

Subject Description Form

Subject Code	ENG2002
Subject Title	Computer Programming
Credit Value	3
Level	2
Prerequisite	Nil
Objectives	<ul style="list-style-type: none"> ▪ To introduce the fundamental concepts of computer programming. ▪ To equip students with sound skills in C/C++ programming language. ▪ To equip students with techniques for developing structured and object-oriented computer programs. ▪ To demonstrate the techniques for implementing engineering applications using computer programs.
Intended Learning Outcomes	<p>Upon completion of the subject, students will be able to:</p> <ol style="list-style-type: none"> a. Familiarize themselves with at least one C/C++ programming environment; b. Be proficient in using the basic constructs of C/C++ to develop a computer program; c. Be able to develop a structured and documented computer program; d. Understand the fundamentals of object-oriented programming and be able to apply it in computer program development; e. Be able to apply the computer programming techniques to solve practical engineering problems;
Contribution to Programme Outcomes (Refer to Part I Section 10)	<ul style="list-style-type: none"> ▪ Programme Outcome 8: Demonstrate an ability to use the computer/IT tools relevant to the Biomedical Engineering (BME) discipline along with an understanding of their processes and limitations. (Teach and Practice)
Subject Synopsis/ Indicative Syllabus	<ul style="list-style-type: none"> ▪ Introduction to Programming: Components of a computer; programming environment; process of application development. ▪ Bolts and Nuts of C/C++: Preprocessor; program code; functions; comments; variables and constants; expressions and statements; operators. ▪ Program Flow Control: Branching and looping; function parameters passing; return values; local and global variables; scope of variables.

	<ul style="list-style-type: none"> ▪ Program Design and Debugging - Structured program design; Debugging a program. Case study: Using the Visual C++ debugger. ▪ Basic Object Oriented Programming: Objects and classes; private versus public; implementing class methods; constructors and destructors. ▪ Pointer and Array: Stack and free store; create and delete objects in the free store; pointer arithmetic; passing function arguments by pointer; returning values by pointer; array of objects; array and pointer; array of pointers; pointer of array; character array; command-line processing. ▪ Stream I/O: Input and output as streams; file I/O using streams. 		
Teaching and Learning Methodology	Teaching and Learning Method	Intended Subject Learning Outcome	Remarks
	Lectures, supplemented with short quizzes	b,c,d	Students are introduced to the knowledge of computer programming through explanation and illustrative examples. Comprehension of the knowledge is strengthened with short quizzes. Students will be able to monitor the skills of using C/C++ and apply the techniques of developing structured object-oriented applications.
	Laboratories/tutorials where problems are given to students for them to solve	a,b,c,d,e	Students apply what they have learnt in lectures and solve problems in exercises. The purpose is to ensure students have captured the important points. Tutors will aid the lecturer in helping the students finishing the exercises, and interactive Q&A will take place.
	Homework, tests and final examination	a,b,c,d,e	By doing homework, students will develop a firm understanding and comprehension of the knowledge taught. They will analyse given C/C++ applications and apply knowledge in solving problems. For some design type of problems, they will have to synthesize solutions by evaluating different alternatives. To enhance the students' problem solving skill in a given programming environment, open-book programming tests are arranged regularly. To assure students'

			understanding of fundamental concepts, a closed-book final examination is arranged.							
Assessment Methods in Alignment with Intended Learning Outcomes	Specific assessment methods/tasks	% weighting	Intended subject learning outcomes to be assessed (Please tick as appropriate)							
			a	b	c	d	e			
	In-class exercises	10%	√	√	√	√	√			
	Short-quizzes	15%		√	√	√				
	Programming tests	30%	√	√	√	√	√			
	Homework	15%	√	√	√	√	√			
	Final examination	30%	√	√	√	√	√			
	Total	100%								
<p><i>Explanation of the appropriateness of the assessment methods in assessing the intended learning outcomes:</i></p> <p>The short-quizzes are for assessing the understanding of fundamental concepts. The in-class exercises are conducted to help students familiarized with the programming language and skills. The programming tests are for assessing the ability of students on solving computer problems through programming within a specified period. Through doing homework, students will be able to experience how to solve computer problems and design solutions by using a systematic approach. The final examination is for assessing the students' ability on using the programming language and analysing computer problems.</p>										
Student Study Effort Expected	Class contact:							39 Hrs.		
	▪ Lectures, Tests and Quizzes							26 Hrs.		
	▪ Laboratory/Tutorial							13 Hrs.		
	Other student study effort:							71 Hrs.		
	▪ Self-study							57 Hrs.		

	<ul style="list-style-type: none"> ▪ Homework 	14 Hrs.
	Total student study effort	110 Hrs.
Reading List and References	<p><u>Reference Books</u></p> <ul style="list-style-type: none"> ▪ S. Rao, Sams Teach Yourself C++ in One Hour a Day. Indianapolis, IN: Sams, 2012. ▪ P.J. Deitel and H.M. Deitel, C++ How To Program, 8th ed. Boston, MA: Prentice Hall, 2012. ▪ J. Liberty and R. Cadenhead, Sams Teach Yourself C++ in 24 hours (5th ed.) Indianapolis, IN: Sams, 2011. ▪ I Horton, Ivor Hortons Beginning Visual C++ 2010 [electronic resource]. Indianapolis, IN: Wiley, 2010. 	
Date of Last Major Revision	14 July 2014	
Date of Last Minor Revision	June 2015	