



Course Syllabus: Programming Methodology and Abstractions - CS 207

Division	Computer, Electrical and Mathematical Sciences & Engineering
Course Number	CS 207
Course Title	Programming Methodology and Abstractions
Academic Semester	Summer
Academic Year	2018/2019
Semester Start Date	06/16/2019
Semester End Date	08/08/2019
Class Schedule (Days & Time)	09:00 AM - 12:00 PM Mon Wed

Instructor(s)				
Name	Email	Phone	Office Location	Office Hours
Malek Smaoui	Malek.Smaoui@KAUST.EDU.SA	+966128080331	4304, 1, Al-Khwarizmi (bldg. 1)	Office: bldg. 1, room 4304 Hours: by appointment. Please request appointments via email at least 12 hours in advance.

Teaching Assistant(s)	
Name	Email
Amani Ageeli	amani.ageeli@kaust.edu.sa

Course Information	
Comprehensive Course Description	This course starts with a familiarization to algorithmic thinking and problem solving by writing C/C++ programs. It initially introduces the basics of the language and structured programming. Then the focus will shift to the C/C++ features that make its power namely low level access to memory via pointers and the illustration of Object-Oriented programming concepts. The final project serves to put in practice all the aspects learned all along the course to produce a significant piece of software with fun purposes and usage.
Course Description from Program Guide	Computer programming and the use of abstractions. Object-oriented programming, fundamental data structures (such as stacks, queues, sets) and data-directed design. Recursion and recursive data structures (linked lists, trees, graphs). Introduction to basic time and space complexity analysis. The course teaches the mechanics of the C, C++ or Java language as well as an example of media library
Goals and Objectives	<ul style="list-style-type: none"> - solving simple to moderate difficulty problems algorithmically - design and write C/C++ structured code solutions - design and write C/C++ object-oriented code solutions - use standard libraries as well as a graphic library as part of code solutions - cooperate with teammate(s) to design and write larger code as solution to more complex problem
Required Knowledge	Prior basic experience with programming and algorithmic thinking is needed. Students will be required to take a test to be allowed to take the course. Please contact the instructor with this regard.

Reference Texts	<ul style="list-style-type: none"> - A tour of C++, Bjarne Stroustrup, Addison-Wesley Professional, 1st edition, 2013, ISBN 978-0321958310. - The C++ programming language, Addison-Wesley Professional, 4th edition, 2013, ISBN 978-0321563842. - Programming: Principles and practices using C++, Addison-Wesley Professional, 2nd edition, 2014, ISBN 978-0321992789. - C++ tutorial: www.learncpp.com - C/C++ reference: www.cplusplus.com
Method of evaluation	<p>25.00% - Group Project(s) 25.00% - Homework /Assignments 25.00% - Midterm exam 25.00% - Final exam</p>
Nature of the assignments	<ul style="list-style-type: none"> - Programming assignments are sets of 3-4 programming exercises or 1 mini-project. - Midterm and final exam consist in 1-2 programming exercises to be solved within a time constrain. Some exercises consist in modifying or completing an existing code. - The project is assigned for the last few weeks and must make use of most of what is learned: recursion, OOP, SDL, ADTs, It's a team project where each team consists of 2 students. Topic should be chosen from the list that will be proposed. Deliverables are: code, report and presentation.
Course Policies	<p>Grading guidelines:</p> <ul style="list-style-type: none"> - A program that does not compile gets a maximum of 20% of the points - A program that runs but produces segmentation faults, irrelevant outputs or does not return/stop gets a maximum of 50% of the points - Programs providing reasonable output with various level mistakes get points deducted accordingly up to 50% of the points - Up to 10% of the points can be deducted for ill-commented and/or ill-indented code. - Late assignment submission costs 5 pts penalty per day. No assignment would be accepted after 5 days of the deadline. <p>Plagiarism:</p> <ul style="list-style-type: none"> - Solutions to assignments need to be the student's original, genuine work. Sharing of code between students and referring to code from online sources will be detected and sanctioned. - For the project, use of a third party code is only allowed according to the code license and after instructor's approval and, of course, with proper credit.
Additional Information	

Tentative Course Schedule

(Time, topic/emphasis & resources)

Week	Lectures	Topic
1	Mon 06/17/2019 Wed 06/19/2019	C++ structured programming basics: development environment and tools, structure of a program, variables, data types, expressions, compound statement, ... C++ structured programming: functions, aggregate data types
2	Mon 06/24/2019 Wed 06/26/2019	C++ structured programming: arrays, strings, file I/O C++ structured programming: pointers, dynamic memory, linked lists
3	Mon 07/01/2019 Wed 07/03/2019	Recursion
4	Mon 07/08/2019 Wed 07/10/2019	Midterm Exam Object Oriented concepts in C++: classification and identification, abstraction and encapsulation, constructors, destructors
5	Mon 07/15/2019 Wed 07/17/2019	Object Oriented concepts in C++: polymorphism, operator overloading
6	Mon 07/22/2019 Wed 07/24/2019	Object Oriented concepts in C++: inheritance and genericity
7	Mon 07/29/2019 Wed 07/31/2019	Simple and Fast Multimedia Library (SFML) Standard Template Library (STL)
8	Mon 08/05/2019 Wed 08/07/2019	Standard Template Library (STL) Project presentations

Note

The instructor reserves the right to make changes to this syllabus as necessary.