



## Course Syllabus: Programming Methodology and Abstractions - CS 207

<b>Division</b>	Computer, Electrical and Mathematical Sciences & Engineering
<b>Course Number</b>	CS 207
<b>Course Title</b>	Programming Methodology and Abstractions
<b>Academic Semester</b>	Fall
<b>Academic Year</b>	2018/2019
<b>Semester Start Date</b>	08/26/2018
<b>Semester End Date</b>	12/11/2018
<b>Class Schedule</b> (Days & Time)	01:00 PM - 02:30 PM   Mon Thu

Instructor(s)				
Name	Email	Phone	Office Location	Office Hours
Malek Smaoui	Malek.Smaoui@KAUST.EDU. SA	+966128080331		By appointments, in the mornings 9:00 am to 12:00 pm. Please email for appointments the day before at latest.

Teaching Assistant(s)	
Name	Email

Course Information	
<b>Comprehensive Course Description</b>	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 in a structured programming fashion. 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.
<b>Course Description from Program Guide</b>	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
<b>Goals and Objectives</b>	<ul style="list-style-type: none"> <li>- solving simple to moderate difficulty problems algorithmically</li> <li>- design and write C/C++ structured code solutions</li> <li>- design and write C/C++ object-oriented code solutions</li> <li>- use standard libraries as well as a graphic library as part of code solutions</li> <li>- cooperate with teammate(s) to design and write larger code as solution to more complex problem</li> </ul>
<b>Required Knowledge</b>	<ul style="list-style-type: none"> <li>- basic algorithmic thinking</li> <li>- basic calculus</li> </ul>

<b>Reference Texts</b>	<p>Essential C++ knowledge:</p> <ul style="list-style-type: none"> <li>- A tour of C++, Bjarne Stroustrup, Addison-Wesley Professional, 1st edition, 2013, ISBN 978-0321958310.</li> </ul> <p>Comprehensive language and standard libraries overview:</p> <ul style="list-style-type: none"> <li>- The C++ programming language, Bjarne Stroustrup, Addison-Wesley Professional, 4th edition, 2013, ISBN 978-0321563842.</li> </ul> <p>Algorithmic thinking and basic programming principles for beginners:</p> <ul style="list-style-type: none"> <li>- Programming: Principles and practices using C++, Bjarne Stroustrup, Addison-Wesley Professional, 2nd edition, 2014, ISBN 978-0321992789.</li> </ul> <p><i>Note: All above titles are available online from the KAUST library</i></p> <p>Online tutorial with simple examples:</p> <ul style="list-style-type: none"> <li>- C++ tutorial: <a href="http://www.learncpp.com">www.learncpp.com</a></li> </ul> <p>C/C++ libraries documentation:</p> <ul style="list-style-type: none"> <li>- C/C++ reference: <a href="http://www.cplusplus.com">www.cplusplus.com</a></li> </ul>
<b>Method of evaluation</b>	<p><b>25.00%</b> - Group Project(s)  <b>25.00%</b> - Homework /Assignments  <b>25.00%</b> - Midterm exam  <b>25.00%</b> - Final exam</p>
<b>Nature of the assignments</b>	<ul style="list-style-type: none"> <li>- Programming assignments are sets of 3-4 programming exercises or 1 mini-project.</li> <li>- 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.</li> <li>- The project is assigned for the last few weeks (typically 4 weeks) and must make use of most of what is learned: OOP, STL, SFML, .... 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.</li> </ul>
<b>Course Policies</b>	<ul style="list-style-type: none"> <li>- Using a programming environment (namely operating system and compiler) different from that used by the grader may result in inconsistencies in the program output or even the program not compiling or crashing. It is preferable to check that the program works properly in an environment similar to that of the grader prior to submission. The grader will specify the grading environment at the beginning of the course.</li> <li>- A program that does not compile gets a maximum of 20% of the points</li> <li>- A program that runs but produces segmentation faults, irrelevant outputs or does not return/stop gets a maximum of 50% of the points</li> <li>- Programs providing reasonable output with various level mistakes get points deducted accordingly up to 50% of the points</li> <li>- Up to 10% of the points can be deducted for ill-commented and/or ill-indented code.</li> <li>- Late assignment submission costs 5 pts penalty per day. No assignment would be accepted after 5 days of the deadline.</li> </ul>
<b>Additional Information</b>	

## Tentative Course Schedule

*(Time, topic/emphasis & resources)*

Week	Lectures	Topic
1	Mon 08/27/2018 Thu 08/30/2018	C++ structured programming basics: development environment and tools, structure of a program, variables, data types, expressions, compound statement, ...
2	Mon 09/03/2018 Thu 09/06/2018	C++ structured programming: functions, aggregate data types
3	Mon 09/10/2018 Thu 09/13/2018	C++ structured programming: arrays, strings, file I/O
4	Mon 09/17/2018 Thu 09/20/2018	C++ structured programming: pointers, dynamic memory, linked lists
5	Mon 09/24/2018 Thu 09/27/2018	Recursion
6	Mon 10/01/2018 Thu 10/04/2018	Midterm exam
7	Mon 10/08/2018 Thu 10/11/2018	Object Oriented concepts in C++: classification and identification, abstraction and encapsulation, constructors, destructors
8	Mon 10/15/2018 Thu 10/18/2018	Object Oriented concepts in C++: polymorphism, operator overloading
9	Mon 10/22/2018 Thu 10/25/2018	Object Oriented concepts in C++: inheritance and genericity
10	Mon 10/29/2018 Thu 11/01/2018	Simple and Fast Multimedia Library
11	Mon 11/05/2018 Thu 11/08/2018	Simple and Fast Multimedia Library
12	Mon 11/12/2018 Thu 11/15/2018	Standard Template Library
13	Mon 11/19/2018 Thu 11/22/2018	Standard Template Library
14	Mon 11/26/2018 Thu 11/29/2018	Project presentations
15	Mon 12/03/2018 Thu 12/06/2018	Final Exam
16	Mon 12/10/2018	
17		
18		

### Note

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