Faculty of Arts and Sciences
Department of Computer Science
CMPS 200 – Introduction to Programming
Course Syllabus – Spring 2022

Instructor

Loa Aoude
la52@aub.edu.lb
Office: Bliss 115
Office hours: Wednesday 12:00 – 3:00

Lectures: MWF 9:00 - 9:50 Room: Bliss 203
Labs:

Lab Section	Day/Time	Room	Instructor	Email
B1	M 2:00 – 5:00	Bliss 011	Maher Jaber	mkj02@mail.aub.edu
B2	M 2:00 – 5:00	Bliss 133	Sara Shoujaa	shs33@mail.aub.edu
B3	R 2:00 – 5:00	Bliss 134	Al-Abbass Khalil	ak202@mail.aub.edu
B4	R 2:00 – 5:00	Bliss 133	Muhamad Sonji	mms158@mail.aub.edu
B5	R 2:00 – 5:00	Bliss 011	Ademola Adesokan	aaa277@mail.aub.edu

Description

CMPS 200 introduces programming and problem solving. The course helps establish a rigorous foundation in programming techniques and introduces the student to software development methodology.

The course assumes no prior programming knowledge. Nevertheless, the course is a demanding one as (1) it moves at a brisk pace and expects students to keep up with material covered in the lectures on a daily basis and (2) the course requires a substantial amount of work in the form of programming assignments.

We aim to bring the excitement of computer science to you through this course. Doing so requires hard work and perseverance on your part, but for which you will amply rewarded by a fundamental understanding of the one most important transformative technologies of recent decades. We go over the key concepts multiple times and emphasize repeated practice. Weekly programming assignments are a key component of the course. For most of you who have not coded before, it is easy to get discouraged sometimes, but don't! The instructors and Teaching Assistants (Tas) are here to help you. The general expectation is that you need to spend no more than 10 hours per week on average to get the work done. If you are in trouble, please let us know.

Learning outcomes

The goal of this course is to give a student the foundation needed to develop programs that are correct, efficient, well structured, and robust, and that adhere to standards of style and documentation. The student will then build on these skills in following courses in his/her course of study. A student will learn how to solve problems by breaking them down, analyzing them, designing solutions, implementing the

designed solutions, and verifying or testing the resulting implementations. A student will also gain good programming skills by studying examples and reinforcing discussions.

Upon successful completion of this course, a student should be able to:

1. Construct working programs that adhere to standards of style and documentation using primitive data types, conditionals, iterative control structures, and arrays.
2. Write functions/ methods that implement given signatures.
3. Decompose a problem into a well-structured set of functional abstractions.
4. Develop strategies for incremental development and debugging of programs.
5. Use classes, including classes from standard libraries, to organize the data that programs operate on and to manage program complexity.
6. Design classes and programs from either informal problem specifications or well-defined specifications of their interface.

Resources

- S. Reges and M. Stepp, "Building Java Programs A Back to Basics Approach 5th Ed", Pearson, 2019.
- John Lewis and William Loftus, JAVA Software Solutions: Foundations of Program Design, 7th Edition, Pearson, 2012.
- Oracle JAVA Development Kit: Java SE Development Kit 16 - Downloads (oracle.com)
- JetBrains IntelliJ IDEA Community Edition development environment. You may download it from https://www.jetbrains.com/

Grading

A tentative breakdown of the final grade for the course is as follows (tentative percentages):

- 10% Labs and participation
- 10% Assignments
- 20% Exam 1
- 20% Exam 2
- 40% Final Exam (date to be scheduled)

Course Policies

Attendance:

You are expected to attend lectures and labs and to be on time. Lectures are a sequence. Missing a lecture may mean that you may not be able to keep up with the subsequent lecture(s) without studying the material covered. Attendance during lab sessions and lectures may be taken.

Lab Sessions

The class lectures are complemented by one 3-hour weekly lab session. Labs are hands on sessions intended to assist you in using the programming environment for creating and debugging your programs and completing the weekly programming assignments. Attending the lab sessions is mandatory for your lab work to be accepted and graded. No attendance = no lab grade.

Assignments:

Assignments will be given weekly once the course advances enough. These assignments will be given a rough deadline of around a week. The deadline will be set for each assignment. All assignments should be submitted to Moodle on or before the due date. No late assignments will be accepted. Discussions with others are encouraged but the submitted work should be your own.

Finger exercises:

Finger exercises are small, graded exercises that may be given in the form of a drop quiz for 5 or 10 minutes at the beginning of a lecture or lab session and will be solved through Moodle's CodeRunner and, hence, will be automatically graded.

Exams:

Makeup exams will only be given to students who present a documented valid excuse within one week from the exam date. Students who fail to do so will get a zero on the missed exam. The University policy regarding incomplete work will be followed. Check university catalogue for details.

University policies

Academic Integrity

Please refer to the AUB Student Code of Conduct, in particular section 1.1, which concerns academic misconduct including cheating, plagiarism, in-class disruption, and dishonesty. Please be aware that misconduct is vigorously prosecuted and that AUB has a zero-tolerance policy. Course policy is that credible evidence of cheating will result in course failure.

Special Needs

AUB strives to make learning experiences as accessible as possible. If you anticipate or experience academic barriers due to a disability (including mental health, chronic or temporary medical conditions), please inform me immediately so that we can privately discuss options. To help establish reasonable accommodations and facilitate a smooth accommodations process, you are encouraged to contact the Accessible Education Office: ***accessibility@aub.edu.lb; +961-1-350000, x3246; West Hall, 314.***

Non-Discrimination

In line with its commitment to the principle of equal opportunity in education and employment, AUB policies protect you from discrimination on the basis of protected characteristics, including discriminatory harassment and sexual harassment. Protected characteristics include race, color, religion, age, national or ethnic identity, sex, gender or gender identity, sexual orientation, pregnancy, marital status, disability, genetic predisposition or carrier status, alienage or citizenship status, and political affiliation.

The policies are applicable to all the AUB Community. If you think you have experienced discrimination, discriminatory harassment, or sexual harassment, we encourage you to inform the Equity/Title IX Coordinator, Mitra Tauk at 01-350000 ext. 2514, **titleix@aub.edu.lb**, report to a Title IX deputy at your faculty or at any other faculty (**www.aub.edu.lb/titleix**), or report online (**www.aub.ethicspoint.com**). Reports may be submitted anonymously or not. Please know that the University will maintain the confidentiality of the complaint and privacy of the persons involved to the greatest extent possible, consistent with its goal of conducting a thorough and complete investigation and to the extent

permitted by law. For the full Title IX syllabus statement, please visit https://www.aub.edu.lb/President/TitleIX/Pages/syllabus.aspx.

Tentative Outline

Week	Date	Material	Chapters
Week 1	24/01-28/01	- Introduction: The Programming Process - println Statements	Chapter 1
Week 2	31/01-04/02	- Primitive data types (Variables and Expressions) - Using Objects (Math, String)	Chapter 2 Chapter 3
Week 3 (2 lectures)	07/02-11/02	- Interactive Programs, Scanner - Conditional Statements	Chapter 3 Chapter 4
Week 4	14/02-18/02	- Functions - Program Organization	Chapter 3
Week 5	21/02-25/02	- Loops	Chapter 2
Week 6	28/02-04/03	- Indefinite Loops	Chapter 5
Quiz 1 – Saturday March 5			
Week 7	07/03-11/03	- Arrays	Chapter 7
Week 8	14/03-18/03	- Collections: ArrayList	Chapter 10
Week 9 (2 lectures)	21/03-23/03	- Recursion	Chapter 12
Week 10	28/03-01/04	- Solving	
Quiz 2 – Saturday April 2			
Week 11	04/04-08/04	- Exceptions - File Processing	Chapter 6
Week 12 (2 lectures)	11/04-13/04	- Data Abstraction and Classes	Chapter 8

Week 13 (1 lecture)	20/04	- Solving	
Week 14 (2 lectures)	27/04-29/04	- Solving	