Concordia University
Computer Science & Software Engineering
Faculty of Engineering and Computer Science

Concordia University
COMP 248 – Object Oriented Programming I
Winter 2020
Course Outline

Section	Instructor	E-mail	Office	Office Hours
U	N. Acemian	nancy.acemian@concordia.ca	EV3.153	***
W	Y.Yan	yuhong.yan@concordia.ca	EV3.237	***

*** Please refer to your section's course web page for your instructor's office hours.

Background Knowledge

You should have some experience with using a computer. It is helpful, but not necessary, to have some experience in programming with a language such as C or C++. The tutors will provide a gentle introduction in the first one or two tutorials for those who are not familiar with the programming environment to be used.

Required Text Book & materials

- *Big Java Late Objects* by Cay Horstmann, Wiley. The same book will be used for COMP 249. We recommend the digital version as this will give you access to programming exercises you can do online.

- *i-Clicker* which you need to bring to every class. Available at the bookstore.

Course Objectives and Content

The primary objective of this course is to enable you to understand the basic principles of programming. The language used for the course is Java, chosen because it supports object oriented programming and because it is becoming widely used in industry.

By the end of this course you will be able to:

CO 1. Develop an algorithm for the solution of a simple problem.
CO 2. Write a complete Java program given an algorithm.
CO 3. Describe the output of a Java program written by someone else.
CO 4. Start judging the efficiency of a program.
CO 5. When writing a program,
a. evaluate which type of statements best suits the problem you are solving.
b. correctly implement the various types of statements.
c. manipulate existing classes.
d. design and implement a class.

The course will include discussions and explanations of the following topics: introduction to programming; writing, compiling, and running simple programs; expressions, variables, and assignments; control structures; objects and classes, methods, and arrays.

DRAFT

DRAFT

CEAB Attribute Assessment

The Canadian Engineering Accreditation Board (CEAB) has put forth specific requirements for the accreditation of engineering programs in Canada. To satisfy these requirements, the faculty is evaluating its courses against specific attributes required for the accreditation. The attribute evaluated in COMP 248 are:

- **Knowledge-base:** Knowledge of basic principles of programming, especially object-oriented programming, including basic data types, variables, expressions, assignments, control flow, classes, objects, methods, information hiding, data abstraction and encapsulation, references and arrays.
- **Design – Implementation and validation:** Design and implement simple programs using an object-oriented programming language.
- **Use Appropriate Resources/Tools:** Determine what appropriate language constructs to use to solve specific problems.
- **Communication Skills:** Internal code documentation.

Computing Facilities

You should obtain a computer account for COMP 248 from the help desk at H–964 or EV-007.182. This account will give you access to the computer laboratories. For more information on CSE Computer accounts please visit the website: http://www.encs.concordia.ca/helpdesk/access.html
If you have a computer at home and prefer to use it, you may do so, but be aware that your programs must compile and run with Java 8.

Course Web Page

Many resources for the course (lecture slides, assignments and solutions, example programs…) will be available on the course website. Individual instructors will give you details on where the course webpages for their section is hosted. (If on Moodle you can access it through the MyConcordia portal at www.myconcordia.ca)

You also have access to the ***COMP248_Plus*** Moodle page, which is a learning environment containing portions of videos of past lectures by N. Acemian. The videos are organized by topics. Many of the topics have an online quiz which you can complete to evaluate your understanding of the topic. It is meant as a supplement to the in-class experience and not a replacement for it.

Tutorials

The tutorials will reinforce the material seen during the lectures with examples and exercises. Assignments as well as solutions to lab problems will be explained during the tutorials. Each week, there will be a set of tutorial exercises posted in the Moodle course page of which a selection will be covered during the tutorial period. The rest will be left to you as additional practice.

To benefit from the tutorial sessions, it is recommended that you attempt a few of the posted questions prior to your tutorial session. This will allow you to identify which concepts you are having difficulties with and ask your TA questions related to these concepts.

Labs

The purpose of the labs is to help you be better programmers and to develop the skills required to *program to specifications a skill you will need when going for interviews for most programming jobs*. You will be asked to write code that produces very specific outputs and/or behaves in very specific ways, unlike the assignments where you have more leeway on the output.

There will be 7 lab exercises to be completed (Lab 1 to 7). These exercises can only be done during the specified lab period you are registered for and must be submitted at the end of your lab period. The best 6 of 7 grades will be considered. The topic of each lab will be posted on your course web page. You must attend the lab session you are registered for as you will not be able to login to the lab system outside of your assigned lab section's time. There is also a Lab 0 which does not count to allow students to get used to the PC^2 lab environment.

Please note: No make-up labs available.

Lab switch policy: If you need to switch lab for a week (not for the term) you need to send an email to your Head Lab Demonstrator (name/email available on your course webpage) at least 48 hours before your scheduled lab time. There is quite a bit of work involved in giving you access to a new time and removing the access to your regular lab time. This cannot be done at the last minute.

Assignments

The purpose of the assignments is to *experiment* with the concepts covered throughout the course. Assignments are different every term and are designed to reflect the concepts covered in class. Do not waste time searching for a solution online. There will be 4 programming assignments. All assignment questions must be downloaded from your section's course webpage and submitted electronically. Instructions on submitting assignments will be available on your section's course web page. For due dates refer to the *tentative* schedule at the end of this handout.

Please note: No late assignments are accepted. No make-up assignments are available.

It is recommended that you write a general algorithm for your solutions before you start typing the Java code. Programming by *trial and error* is not the way to succeed in this course. Be sure to understand why a certain version of your program did not work and why another one did. Questions to ask yourself each time you finish a question and/or assignment to evaluate your learning: "If I were to redo this assignment would it take me less time? Would I need to refer to my notes as much? Did I learn from this assignment?" If the answer is *Yes*, bravo! If the answer is *No*, seek help right away from your teacher and/or TA to better understand the issues you are still unclear on.

Examinations

- **Midterm Exam:** There will be one in-class midterm exam which will take place during regular lecture times. The term test will occur in week 7 of the term (Feb 21, Friday). List of exact term test dates for each section can be found at the end of the tentative schedule on page 5 of this document. There are no make-up midterm exams.

DRAFT

- **Final Exam:** The final examination will last three hours, and will be administered during the examination period at the end of the term. The final examination covers all material seen during the term.

Note: The final exam is scheduled by the Exams Office. Do not make any travel plans until the official Exam Schedule is announced.

Evaluation Scheme

4 Assignments	18% (2% + 3% + 5% + 8%)
7 Lab Exercises	12% (2% each – Best 6 out 7)
1 in-class theoretical tests	25%
1 Final Exam	45%

OR

4 Assignments	18% (2% + 3% + 5% + 8%)
7 Lab Exercises	12% (2% each – Best 6 out 7)
1 Final Exam	70%

1. The best of both options above will be awarded to each student.

2. In order to pass the course, you **must** complete all components of the course (assignments + labs + midterm and final).

3. ENCS Academic Regulation – The C- Rule: *All 200-level courses within the program which are prerequisites for other courses must be completed with a C- or higher. A 200-level course in which a student has obtained a D+ or lower must be repeated before attempting a course for which it is a prerequisite.*

4. There is **no standard relationship** between percentages and letter grades assigned.

5. Although we encourage discussion of the assignment questions among students, you should be aware of the University regulations concerning plagiarism described in the undergraduate Calendar. All students should become familiar with the University's Code of Conduct located at http://www.concordia.ca/students/academic-integrity/code.html In cases where cheating or plagiarism is suspected, the case will be forwarded directly to the appropriate university office.

How to study in this course

- As the material is cumulative, keep up with the content covered in the weekly lectures.

- If certain concepts are unclear to you, seek help right away. Ask your TA during the tutorial and/or your instructor for help. **Make use** of your instructor's office hours.

- Programming is not a "spectator sport". You need to get your hands dirty by doing the assignments and trying the examples discussed in class and in the posted tutorial exercises.

DRAFT

- Please refer to the STUDY TIPS under *Assignments* and *Tutorials* sections.

- Take advantage of the exercises in your textbook available through your e-textbook

Tentative **Schedule**

The table below provides a summary of the material that will be covered during the course as well as a ***tentative*** schedule. Please check your section's course web page for any changes.

Week	Dates	Chapter	Topic	Lab Exercise	Assignment
1	Jan 6 - Jan 12	1.1, 1.3 to 1.7 & 2	Introduction & Java Basics	☒	
2	Jan 13 - Jan 19	2	Java Basics	0	A1 available
3	Jan 20 - Jan 26	3	Flow of Control (Selection)	0	
4	Jan 27 - Feb 2	4	Flow of Control (Iteration)	1	A1 due * A2 available
5	Feb 3 – Feb 9	3	Flow of Control (Iteration)	2	
6	Feb 10 – Feb 16	6.1, 6.3, 6.5 to 6.7	Arrays of primitive types	3	A2 due * A3 available
7	Feb 17 – Feb 23	6.1, 6.3, 6.5 to 6.7	Arrays of primitive types **Midterm Exam****	☒	
	Feb 24 – Mar 1	Spring Break/Reading week			
8	Mar 2 – Mar 8	5	Defining Classes	4	
9	Mar 9 – Mar 15	8.1 to 8.7	Defining Classes	5	A3 due * A4 available
10	Mar 16 – Mar 22	----	Arrays of objects	☒	
11	Mar 23 – Mar 29	8.8, 8.10, 8.11.1, 8.11.2, 8.11.4 to 8.11.6	More on Classes	6	
12	Mar 30 – Apr 5	9.5.1, 9.5.2	More on Classes	7	
13	Apr 6 – Apr 9		Catch Up and/or Review	☒	A4 due *

***** **All assignments are due by 11:55pm on the Friday of the due week.**

**** **Midterm Exam Dates**

- **Section U** (Acemian): Friday February 21, 2020
- **Section W** (Yan): Friday February 21, 2020

Note

In the event of extraordinary circumstances beyond the University's control, the content and/or evaluation scheme in this course is subject to change.

DRAFT