

# Syllabus

AS/AK/ITEC 1620 3.0  
Object-Based Programming  
Section C, Fall 2002

Instructor:	Prof. S. Chen
Office Hours:	Thursdays 4-6pm, CCB152
TA's:	TBA
Classroom:	Computer Science Building, Theatre A
Time:	Monday 7:00 – 10:00 pm
Lab:	ITEC common image
Final:	TBA (Note: may not be on Monday night)
Textbooks:	Java Software Solutions: Foundations of Program Design, Lewis & Loftus – 3 <sup>rd</sup> Ed.  Computing Concepts with JAVA 2 Essentials, Horstmann
Homepage:	<a href="http://www.atkinson.yorku.ca/~sychen">http://www.atkinson.yorku.ca/~sychen</a>

## **Course Description**

This is an introductory course in computer programming and problem solving. This is not a course on JAVA programming. You will be expected to learn JAVA to demonstrate that you understand the primary concepts. Writing computer programs to solve real-world problems involves many unique tools – using precise logic, developing complete specifications, communicating intent, and paying excruciating attention to details. If you learn these tools well, learning JAVA programming will be relatively easy.

## **Computer Labs**

The required computing platforms and programming environments are available across campus in GLADE, CLAS, PS-2, GAUSS, and McLaughlin. When available, a link to the lab homepage will be put on the course homepage. Please monitor this future link to learn more about lab and TA availability.

## **Textbook**

There are two textbooks for this course: Java Software Solutions: Foundations of Program Design, 3<sup>rd</sup> Ed. by Lewis and Loftus (required), and Computing Concepts with JAVA 2 Essentials by Horstmann (recommended). I find that the first text is better for introductory concepts, and the second text is better for advanced details. However, it should be possible to handle the course using either text. Suggested readings from both texts are listed at the end of this syllabus. However, suggested problems are only given for Lewis and Loftus.

## **Class Format**

I will divide each lecture into two halves. Each half will be about one hour and 15 minutes each, and they will be separated by a 10-15 minute break. During this break, I will not answer any individual questions. Please ask relevant questions during class, or individual questions at the end of class.

## **Important Dates**

There will be no class on September 16 for Yom Kippur and October 14 for Thanksgiving. I will have a make-up lecture on Saturday September 14 from 3-6 pm in CSB A. Additional tutorials have been scheduled on Saturdays from 3-6 pm in CSB A for September 28, October 26, and November 23. You may also attend the scheduled tutorials for the other sections.

## **Questions and Other Lecture Interruptions**

Learning is an interactive process. You will learn more about programming and I will learn more about teaching if you participate actively during lectures. If something is unclear or confusing, please request a clarification—it will likely be of benefit to the entire class. However, I may delay extended questioning until the review/tutorial sessions in order to maintain the course schedule.

## Evaluation

Program 1 (due 9/23):	2%
Program 2 (due 10/21):	3%
Program 3 (due 11/4):	5%
Program 4 (due 11/18):	5%
Program 5 (due 12/2):	5%
Midterm (on 10/28):	30%
Final (TBA):	50%

For grade conversion, please see

<http://calendars.registrar.yorku.ca/calendars/2001-2002/ugfiles/acad/3.htm>

## Late Policy

Late assignments will NOT be accepted unless medical or other acceptable documentation is supplied. Such documentation must indicate that:

- some problem prevented you from working on the task for at least 50% of the time that you had to do it.

When the medical documentation indicates that you were in the hospital for the period of time corresponding to that assignment, the weight of that assignment will be added to the weight of the final exam. A make-up midterm will NOT be provided. If you miss the midterm for medical reasons the weight will be added to the weight of the final exam.

## Academic Honesty

Computer programs are the equivalent of essays. While you are encouraged to work with classmates as part of your educational experience, you must submit **independent** work – nothing that has been copied electronically. The penalty for submitting a copied program is zero for that program and an additional penalty of 5% deducted from your final grade.

## Quotes (useful comments about programming)

It goes against the grain of modern education to teach students to program. What fun is there to making plans, acquiring discipline, organizing thoughts, devoting attention to detail, and learning to be self critical.

- A. Perlis

In practice, debugging often takes the place of understanding how programs work (i.e. if we all understood perfectly how our own code worked, we would not need to debug it to find out why it is not doing what we think it should).

- Literate Programming page

<http://www.cs.cmu.edu/~vaschelp/Programming/Literate/literate.html>

## Lecture Topics

<u>Lect</u>	<u>Day</u>	<u>Topic</u>
1a	9/9	Introduction
1b	9/9	Primary Concepts: Structured and Object-Oriented Programming
2a	9/14	More object-oriented programming and an introduction to JAVA
2b	9/14	Mathematical expressions and primitive data types
3a	9/23	Problem solving tools 1: Flowcharts and Pseudocode
3b	9/23	More algorithm development
4a	9/30	Boolean algebra and Relational operators
4b	9/30	JAVA control structures 1: the <code>if</code> statement
5a	10/7	JAVA control structures 2: the <code>while</code> and <code>for</code> statements
5b	10/7	JAVA control structures 3: Advanced topics and Spaghetti code Discussion of Program 2 and Program 3
6a	10/21	Problem solving tools 2: Object-Oriented Analysis and Design
6b	10/21	Review (sample midterm)
7a	10/28	MIDTERM
7b	10/28	Classes and Objects, Access modifiers
8a	11/4	Understanding Classes 1: Methods, Constructors, Object diagrams, References
8b	11/4	Understanding Classes 2: API's, Over-loaded methods, Cascaded constructors Discussion of Program 4
9a	11/11	Using References 1: creating, copying, equals, and equality
9b	11/11	Using References 2: scope, lifetime, pass-by-copy, pass-by-reference, <code>this</code>
10a	11/18	Using References 3: review
10b	11/18	Arrays
11a	11/25	Strings
11b	11/25	Using existing classes (e.g. Vectors), Discussion of Program 5
12a	12/2	Simple Array Algorithms
12b	12/2	Review

## Suggested Readings from Lewis & Loftus

<u>Lect</u>	<u>Topics</u>	<u>Sections</u>
1a	material covered in ITEC1011	1.0-1.2, 1.5
1b	intro to programming	1.3 (27-28), 1.4
2a	intro to JAVA	1.3 (29-38)
2b	variables, assignment, primitive datatypes, expressions	2.3-2.5, 3.5 (153-156)
3a	program design	3.0, 3.9
3b		
4a	Boolean expressions	3.2 (136-137), 3.4
4b	if statements	3.2
5a	while and for statements	3.6, 3.8
5b	do statement	3.7
6a	software engineering and object-oriented design	10.0-10.3
6b		
7a		
7b	classes and objects access modifiers	2.0, 2.1, 2.7, 4.0 4.1 (220-223)
8a	methods constructors static fields and methods	2.8, 4.1, 4.2 2.6 (87-88), 4.2 (232-233) 5.1
8b	API's and overloading	4.3
9a	creating and copying references, equals and equality	5.0 (270-275)
9b	scope pass-by-copy, this	4.2 (233) 5.0 (276-281)
10a		
10b	arrays, arrays of objects	6.0, 6.1
11a	Strings – methods and peculiarities	2.2,2.6 (88-91), 5.1 (272-275)
11b		
12a		
12b		

## Suggested Readings from Horstmann

Note: The Horstmann text has lots of extra stuff, so page omissions are too numerous to list.

<u>Lect</u>	<u>Topics</u>	<u>Sections</u>
1a	material covered in ITEC1011	1.1-1.3
1b	intro to programming	1.4-1.5, 1.10
2a	intro to JAVA	1.6-1.9
2b	variables, assignment, primitive datatypes, expressions	2.1-2.5
3a	program design	1.12
3b		
4a	Boolean expressions	5.2.1, 5.4
4b	if statements	5.1, 5.3
5a	while and for statements	6.1, 6.2
5b	do statement, the “loop and a half” problem	6.3, 6.5
6a	software engineering and object-oriented design	14.1-14.3
6b		
7a		
7b	classes and objects access modifiers	1.11, 3.1, 3.3 3.2
8a	methods constructors static fields and methods	3.2, 3.4, 7.1, 7.2 3.5 7.3, 7.5
8b	API’s and overloading cascaded constructors	3.5 (117) 3.5 (117-118)
9a	creating and copying references, equals and equality	3.8, 3.9, 5.2.4
9b	scope references as parameters, this	7.6 7.3, 3.3 (116-117)
10a		
10b	arrays, arrays of objects	11.1, 11.2, 11.4, 11.5
11a	Strings – methods and peculiarities	2.6, 5.2.3
11b	vectors	11.6
12a	simple array algorithms	11.3
12b		

## Program Marking Guide

A program that does not run / solve the base case cannot receive more than 50%. However, a working program may still not receive a perfect score. Marks will be deducted for poor style and/or incorrect logic. Examples are provided below:

- using `break`
- using `continue`
- bad variable names
- using a formal parameter as a local variable
- redundant code
- poor indentation/formatting
- poor use of structures
  - using `for` instead of `while`
  - using `while` instead of `for`
  - using `else if` instead of `else`
- using over-optimized/unreadable code
  - ternary operator
  - non-obvious use of auto increment/decrement
- anything that would cause an error
  - wrong loop conditions
  - missing cases