Subject Description Form

Subject Code	EIE2111
Subject Title	Computer Programming
Credit Value	6
Level	2
Pre-requisite/ Co-requisite/ Exclusion	Nil
Objectives	1. To introduce the fundamental concepts of computer programming. 2. To equip students with sound skills in C/C++ programming language. 3. To equip students with techniques for developing structured computer programs. 4. To demonstrate the techniques for implementing engineering applications using computer programs.
Intended Subject Learning Outcomes	**Upon completion of the subject, students will be able to:** Category A: Professional/academic knowledge and skills 1. Familiarize with at least one C/C++ programming environment. 2. Be proficient in using the basic constructs of C/C++, such as variables and expressions, looping, arrays and pointers, to develop a computer program. 3. Able to develop a structured and documented computer program. 4. Understand the fundamentals of object-oriented programming and be able to apply it in computer program development. 5. Able to apply the computer programming techniques to solve practical engineering problems. Category B: Attributes for all-roundedness 6. Solve problems by using systematic approaches. 7. Write technical reports and present the findings. 8. Learn team working skills.
Subject Synopsis/ Indicative Syllabus	**Syllabus:** 1. Introduction to programming Software components of a computer – Operating system, directories, files. Evolution of programming languages. Programming environment – Compiler, linker and loader. Building the first program – Hello World. 2. Bolts and Nuts of C/C++ Preprocessor, program codes, functions, comments. Variables and constants. Expressions and statements. Operators. 3. Program Flow Control If, else, switch, case. Looping – for, while, do. Functions, parameters passing, return values. Local and global variables. Scope of variables. 4. Program Design and Debugging Structured program design. Improving program readability. Flow chart. Modular programming – static library. Programming bugs, errors, mistakes and code rot. Exceptions and debugging. Case study: Using Visual C++ debugger. 5. Basic Object Oriented Programming Objects and classes. Encapsulation. Private versus public. Implementing class methods. Constructors and destructors.

6. Pointer and Array
   The stack and free store. Create and delete objects in free store. Pointer arithmetic. Passing function arguments by pointer. Returning values by pointer. Array of Objects. Multidimensional array. Array and pointer. Array of pointers. Pointer of array. Character array – Strings. Command line processing.

7. Dynamic Data Structures
   Linked list. Basic operations. Other dynamic data structures (stacks, queues and trees).

8. File Processing
   Files and streams. Create a sequential file. Read data from a sequential file. Updating sequential files. Create a random-access file. Write data to a random-access file. Read data from a random-access file.

9. Graphical User Interface (GUI)
   Introduction to C#. Some Simple GUI programs. C# with C++. Read/write text files by using C#. Multiple Forms. Windows Graphical Device Interface (GDI).

10. Using C/C++ in Engineering Applications
   Solving numerical problems using C/C++.

Teaching/ Learning Methodology

Teaching and Learning Method	**Intended Subject Learning Outcome**	**Remarks**
Lectures	1, 2, 3, 4, 5	Fundamental principles and key concepts of the subject are delivered to the students
Laboratory	1, 2, 3, 4, 5, 6	Students will be able to clarify concepts and to have a deeper understanding of the lecture material. Problems are given to be solved.

Assessment Methods in Alignment with Intended Learning Outcomes

Specific Assessment Methods/Tasks	**% Weighting**	**Intended Subject Learning Outcomes to be Assessed (Please tick as appropriate)**							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Continuous Assessment									
• Quizzes	8%	✓	✓	✓	✓	✓	✓		
• Laboratory Exercises	10%	✓	✓	✓	✓	✓	✓		
• Assignments	10%	✓	✓	✓	✓	✓	✓	✓	
• Mini-project	30%	✓	✓	✓	✓	✓	✓	✓	✓
• Tests	42%	✓	✓	✓	✓	✓	✓		
Total	100%								

For this subject, students need to go through two 2-hours programming tests in which students will be asked, within the allowed time period, to develop a set of computer programs using C/C++ programming language to solve a problem. These two tests are worth 42% of the total marks.

Besides, students need to finish a mini-project in this subject. Students are expected to spend not less than 35 hours of self-studying in order to finish the mini-project. The mini-project is worth 30% of the total marks.

The remaining 28% of marks are allotted to assignments, quizzes and laboratory exercises that will be given during and after the classes.

Explanation of the appropriateness of the assessment methods in assessing the intended learning outcomes:

Specific Assessment Methods/Tasks	**Remark**
Laboratory Exercises/Quizzes	Students will be able to clarify concepts and to have a deeper understanding of the lecture material. Problems are given to be solved.
Assignments	Students will be able to clarify concepts and to have a deeper understanding of the lecture material. Problems are given to be solved.
Mini-Project	Students will be able to clarify concepts and to have a deeper understanding of the lecture material. Problems are given to be solved.
Tests	Evaluate students' ability in applying computer programming skills learned in classes. Problems are given to be solved.

Student Study Effort Expected	**Class contact (time-tabled):**	
	• Lecture/Tutorial/Laboratory/Practice Classes	78 Hours
	Other student study effort:	
	• Lecture: preview/review of notes; homework/assignment; preparation for test/quizzes/examination	78 Hours
	• Tutorial/Laboratory/Practice Classes: preview of materials, revision and/or reports writing	78 Hours
	Total student study effort:	**234 Hours**
Reading List and References	**Textbooks:** 1. H.M. Deitel and P.J. Deitel, *C++ How To Program*, 10th ed., Prentice-Hall, 2017. **Reference Books:** 1. K. Gregory, *Microsoft® Visual C++® .NET 2003 Kick Start*, Sams Publishing, 2003. 2. H.M. Deitel, P.J. Deitel, J.P. Liperi and C.H. Yaeger, *Visual C++.NET How to Program*, Prentice-Hall, 2004.	
Last Updated	October 2019	
Prepared by	Dr Lawrence Cheung	