
INTRO TO COMPUTER PROGRAMMING I in Fall 2018 (CS1040)

Course Code	CS1040	Professor(s)	Claudia Roda
Prerequisites	None	Office Number	SD
Class Schedule	F: 13:45-16:40 in PL-2 T: 15:20-16:40 in PL-2	Office Hours	by appointment
Credits	5	Email	croda@aup.edu
Semester	Fall 2018	Office Tel. Ext.	

Course Description

Introduces the field of Computer Science and the fundamental concepts of programming from an object-oriented perspective using the programming language Java. Starts with practical problem solving and leads to the study and analysis of simple algorithms, data types, control structures, and use of simple data structures (arrays and strings). The course emphasizes good software engineering principles and developing fundamental programming skills. The pedagogical methodology emphasizes experiential learning by asking students to solve real problems from the start. Analyzing and representing problems, as well as finding the information necessary to arrive to their solutions are the recurring themes of this course. There will be a considerable amount of practical work; students should allow at least eight hours of home work per week for this. All classes will take place in the Computer Laboratory so that it will be possible to experiment and build-on all the theories and algorithms introduced.

Course Learning Outcomes

Students will demonstrate knowledge of the theory, practices, and tools for the specification, design, implementation, and evaluation of digital systems. Measurable outcomes include the ability to read a program, understand the concept of syntax and semantics of a programming language, understand simple object oriented design (class + client structure)

Students will be able to apply the above knowledge to solve practical problems demonstrating understanding of simple trade-offs involved in system implementation. Measurable outcomes include the ability to use control structures (conditions and loops) and structured data types (e.g. newly defined classes, string, random) to build clear and efficient solutions to simple problems.

Students will be able to communicate efficiently and with the appropriate vocabulary about simple technical problems and their solutions.

General Education

The general education program at AUP consists of four requirements: Speaking the World, Modeling the World, Mapping the World, and Comparing Worlds Past and Present.

This course can be used to fulfill the Mapping the World requirement and as such has the following learning outcomes. At the end of this course:

- Students will have a clear understanding of algorithmic reasoning
- Students will have practiced algorithmic reasoning
- Students will understand how programming languages relate to natural languages
- Students will be able to generate formal descriptions of worlds properties and interactions

Course Outline

A detailed description of the course is available on the course Web site:

<http://ac.aup.fr/roda/cs140>. Note that content schedule, assignments, readings, etc. are updated online during the semester therefore you should make sure you check the course site regularly.

Textbooks

Title	Author	Publisher	ISBN	Required
JAVA SOFTWARE SOLUTIONS: FOUNDATIONS OF PROGRAM DESIGN., 9TH NEW ED.	LEWIS, JOHN ET AL.	ADDISON WESLEY	9781292221724	Yes

Attendance Policy

Students studying at The American University of Paris are expected to attend ALL scheduled classes, and in case of absence, should contact their professors to explain the situation. It is the student's responsibility to be aware of any specific attendance policy that a faculty member might have set in the course syllabus. The French Department, for example, has its own attendance policy, and students are responsible for compliance. Academic Affairs will excuse an absence for students' participation in study trips related to their courses.

Attendance at all exams is mandatory.

IN ALL CASES OF MISSED COURSE MEETINGS, THE RESPONSIBILITY FOR COMMUNICATION WITH THE PROFESSOR, AND FOR ARRANGING TO MAKE UP MISSED WORK, RESTS SOLELY WITH THE STUDENT.

Whether an absence is excused or not is ALWAYS up to the discretion of the professor or the department. Unexcused absences can result in a low or failing participation grade. In the case of excessive absences, it is up to the professor or the department to decide if the student will receive an "F" for the course. An instructor may recommend that a student withdraw, if absences have made it impossible to continue in the course at a satisfactory level.

Students must be mindful of this policy when making their travel arrangements, and especially during the Drop/Add and Exam Periods.

Grading Policy

Class test	45%
Class participation	10%
Final Examination	45%

Typically a grade of A will be above 90% and F below 50%.

Other

DETAILED OUTCOMES:

A more detailed list of measurable outcomes for this course include the following:

- Identify possible design solutions to simple problems (typically including 2 to 4 students' defined classes, each containing several methods and variables) and describe these solutions in English (e.g. "I will need to define two classes. The first class will contain three instance variables, and one class variable, ...")
- Implement Java programs for design solutions that students have created or have been given.
- Understand and use data abstraction, encapsulation, and message passing. Note that the concepts of polymorphism and interfaces will not be formally introduced in this course that only covers the basics of inheritance.
- Use comments efficiently. Appreciate the importance of appropriate naming.
- Understand the basics of using APIs
- Read and understand programs that may make use of libraries that have not been studied before (within a reasonable choice)
- Understand the use of Java syntax and semantics and being able to find out missing information (e.g. using the online Java API)
- Fully understand the differences between primitive types and reference types (this includes concepts such as shallow copying and practical issues such as parameter passing in the two cases).
- Use variables and constants, understand their scope, visibility, and accessibility.
- Use basic Java operators. Understand precedence rules, arity, and general composition of Java expressions.
- Create and use instance and class methods, passing parameters, and returning values. The message passing metaphor, methods' signatures, and overloading. Understand the role and be able to build different type of methods such as constructors, getters, setters, and modifiers. Understand the use of these constructs.
- Understand control-flow and use selection and repetition constructs.
- Simple I/O.
- Demonstrate understanding of the role and issues related to the creation of testing units. As well as being able to create meaningful testing units for small programs.
- Develop a good experience with at least one IDE for Java.
- Use simple collections such as Strings, and if time allows Arrays, ArrayList, and Vector. Understand their different properties and being able to select the appropriate structure for a given problem.

