



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	Spring
Academic Year	2016/2017
Semester Start Date	01/22/2017
Semester End Date	05/18/2017
Class Schedule (Days & Time)	09:00 AM - 10:30 AM Mon Wed

Instructor(s)				
Name	Email	Phone	Office Location	Office Hours
Malek Smaoui	Malek.Smaoui@KAUST.EDU.SA	+966128080331 8080331		By appointment. Please email for appointment.

Teaching Assistant(s)	
Name	Email
Anas Ismail	anas.ismail@kaust.edu.sa

Course Information	
Comprehensive Course Description	<p>For students new to programming, this course is intended to familiarize them with algorithmic thinking and solving problems by writing C/C++ programs. It starts by introducing the basics of the language and structured programming.</p> <p>For students with already some programming experience (eventually with other languages), focus will be on the C/C++ features that make its power namely low level access to memory via pointers and the illustration of Object-Oriented programming concepts.</p> <p>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.</p>
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	<ul style="list-style-type: none"> - basic algorithmic thinking - basic calculus

Reference Texts	Textbook: - Programming Abstractions in C++, Eric Roberts, Prentice Hall, 2013. Additional references: - C++ tutorial: www.learncpp.com - C/C++ reference: www.cplusplus.com
Method of evaluation	25.00% - Final exam 25.00% - Midterm exam 30.00% - Homework /Assignments 20.00% - Course Project(s)
Nature of the assignments	- 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 (typically 4 weeks) and mustmake 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	-All delivered programs are expected to compile with the GNU C compiler (gcc) and run correctly in a Linux environment. You can still develop (write and debug) your code in another OS and Integrated Development Environment (IDE): Xcode (or other) on Mac or MS Visual Studio, CodeBlocks, DevC++, ... on windows. However, check that the final code runs properly on therequested environment before submission -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 eventually more or less serious 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.
Additional Information	

Tentative Course Schedule

(Time, topic/emphasis & resources)

Week	Lectures	Topic
1	Mon 01/23/2017 Wed 01/25/2017	C++ structured programming basics: development environment and tools, structure of a program, variables, data types, expressions, compound statement, ...
2	Mon 01/30/2017 Wed 02/01/2017	C++ structured programming: functions, aggregate data types
3	Mon 02/06/2017 Wed 02/08/2017	C++ structured programming: arrays, strings, file I/O
4	Mon 02/13/2017 Wed 02/15/2017	C++ structured programming: pointers, dynamic memory, linked lists
5	Mon 02/20/2017 Wed 02/22/2017	Recursion
6	Mon 02/27/2017 Wed 03/01/2017	Object Oriented concepts in C++: classification and identification, abstraction and encapsulation, constructors, destructors
7	Mon 03/06/2017 Wed 03/08/2017	Midterm exam
8	Mon 03/13/2017 Wed 03/15/2017	Object Oriented concepts in C++: polymorphism, operator overloading
9	Mon 03/20/2017 Wed 03/22/2017	Object Oriented concepts in C++: inheritance and genericity
10	Mon 03/27/2017 Wed 03/29/2017	Simple DirectMedia Library
11	Mon 04/03/2017 Wed 04/05/2017	Simple DirectMedia Library
12	Mon 04/10/2017 Wed 04/12/2017	Standard Template Library
13	Mon 04/17/2017 Wed 04/19/2017	Standard Template Library
14	Mon 04/24/2017 Wed 04/26/2017	Project presentations
15	Mon 05/01/2017 Wed 05/03/2017	Final Exam
16	Mon 05/08/2017 Wed 05/10/2017	
17	Mon 05/15/2017 Wed 05/17/2017	
18		

Note

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