

University of Macau
Undergraduate Civil Engineering Programme

Coordinating Unit:	Department of Civil and Environmental Engineering, Faculty of Science and Technology		
Supporting Unit(s):	Nil		
Course Code:	CEEB110	Year of Study:	1
Course Title:	Computer Science		
Compulsory/Elective:	Compulsory		
Course Prerequisites:	Nil		
Prerequisite Knowledge:	Nil		
Duration:	One semester	Credit Units:	3
Class/Laboratory Schedule:	Two hours of lecture and two hours of tutorial/laboratory per week.		
Laboratory/Software Usage:	Microsoft Visual C++		
Course Description:	This course introduces computer science with basic logic and fundamental programming techniques, and demonstrates how to improve procedural solutions by the object-oriented programming (OOP) approach as well. Fundamental of programming includes primitive data types, expressions, control statements, functions and arrays, which prepare students to learn object-oriented programming. Both theoretical concepts and practical skills are involved; students can experience problem solving by formulating programmatic solutions for real problems and translate them into programs with the programming language C/C++.		
Course Objectives:	<ol style="list-style-type: none"> 1. Introduce the essential of computers, programs, and C++. 2. Learn basic concepts of software design principles and problem solving skills. 3. Learn elementary and advanced techniques of C++ programming. 4. Study the fundamentals of object-oriented programming. 		
Learning Outcomes (LO):	<p>Upon completion of this course, students should be able to:</p> <ol style="list-style-type: none"> 1. recognize the fundamentals of computers, programs, C++, software design principles and problem solving skills. [POs: a, l] 2. use the basic elementary of C++ programming, such as expressions, assignments, control and iterative statements. [POs: a, c, l] 3. deal with more advanced C++ programming techniques, including building functions, creating arrays and working with files. [POs: a, c, l] 4. realize the fundamentals of object-oriented programming, like using libraries, defining classes and invoking methods. [POs: a, c, l] 5. apply learned C++ programming techniques to write computer programs solving specific problems. [POs: a, c, l] 		
Texts & References: * recommended textbook	<ol style="list-style-type: none"> 1. Y. Daniel Liang, <i>Introduction to Programming with C++</i>, Third Edition, Pearson Education Limited, 2014*. 2. http://www.cs.armstrong.edu/liang/cpp3e/ 3. http://www.cs.ust.hk/~dekai/library/ECKEL_Bruce/ 4. http://msdn.microsoft.com/en-us/library/Aa733747/ 5. http://www.cplusplus.com/doc/tutorial/ 		
Student Assessment:	<ul style="list-style-type: none"> • In-class/lab exercises: 10% • e-Quizzes: 10% • Assignments: 20% • Mid-term exam: 20% • Final exam: 40% 		

Learning Outcome Assessment:	<ul style="list-style-type: none"> • Class and lab participation, homework, quizzes and examinations. • Course evaluation
Pedagogical Methods:	<input checked="" type="checkbox"/> Lecture <input type="checkbox"/> Guest speakers <input type="checkbox"/> Case study <input type="checkbox"/> Role playing <input type="checkbox"/> Student presentation <input type="checkbox"/> Project <input type="checkbox"/> Simulation game <input checked="" type="checkbox"/> Exercises and problems <input type="checkbox"/> Service learning <input type="checkbox"/> Internship <input type="checkbox"/> Field study <input type="checkbox"/> Company visits <input type="checkbox"/> e-learning <input checked="" type="checkbox"/> Independent study <input type="checkbox"/> Others:

Major Assessment Methods:	Case Study	Role Playing	Student Presentation	Individual project/paper	Group project/paper	Simulation Game	Exercises & problems	Service learning	Internship	Field Study	Company visits	Written examination	Oral examination	Others (please specify):
Class Participation/ Discussion							✓							
Assignment(s)							✓							
Test(s)												✓		
Examination												✓		
Others:(please specify) e-Quizzes							✓							
Course Web: (if any)	Course materials are available in UMMoodle (http://webcourse.umac.mo/).													

Course Content: (topic outline)	Week no.	Topics	LO no.
	1	Introduction to computers, programs, and C++ Review computer basics and programs; realize the fundamental of C++; understand software development cycle; and develop a simple C++ program for console output using Visual C++.	1
	2,3	Elementary programming Study basic data types, their declarations and initializations; use variables to store data; read input from keyboard; program with assignment statements and expressions; familiar with C++ documentation, programming style; and experience various errors and debug logic errors.	1, 2
	3,4	Selections Implement selection control using if and switch statements; combine conditions using logical operators; write expressions using the conditional operator; format output using stream manipulators; and examine the rules governing operator precedence and operator associativity.	2
	4,5	Loops Write loops using do-while, while and for statements; control a loop with the user confirmation or a sentinel value; write nested loops; learn the techniques for minimizing numerical errors; and implement program control with break and continue.	2
	6	Function basics Define and invoke different types of functions; use function prototypes for function headers; know how to pass arguments; create header files for reusing functions; develop functions for various tasks; and develop applications using C++ functions.	2, 3, 5
	7	Midterm Exam	
	8,9	Advanced function features Experience advanced topics on pass-by-value, pass-by-reference; understand the difference between them; determine the scope of local and global variables; define functions with default arguments; and improve runtime efficiency by using inline functions.	2, 3, 5
	9,10	Arrays Understand the necessity of an array in programming; know how to declare and initialize an array; program common array operations; develop and invoke functions with array arguments; and process string using C-strings.	2,3
	11,12	Objects and classes Describe objects and classes; create objects using constructors; distinguish between instance and static variables and functions; access data fields and invoke functions using the object member access operator; and declare private data fields for data field encapsulation and make classes easy to maintain.	2, 3, 4, 5
	13	Files and streams Learn ifstream, ofstream, and fstream classes for processing and manipulating files; read and write data using the getline, get and put functions; study functions to test file existence and the end of a file; and open a file for both input and output to update files.	3,5
	14	Final Review	

Percentage Content of:	Mathematics and Basic Sciences	Engineering Subjects	Complementary Studies	Total	
	0	100	0	100	
Timetabled work in hours per week:	Lecture	Tutorial	Laboratory	Other	Total
	2	---	2	---	4
Contribution to Program Outcomes:	Programme Outcomes			Contribution to POs	
				Primary	Secondary
	(a) an ability to apply knowledge of mathematics, science, and engineering appropriate to the degree discipline				✓
	(b) an ability to design and conduct experiments, as well as to analyse and interpret data				
	(c) an ability to design a system, component or process to meet desired needs within realistic constraints, such as economic, environmental, social, political, ethical, health and safety, manufacturability and sustainability			✓	
	(d) an ability to function on multi-disciplinary teams				
	(e) an ability to identify, formulate and solve engineering problems				
	(f) an ability to understand professional and ethical responsibility				
	(g) an ability to communicate effectively				
	(h) an ability to understand the impact of engineering solutions in a global and societal context, especially the importance of health, safety and environmental considerations to both workers and the general public				
	(i) an ability to stay abreast of contemporary issues				
	(j) an ability to recognise the need for, and to engage in life-long learning				
(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice appropriate to the degree discipline					
(l) an ability to use the computer/IT tools relevant to the discipline along with an understanding of their processes and limitations			✓		
Course Instructor:	Mr. Miguel Gomes da Costa Junior				