

**Computer Science 2SC3  
Software Engineering 2S03  
Principles of Programming  
McMaster University, Fall 2009  
Course Outline**

Revised: 13 October 2009

*Note: This course outline contains important information that may affect your grade. You should retain it and refer to it throughout the semester, as you will be assumed to be familiar with the rules specified in this document.*

**Instructor**

Dr. William M. Farmer  
Office: ITB 163  
Extension: 27039  
E-mail: [wmfarmer@mcmaster.ca](mailto:wmfarmer@mcmaster.ca)  
Web: <http://imps.mcmaster.ca/wmfarmer/>  
Office hours: M 9:30–10:30, T 16:30–17:30

**Teaching Assistants**

Pouya Larjani	graduate	<a href="mailto:larjanp@mcmaster.ca">larjanp@mcmaster.ca</a>
Han Yin (Chris) Zhang	graduate	<a href="mailto:zhanghy2@mcmaster.ca">zhanghy2@mcmaster.ca</a>
Rebecca Dreezer	undergraduate	<a href="mailto:dreezerj@mcmaster.ca">dreezerj@mcmaster.ca</a>
Kwadwo Amoako Sakyi	undergraduate	<a href="mailto:sakyika@mcmaster.ca">sakyika@mcmaster.ca</a>

**Schedule**

Lectures:	MW	08:30–09:20	MDCL 1105
	F	10:30–11:20	MDCL 1105
Tutorial 1:	M	13:30–14:20	BSB B103
Tutorial 2:	W	09:30–10:20	BSB B103
Tutorial 3:	R	08:30–09:20	JHE A102
Tutorial 4:	R	12:30–13:20	ABB 270
Tutorial 5:	T	10:30–11:20	ITB 139
Tutorial 6:	T	09:30–10:20	ITB 139

**Course Web Site**

<http://imps.mcmaster.ca/courses/SE-2S03-09/>

Some services for this course will be offered via E-LearnMac (ELM). Go to

<http://elm.mcmaster.ca/>

to access the course's ELM page.

Students should be aware that, when they access the electronic components of this course, private information such as first and last names, user names for the McMaster e-mail accounts, and program affiliation may become apparent to all other students in the same course. The available information is dependent on the technology used. Continuation in this course will be deemed consent to this disclosure. If you have any questions or concerns about such disclosure please discuss this with the course instructor.

*It is the student's responsibility to be aware of the information on the course's web site and ELM page and to check regularly for announcements.*

## **CS 2SC3 and SE 2S03 Calendar Description**

“Fundamental concepts of imperative programming (procedures, statements, control structures, iteration, recursion, exceptions); basic data structures (references, records, arrays, dynamic structures); basic concepts of operating systems.”

### **Mission**

Imperative programming is the oldest, and still one of the most popular, programming paradigms. The mission of the course is to teach students the underlying principles of imperative programming and basic data structures. By the end of the course the student should:

1. Understand the imperative programming paradigm and how it is different from other programming paradigms, particularly object-oriented programming and functional programming.
2. Be able to program in the imperative style in Objective Caml (OCaml), an advanced, high-level, multiparadigm programming language.
3. Be able to program in the imperative style in C, a traditional, intermediate-level, single-paradigm programming language.
4. Have a working knowledge of Unix-style operating systems, command-line interfaces, and version control systems such as Subversion.
5. Have a sense of what are the professional responsibilities of computing professionals such as computer scientists, software engineers, and mechatronics engineers.

### **Required Texts**

1. E. Chailloux, P. Manoury, and B. Pagano, *Developing Applications with Objective Caml*, O'Reilly France, 2000. English translation available for free at <http://caml.inria.fr/pub/docs/oreilly-book/>.

2. K. N. King, *C Programming: A Modern Approach*, Second Edition, Norton, 2008. ISBN 978-0-393-97950-3.

## Work Plan

The first two-thirds of the course will present imperative programming using the OCaml programming language. Although known for its functional programming capabilities, OCaml provides excellent support for imperative programming and produces exceptionally fast code. The availability of an interpreter for OCaml speeds up both learning and program development. The last one-third of the course will be a case study of imperative programming in the C programming language. C is the quintessential imperative programming language. It gives the programmer access to machine-level concepts and the capability to produce highly efficient and portable code.

There will be lectures, tutorials, programming assignments, quizzes, a midterm test, and a final exam. The lectures will be given by the instructor during regular class sessions. The tutorials will be conducted by the teaching assistants during the regular tutorial sessions. The programming assignments will be done by the students during their own time. There will be six assignments. The first four will be performed using OCaml and the last two using C. The assignments will be submitted using Subversion. Details concerning the assignments will be provided later.

There will be a short quiz given each Friday at the beginning of the lecture session using the i>clicker system. The first quiz will be given on September 18, 2009. There will be ten quizzes in total. The midterm test will be held on Friday, October 30, 2009 during the regular lecture time at 10:30–11:20. The final exam will be 3 hours long. It will take place on the date scheduled by the University.

The class will pick a *class representative* who will serve as a liaison between the students and the instructor.

## Log Book

Each student is required to keep a detailed, up-to-date log book that records all the steps performed on the programming assignments. Sources of information, consultations with instructors, teaching assistants, and fellow students; successful and failed experiments; discovered errors; and lessons learned should be recorded. The log book should be written as a text file, and the entries in the log book should be listed chronologically with dates and times. *A copy of the student's log book must be submitted with each programming assignment.*

## Clickers

A clicker system is for real-time, in-class feedback. This course will be using the i>clicker system, and all students taking this course are required to use i>clicker remotes. Students are expected to do the following:

1. Purchase an i>clicker remote. You only need one i>clicker for all your classes. You must have your own i>clicker; i>clickers may not be shared among students.
2. Register the i>clicker at

<http://www.iclicker.com/registration/>

prior to the second lecture. You must use your MacID where the “Student ID” is requested. Do not use your student number.

3. Bring the i>clicker to all lectures.
4. Maintain the i>clicker in working order throughout the course.
5. Attach a label to your i>clicker with your name so that you will not confuse your i>clicker with someone else’s.
6. Record the serial number of your i>clicker in case it is rubbed off.
7. Failure to follow the policies related to i>clickers may result in confiscation of the device(s).

*Students are required to attend and participate in the lectures. The i>clicker system will be used by the instructor to assess understanding of course work and measure participation. (See Marking Scheme below for how participation affects your final grade.) Using another student’s i>clicker or lending an i>clicker to someone to whom the i>clicker is not registered will be considered as academic dishonesty.* Attendance and participation in the tutorials is highly recommended but will not be recorded.

## **Academic Dishonesty**

You are expected to exhibit honesty and use ethical behavior in all aspects of the learning process. Academic credentials you earn are rooted in principles of honesty and academic integrity.

Academic dishonesty is to knowingly act or fail to act in a way that results or could result in unearned academic credit or advantage. This behavior can result in serious consequences, e.g., the grade of zero on an assignment, loss of credit with a notation on the transcript (notation reads: “Grade of F assigned for academic dishonesty”), and/or suspension or expulsion from the university.

It is your responsibility to understand what constitutes academic dishonesty. For information on the various types of academic dishonesty please refer to the Academic Integrity Policy, located at

<http://www.mcmaster.ca/academicintegrity/>

The following illustrates only three forms of academic dishonesty:

1. Plagiarism, e.g., the submission of work that is not one's own or for which other credit has been obtained.
2. Improper collaboration in group work.
3. Copying or using unauthorized aids in tests and examinations.

*Your work must be your own.* Plagiarism and copying will not be tolerated! If it is discovered that you plagiarized or copied, it will be considered as academic dishonesty.

Students may be asked to defend their written work orally.

## **Discrimination**

The Faculty of Engineering is concerned with ensuring an environment that is free of all adverse discrimination. If there is a problem, that cannot be resolved by discussion among the persons concerned, individuals are reminded that they should contact their Department Chair and the Human Rights and Equity Services (HRES) office as soon as possible.

## **Missed Work**

A student needs a Missed Work Form (MWF) for any work he or she misses. To get an MWF, the student must go to Engineering Services in JHE A214 with appropriate official documentation. MWFs will not be accepted for missed lectures.

## **Course Modifications**

The instructor and university reserve the right to modify elements of the course during the term. The university may change the dates and deadlines for any or all courses in extreme circumstances. If either type of modification becomes necessary, reasonable notice and communication with the students will be given with explanation and the opportunity to comment on changes. It is the responsibility of the student to check their McMaster e-mail and course web sites weekly during the term and to note any changes.

## **Other Policy Statements**

1. Each student is strongly urged to submit a completed Questionnaire and Biographical Sketch which is available on the course web site.
2. Significant study and reading outside of class is required.
3. The student is expected to ask questions during class.
4. You may want to discuss the assignments with your fellow students. *If you do that, you must record a summary of your discussions in your log book including a list of all those with whom you had discussions*

*and a description of what information you received.* It is part of your professional responsibility to give credit to all who have contributed to your work.

5. If there is a problem with the marking of an assignment, the student should first discuss the problem with the TA who marked the assignment. Assignment marks will only be changed if the problem was reported within two weeks of the date that the assignment was returned.
6. A student may use his or her texts and notes during the midterm test and final exam but not during the quizzes.
7. Programming assignments may not be submitted late and the quizzes and midterm test may not be taken later without *prior* approval from the instructor.
8. No electronic devices (calculators, cell phones, etc.) may be used on tests or brought into testing rooms. Hats are also not allowed in testing rooms.
9. The instructor reserves the right to require a deferred final exam to be oral.
10. Suggestions on how to improve the course and the instructor's teaching methods are always welcomed.

## Marking Scheme

The course grade will be based on the student's performance on the quizzes, programming assignments, midterm test, and final exam as follows:

Quizzes (10)	20%
Programming assignments (6)	20%
Midterm test	20%
Final exam	40%
<b>Total</b>	<b>100%</b>

Notes:

1. A student's final score will be reduced by one half point for each missed lecture. However, there is no penalty for the first *five* missed lectures for which participation is recorded. A student who answers less than half of the questions asked during a lecture via i>clickers will be considered as having "missed" the lecture. Missed lectures will be excused only in highly exceptional cases.

2. The instructor reserves the right to adjust the marks for a programming assignment, quiz, midterm test, or final exam by increasing or decreasing every score by a fixed number of points.

## **Syllabus**

- 00 Preliminaries
- 01 Programming Languages
- 02 Functional Programming
- 03 Imperative Programming
- 04 Basic Data Structures
- 05 Case Study: C Programming Language