YORK UNIVERSITÉ UNIVERSITY

Introduction to Computer Science II

COSC1030.03 -- Fall term 1997/98

Please remember to reload this page occasionally since it will be updated frequently and you will want to replace the version in your cache.

Last modified: 1997 Nov 25, 5 pm

NEW I've added a clarification regarding procedure ModifyElement in Assignment #4. See the Assignments section (under Assignment 4) for details.

NOTE:

On Assignment #3, a number of students ''pulled a rabbit out of their hat'' in the proof of correctness. That is, they came up with a loop invariant by some magical means, and clearly did not understand (a) why the expression was an invariant; (b) how to come up with the invariant in the first place; and (c) how the invariant is to be used in doing proofs. Some may have found the invariant in a book, or got the invariant from a friend. Such techniques will **NOT** be of any help on the final exam. We strongly suggest that you spend some time on working with invariants in preparation for the exam.

Save paper. Do not blindly print files. Click here to get a Postscript version of this WWW page that prints two logical pages per physical page.

Contents

Instructors

- Richard Paige -- Course director -- Section A

Course Outline

Click here to read the course outline that was handed out in class (in HTML).

Class Times and Locations

```
Section A  Mon, Wed, Fri 5:30-6:30    121 CCB
```

Facilities

COSC 1030 will make use of the Ariel computing facilities in CCB 164. See the WWW page for Ariel for details on fair use of Ariel, hours, and available tools.

Unlike COSC 1020, this course does not have blocks of time in Ariel reserved for you. Therefore, you will have to make use of Ariel's facilities in a first-come first-served manner.

Office Hours:

```
Monday     3:30 -  4:30  250 CCB  R. Paige
Tuesday   11:30 - 12:30  152 CCB  J. Huang  *starting Oct 6
Wednesday 11:30 - 12:30  152 CCB  J. Huang  *starting Oct 6
Wednesday  3:30 -  4:30  250 CCB  R. Paige

Further office hours for the 1030 course TA will be announced (there will
be more hours the week before an assignment is due).
The TAs in the lab CCB 164 are strictly for 1020 students.
```

Course Description

This course is a continuation of COSC1020 and covers some of the fundamentals of software development, various data structures (arrays, queues, stacks, trees, lists), and algorithms that use these structures (sorting, searching). An object oriented approach will be introduced. Students will use the Unix operating system with the X Window System.

Required Text:

- **Data Structures: An Object-Oriented Approach** , Barnard, Holt, and Hume, Holt Software Associates Inc., 1995.

Suggested Reference:

- **Turing Reference Manual**, R.C. Holt, T. West, 6th edition, Holt Software Associates, Inc., 1995. You may be able to find this at the computer store in York Lanes, or at the University of Toronto Bookstore at College & St. George downtown. See the section Obtaining OOT Software for HSA telephone numbers.

Useful books:

- Koffman, Holt and Dimarco, **Turing, Problem Solving and Program Design**, Addison-Wesley, 1995.
- Grogono, **Programming with Turing and Object Oriented Turing**, Springer Verlag, 1995.
- **The Little Grey Book: An Ultrix Primer**, Digital Press, 1990.
- **X Window System User's Guide**, O'Reilly & Assoc., May 1993.
- G. Polya, **How to solve it: a new aspect of mathematical method**, Princeton Univ. Press, 1973.
- J. Bentley, **Programming Pearls**, Addison-Wesley, 1989.
- B. Kernighan and P. Plauger, **The Elements of Programming Style**, McGraw-Hill, 1978.
- B. Kernighan and P. Plauger, **Software Tools in Pascal**, Addison-Wesley, 1981.
- F.P. Brooks, **The Mythical Man-Month: 20th Anniversary Edition**, Addison-Wesley, 1995.

Extra Reading

This section contains extra reading material that is mentioned in class. Postscript files can be read on Ariel by using *ghostview* (read the man-page for *ghostview* for all its options). You can copy the Postscript files to your home directory by holding down left-shift, and right-clicking on the link. You can print these files by using *lpr*.

These files are also in the course directory, `/cs/course/1030`. Look in the subdirectory `notes`.

- How pointers work (written for C). Please read this document online!
- A few slides used in the first lecture (Sept. 8) in 4-up Postscript
- Testing slides from Sept. 10 lecture, plus one more that slipped through the cracks.
- Slides on pointers used by Tim Brecht during week of Sep. 15-Sep 19.
- More example slides used by Tim.
- Pointer examples from previous years.
- Tips on programming style and testing
- Some review questions on proof

Suggested Reading

Each week, readings from the textbook will be suggested. You are advised to go over the material carefully, since it will be fair game for assignments and tests. Also, by reading the material in advance of it being covered in lecture, you will be better prepared to listen in lecture, and to ask questions.

Please note that the reading list from 1996/97 is **NOT** the same as that for this term; they covered things in a different order. However, you can look at the 1996/97 reading list to get an idea of what might be covered in the future (providing that you can rearrange things in your mind).

- *Readings for Week 1:* Chapters 1 and 2 of Barnard et al., in preparation for Week 2 of classes.
- *Readings for Week 2:* Chapter 4, the material on Abstract Data Types, and Chapter 8, the material on object-orientation. Specifically: Section 4.1-4.3, and Section 8.1-8.4 (though we won't cover all of 8.1-8.4 in lectures).
- *Readings for Week 3:* all of Chapter 4, and Sections 8.1-8.3.
- *Readings for Week 4:* Chapter 3, first few sections, up to 3.3.
- *Readings for Week 5:* Chapter 3, up to and including 3.4. Review of induction would also prove useful.
- *Readings for Week 6:* 3.6-3.8. You should read 3.5, too, but you are not responsible for understanding the binary search algorithm for the midterm. In preparation for next week, look at Section 3.9.
- *Readings for Week 7:* Section 3.9, on timing and big-O.
- *Readings for Week 8:* Chapter 5, on stacks.
- *Readings for Week 9:* Chapter 9, on recursion.
- *Readings for Week 10:* Chapter 10, on trees. We may go briefly into parts of Chapter 11 as well.

Course Goals

Students successfully completing this course should be able to demonstrate the following knowledge and/or skill sets:

- Develop a correctly working, well structured, readable, maintainable and documented program that meets the given specifications and is completed on time.
- Ability to design, implement, test, and debug functions and procedures as well as the program as a whole.
- Experience with larger programs (larger than those seen in 1020).
- Ability to perform algorithm analysis and to be able to compare the time and space tradeoffs of various algorithms.

- Be comfortable writing, testing, debugging, and analyzing the complexity of recursive functions and procedures, as well as understanding how recursion works.
- Thorough understanding of what pointers are, how they are used, and when they should and should not be used.
- UNIX skills including: the use of an editor, high-level understanding of what separate compilation is and how to compile and link separate components of a program.
- General understanding of what lists, stacks, queues, priority queues, dictionaries, trees and graphs are, how they work and how they can be implemented.
- Different sorting methods their pros and cons and time versus space tradeoffs.

Evaluation

There are four assignments, one midterm and a final exam. Each piece of work will receive a letter grade (A+, A, B+, B, C+, C, D+, D, E or F). A final grade will be obtained by weighting the individual pieces of work as shown in the following table.

Work	Date	Weight	Topic
Assignment 1	Mon, Sep 29	7%	Testing and ADTs
Assignment 2	Mon, Oct 20	10%	Pointers and classes
Midterm Test	Mon, Oct 27	25%	(50 minutes, in class)
Assignment 3	Mon, Nov 10	8%	Stacks and theory
Assignment 4	Fri, Dec 5	10%	Trees, pointers, recursion
Final Exam	after Dec 8	40%	Entire Course

NOTE: You must obtain a weighted GPA of 2.0 on each component of the course (exams and assignments) to pass the course. That is, you must obtain a weighted GPA of >= 2.00 on the assignments and you must obtain a weighted GPA of >= 2.00 on the exams to pass the course.

NEW A grades list, updated November 18, 1997, is now available through the **courseInfo** command. Log into Ariel, and type the command

```
courseInfo 1030 A
```

which will output *only your* list of marks for COSC 1030, Section A. You cannot see any marks but your own using courseInfo. Note that you can view your marks **ONLY** while logged into Ariel.

If you find any mistakes in your marks list, please contact the instructor, and make available all the original marked material. If you find that you cannot see your marks, try again to make sure that you haven't made a mistake in typing. Otherwise, contact the instructor.

The following York University standard letter grade system will be used.

Grade	Definition	Point Value
A+	Exceptional	9
A	Excellent	8
B+	Very Good	7
B	Good	6
C+	Competent	5
C	Fairly Competent	4
D+	Passing	3
D	Barely Passing	2
E	Marginally Failing	1
F	Failing	0

Click here to read more details on the meaning of grades.

Important dates

First day of classes:	Monday, September 8
Last day to enrol in the course:	September 20
Last day to drop the course:	November 8
Last day of classes:	Friday, December 5
Examination period:	starts December 8

Assignments

Assignments represent your own work. Do not misrepresent another's work as your own; it will be considered academic dishonesty if you do. Some guidelines are given in the ASCII document On Academic Honesty. All students should be familiar with the section on academic honesty in the university calendar. If you have any questions about this please ask your course instructor.

Assignments are to be handed in at the start of class (17:30). Assignments handed in **after** the start of class, but before the start of the next class, will have two full letter grades deducted (e.g., A becomes C, B+ becomes D+, and C+ or lower becomes F). (The exception to this rule is for Assignment 4, which is due on the last day of class; **NO LATE ASSIGNMENT 4s WILL BE ACCEPTED!**) After the start of the next class, NO late assignments will be accepted, with the sole exception being if you give your instructor a signed medical note and ask for permission to be late. In this case, you must hand in the medical note and late assignment to your instructor. Please note that events such as last minute printer slowdown or breakdown DO NOT constitute valid excuses for lateness.

Course assignments require you to prepare and hand in reports. These must be readable and, for programming assignments, should be prepared in a manner consistent with the **How to do a report** as outlined in the laboratory book for COSC1020. If you do not have a copy, please see the course directory /cs/course/1020. The goals for assignments in COSC1030 differ from those in COSC1020, as a result your reports will have a different structure. Follow this link for a general guideline on What is a Report?

We will provide cover sheets for your assignments that you can print. Please include them as the **very first page** of your report or else you will lose one full letter grade (e.g., A becomes B, C+ becomes D+, and D or lower becomes F). **NO BINDERS OR ENVELOPES**.

REMEMBER: That in order to receive a passing grade in the course your weighted GPA on each component must be >= 2.00. That is, the weighted GPA on the assignments must be >= 2.00 and the weighted GPA of the midterm and the exam together must be >= 2.00 to pass the course.

Throughout this course and for all assignments the following will be stressed:

- clear, well organized, divided into sections and subsections.
- grammatically correct writing (including spelling)
- proper program design
- good implementation style and techniques
- debugging
- demonstrating correctness using proofs and testing

Assignments

Assignment 1: Sets and ADTs

NOTE!: Several students have pointed out two small bugs in the definition of the operations *Remove* and *IsElement* in the file sets.t. These have been corrected, and the new version is on the WWW page here, and in the course directory /cs/course/1030/a1 on Ariel.

A solution to Assignment 1 has now been posted.

- Assignment handout in Postscript
- Assignment handout in HTML
- Repaired file sets.t
- Cover page for Assignment 1 in Postscript
- Solution to Assignment 1, Sections 4 and 5 (for a solution to Section 3, simply replace "module" with "class" in the given file). Click here to see sketch solutions to Section 2, plus the two other questions in Sections 3 and 4.

Assignment 2: Pointers and ADTs

- Assignment handout in Postscript (use *lpr* to print this on Ariel, and *ghostview* to view it)
- Assignment handout in HTML (this is viewable using Netscape or Explorer)
- Cover page for Assignment 2 in Postscript
- Partial solution to Assignment 2

Assignment 3: Theory and Practice

- Assignment handout in Postscript
- Assignment handout in HTML (this is viewable using Netscape or IExplorer)
- Cover page for Assignment 3 in Postscript
- Solution to Assignment 3, Questions 1-3
- Solution to Double-Ended Stack question

Assignment 4: Trees and Recursion

```
 This was posted to  york.cs.course.1030, which you should already
 read regularly.

---------------------------------------

  I've had a few questions about the purpose of procedure
 ModifyElement, which has the following interface:

  procedure ModifyElement(var tree: TreeType, modElement: TreeElementType)

  ModifyElement searches through the tree, finding a node that
 contains modElement.key. It then changes the info field of that
 node.

  ModifyElement *CANNOT* be used to change keys. It is assumed that
 search keys are inviolate, once they have been added to the tree.
 ModifyElement can be used to change only the *data* associated with
 the node in a tree (data is not the same as the key; the key is used
 solely for searching).

  You may ask: what data? Nodes contain records (of TreeElementType),
 which contain only keys of type integer. So no further data is
 included in the tree nodes. In order for ModifyElement to do anything
 non-trivial, you have to add further fields to TreeElementType.

  To alter keys, you may have to add a new procedure, ModifyKey, which
 is similar to ModifyElement, except it changes only the key field of
 info.

  I suggest that you avoid adding a ModifyKey procedure. It isn't
 necessary. Create non-BSTs by using Mirror, or by creating a BST and
 modifying keys by hand.
```

- Assignment handout in Postscript
- Assignment handout in HTML
- File tree.t
- Cover page for Assignment 4
- Solution to Assignment 4

Reappraising Assignments

If you would like to have an assignment or exam reappraised you must do so within two weeks of it being returned. Note that the **entire** assignment or exam will be remarked and that your grade may go up or down as a result of being remarked.

Before resubmitting ask yourself if it is worth the time required by yourself, the TA, and the professor. For example, suppose you want an extra mark on a question that is worth 1/8 of the assignment and the assignment is worth 7.5% of the final mark. Then if you get the extra mark, your final GPA will change by 1/8 * 0.075 = 0.01. More over, last term the marks of only two out of approximately 25 resubmissions were increased. This means that the expected change to your GPA is 1/8 * 0.075 * (2/25) = 0.00075. Be sure, however, to determine how to do better on the next assignment.

In order to resubmit an assignment for remarking, you must fill out this form and hand it in along with your assignment.

- Reappraisal Form

Midterm test and Final Examination

REMEMBER: That in order to receive a passing grade in the course your weighted GPA on each component must be >= 2.00. That is, you must obtain a weighted GPA of >= 2.00 on the assignments and you must obtain a weighted GPA of >= 2.00 on the exams to pass the course.

Final Exam

The final exam is scheduled for **December 9, 1997** from **3:30-6:30**, in Vari Hall A. Click here to see the Registrar's exam time listing for all COSC courses.

The exam for COSC 1030 will be closed book and notes. No aids will be allowed. Questions on the exam will be similar in style to those on the midterm. There will be 10 questions, and you will have 3 hours.

Example Exams

- Midterm Fall 96-97 You should be aware that some of the material on this midterm may not be covered on this year's midterm!
- NEW Midterm solutions for Fall 1997, in 2-up Postscript. Use *ghostview* to view them online, or *lpr* to print them.

Please note that sample final exams are not available; the instructor cannot supply any.

The Course Newsgroup

- The Course Newsgroup

The course newsgroup (`york.cs.course.1030`) is the first place you should look for course-related announcements and for answers to course-related questions. If you are registered in 1030 you should read the newsgroup regularly.

You may read the course newsgroup using your favorite newsreader. Alternatively, you may use your WWW browser to read the group by clicking on the hypertext reference above. Some WWW browsers, like mosaic, require you to set the environment variable NNTPSERVER before reading newsgroups. If you have trouble reading the course newsgroup using your WWW browser, try this

```
setenv NNTPSERVER blue.cs.yorku.ca
```

before starting your browser. Better yet, add this line to your .cshrc file.

The Course Directory

The course directory lives on the Ariel machines and is: `/cs/course/1030`. You may find the following subdirectories of interest.

- `notes` -- contains files and subdirectories with notes on different topics in the course.
- `text-code` -- all of the programs in the course text book are available
- `examples` -- all the examples used in lectures that went by too fast

---

Obtaining OOT Software

NOTE: the Computer Science Department and in particular the people in 126 CCB do not provide technical support for versions of Object Oriented Turing (OOT) that run under DOS, Windows, Windows '95 or the MacIntosh Operating System. If you have any problems with your disks, the installation of the software or with the actual software please contact Holt Software Associates Inc.

Holt Software Associates Inc.
203 College St., Suite 305,
Toronto, Ontario, CANADA, M5T 1P9
Phone: (416) 978-6476, 1 (800) 361-8324

Note that the "Computer Science Introductory Handbook" (September, 1995) contains information that should be helpful when installing this software on your machine.

If you purchased the OOT software last term you'll be able to use it for this course. Assignments will be done using OOT.

You can purchase Object Oriented Turing (OOT) at a special price of $35.00 (for disks only). You must know which version you will need **BEFORE** you purchase your software and after you have puchased your disks you should not bother the people who sold you the disks all problems should be directed to Holt Software Associates.

There are three versions of OOT available:

- MacIntosh Version
- DOS Version
- Windows Version (also works on Windows '95)

The software can be purchased in CCB 126 during the first two weeks of classes (starting Sept 8, through Sept 19) and during the following times only:

- 10 - 12 am
- 2 - 4 pm

Other Computer Science Courses