ICS3U (Computer and Information Science)

Grade 11, University Preparation, 2010-2011

Instructor

Sam Scott (sam.scott@vufa.tdsb.on.ca)

Course Description

This is a first course in computer science. In this course, you will learn how to:

- design simple graphics, animations, and computer games;
- design programs that solve interesting and practical problems; and
- write programs that are readable, efficient, and re-usable.

Along the way, you will also learn something about:

- how programming languages work;
- what goes on inside a computer;
- current topics in computer science; and
- the impact of computing technology on society.

What is Computer Science?

Computer science is the study of mechanical computation. We will spend a lot of time learning about computers and how to program them, but the most important lessons of this course will be about **problem solving**, **logical reasoning**, and **analysis**. We will look at how to break down big problems into smaller problems, how to break down smaller problems into sequences of very small steps that can be performed by a machine, and how to program a computer to carry out these small steps. Working computer scientists bring their **theory** of computation and problem solving to their **practice** of programming computers to do useful things.

Programming Environment

We will be using **Java version 6**, the **Dr. Java** development environment, and the **hsa_ufa** package, which contains a lot of helper code for beginning programmers. This software is free and available for you to install at home for both PCs and Macs.

Readings

There is no text book for this course, but you should keep every handout and worksheet for later reference.

Portfolio

Students will be expected to keep a structured notebook containing all their handouts, paper exercises, and reflections on their learning. The portfolio will be graded for organization, completeness, and thoughtfulness. Students will be permitted to use this portfolio on all tests.

Content

Unit 1: Sequencing and Output

Text and graphics output, colors and fonts, flow charts, pausing the action, internal documentation standards, understanding programming languages.

Unit Project: An interactive greeting card

Unit 2: Input, Processing, and Output

The IPO Flow Pattern, binary representation, kinds of errors, algorithms, variables, data types, Strings, ints, doubles, input, arithmetic expressions, flow charts, tracing, testing, debugging, internal documentation standards.

Unit Project: Automated computer store software.

Unit 3: Selection Statements

Branch & Boundary testing, random numbers, nested if statements, elseif chains, switch statement, the char type, Boolean logic, flow charts, tracing, testing, and internal documentation standards. Flow patterns: DoOrDie, EitherOr, OneOfMany, RandomChoice, DecisionTree.

Unit Project: A "choose your own adventure" game.

Unit 4: Modularity (Mini-Unit… no test or major assignment)

Modularity in design, methods, variable scope, parameters, return values, tracing, the Boolean type.

Unit Project: N.A.

Unit 5: Repetition Statements

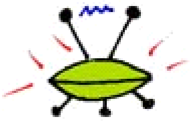
While loops, for loops, do..while loops, flow charts, tracing, testing. Flow patterns: ITACL, CountOff, SuddenDeath, Greedy, IACTL.

Unit 6: Indexed Data Structures

Arrays, array processing, array searching, related arrays. Patterns: CountOff, SuddenDeath, Greedy. May also include a section on computer architecture and performance.

Unit Project: Student grades analysis program.

Unit 7: Final Project



Possible projects: A game, a database or inventory system, a project that integrates with another course, or any other project that involves concepts from Units 1 to 4.

To hand in: A project plan, a design document, a test document, and a working, user-friendly program with well-documented code.

Note: Students may use this project to expand on what they have already learned, for example by learning about String processing or two-dimensional arrays.

Assessment and Evaluation

Unit tests and Quizzes:	20%	Final project:	30%
Unit projects & Exercises:	38%	Wednesday:	7%
Portfolio:	5%		

Advice for Students

You learn computer science by doing it. To do well in this course, you need to be actively engaged. This means taking notes, using class time to complete exercises and assignments, working on exercises and assignments outside of class time when necessary, and actively seeking help when necessary from the teacher, other students, on-line resources, textbooks, and whatever other resources are available to you.

Working Together and Helping Each Other

Students are encouraged to work together and help each other with assignments. When one student helps another in a productive way, they both end up understanding it better. But some ways of helping are more productive than others, and some ways of "helping" are no help at all.

Good ways to help…

- Talking things over with someone to help them understand a concept
- Helping someone find the information they need
- Testing another student's program to look for mistakes
- Sitting with someone to advise them while they write or debug a program something they are having trouble with

Bad ways to help…

- Writing a part of somebody's code for them.
- Mailing somebody your program so they can use it as a template, cut and paste parts of it, or change it slightly and hand it in as their own.

Plagiarism

In the worst case, both of the "bad" cases above could lead to plagiarism (somebody taking credit for somebody else's work). Plagiarism in a computer program is as easy to spot as if two students handed in the same essay, even if the program has been changed a bit to make it look different. Plagiarizing in this way is dishonest and does not help anybody.

When plagiarism happens, all students involved are equally guilty, whether the plagiarism was intentional or not. The consequences range from a zero on the assignment to suspension from school. Protect yourself: don't write somebody else's program for them, and never anybody your own program to look at. It's too risky.

TDSB Plagiarism Policy: http://schools.tdsb.on.ca/whci/policies/academicdishonesty.html