapter 5
<i>Friday, January 17, 2014</i>	<i>Last day for late registration and change of course. Last day for full tuition refund.</i>	
3	Multi-scripted arrays; Algorithms: Sort, Search, Problem Solving	Chapters 6, 7
4	Algorithms: Sort, Search, Problem Solving	
5	Pointers, Characters and Strings Standard Library Functions	Chapters 7 and 8

6	Pointers, struct	Chapter 10
	<i>Study Week – Saturday, February 15 to Sunday, February 23, 2014</i>	
7	Abstract Data Types and Dynamic Memory Concepts Midterm Exam #1 (chapters 1-9) Wed. Feb 13, 2013	Chapter 10
8	Static Data Structures	Chapter 12
9	Dynamic Data Structures	
<i>Friday, March 14, 2014</i>	<i>Last day to withdraw voluntarily from courses. After this date students remain registered in courses and receive final grades as appropriate. Last day for partial tuition refund.</i>	
10	Sequential Text Files Midterm Exam #2 (up to chapter 10) Wed. Nov. 14, 2009	Chapter 11
11	Random Access Files	Chapter 11
12	Applications and Advanced Problem Solving in C. Review. <i>Student Evaluation of Teaching (SET)</i>	Chapter 13
<i>Friday, April 04, 2014</i>	<i>Last day of classes</i>	
Friday, April 11, 2014	Final Exam 12:00-03:30PM (Midday) – Location TBA by Registrar	

NOTE: Chapters 9 and 13 must be read and studied by the student. This material will not be formally discussed in the lectures, but questions should be asked in the lecture, labs or during office hours.

The Student Evaluation of Teaching (SET) will be conducted during the last 2 weeks of class.

CAUTION: This course assumes the student will allocate a significant amount of independent study and time spent on coding programs in C. You are strongly encouraged to ensure that sufficient time is allocated in order to succeed in this course.

NOTE: Students are advised that courses may be taken only twice to obtain credit. If a passing grade is not obtained upon the second attempt students will normally be required to withdraw from the program of study.

Notes to Students:

- Attendance and student participation are essential to succeed in this course. You are encouraged to ask questions.
- A website is setup for the course. This should be regarded as a complementary source of information and not as a primary reference. It is the responsibility of the student to attend classes and keep up with the latest course contents and announcements. Note that the website is not guaranteed to be up-to-date or accurate. Any student who relies solely on the website information, contrary to this warning notification, may place him/herself in academic jeopardy.
- Students are encouraged to keep class notes in good order, repeat the examples demonstrated in class, and ask questions. Solving exercises on your own and participating in class are very important to succeed.
- Assignments and projects are expected to be completed on the assigned due date and time. **LATE ASSIGNMENTS ARE NOT ACCEPTED.** You must allocate enough time to complete the assignments; start early and report difficulties to the instructor. **UNDOCUMENTED CODE WILL NOT BE GRADED** and will receive a mark of zero. Failure to submit the work in the required format will be penalized. (i.e. incorrect email subject or unreadable/missing file attachments as instructed, etc...).
- **THERE IS NO MAKE-UP EXAM FOR MIDTERMS.** Missing a test or an exam will greatly affect your grade since they carry a lot of weight. In the case of illness or serious and unavoidable reason (as per the Senate Bylaws), please consult with the instructor in advance if possible to make alternate arrangements. You must formally inform the instructor in writing and present proper supporting documents (stating that the student was unable to attend the exam at the specified time and date) within a week from the midterm, or earlier, to be verified. If the reason is deemed valid, the weight of the midterm will be added to that of the final exam. (i.e. the final

exam's weight will increase to include the missed midterm weight). In cases where a midterm examination has been missed for valid excuse, all students are required to write and PASS the final exam in order to pass the course.

- You will need access to a C compiler and a university email account. You need a UNIX (UWinID) student account (from ITS x4440 or helpdesk@uwindsor.ca for support). Programs may be developed and tested using the Computer Science server machines and all programs submitted for marking MUST run on these Computer Science machines. Do not use proprietary MS Windows compatible C compilers and editors as they do not generate ANSI compatible source files, in general. Linux based machine configurations should be able to use standard GNU compilers and editors.
- **PLAGIARISM:** Should the instructor or grader find a reason for suspicion – or just cause – in plagiarized student work (assignment, lab or test), the work in question will not be graded and the student(s) will have to answer to the department's Director and the Dean of Science. Refer to the University's policy on Plagiarism in the Senate Bylaws. This issue is taken extremely seriously and students who are found guilty of plagiarism may suffer considerable short and long term negative effects.
- **Students who submit semantically equivalent assignments [or labs] (in other words, the same thing to within trivial modifications) will receive a grade of zero on each such assignment [or lab].**
- All students should keep the instructor informed about their difficulties with the course.
- To the extent possible, students should contact the instructor outlining their problems with the course. Most consultations will be by e-mail or in person.

On Accepting Responsibility

Students must adapt to the severe change in responsibility that occurs in the University setting versus the High School setting.

Each student is responsible for ensuring that a good study schedule is developed and adhered to, that all assignment deadlines are met, that all course requirements are fulfilled and that excuses are not accepted (except for medical or some few personal reasons, such as death in the family). Lectures are a necessary part of the learning process, but they are not the whole story – each student must accept responsibility for extensive reading and study beyond the lecture material. Web sites may contain lecture notes and other helpful materials, but these may not be available prior to lectures and may contain errors. One of the most important lessons to be learned from a successful program of study at University is the ability to self-direct learning. At some point your formal education will be completed and then your informal and continuing education proceeds throughout the rest of your career. Learn how to use Library facilities in addition to exploring the Web. Be forewarned that websites cannot be guaranteed to contain factual and accurate information. Be advised that unless you are able to tackle and solve a problem completely by yourself, you do not understand what is expected, even demanded, to be called a professional or competent programmer.

TIP: To succeed in this course you need to PRACTICE! Allocate a **minimum of 2 to 6 hours a week** on reading and code development!