1.	**Module Code**	C++
2.	**MODULE TITLE**	Computing
3.	**Schools involved in delivery**	None
4.	**Name of Course**	Certificate of Higher Education in Engineering
5.	**Module Leaders**	Dr Mahmoud Dhimish
6.	**Location for delivery**	Queensgate ISC
7.	**Module Type**	Core
8.	**Credit Rating**	20
9.	**Level**	F
10.	**Learning Methods**	Seminar/ lecture / laboratory and small group work – 50 hours Self-directed study – 150 hours
11.	**Pre-requisites**	None
12.	**Recommended Prior Study**	None
13.	**Co-requisites**	None
14.	**Professional Body Requirements**	None
15.	**Barred Combinations**	None
16.	**Graded or Non Graded**	Graded

17. Synopsis

The aim of this module is to introduce students to the principles of and best practice in software design and development.

18. Outline Syllabus

1. Number systems: Binary, hexadecimal and BCD
2. Procedural versus object-oriented approaches
3. Procedural and object oriented concepts - abstraction, modularisation, polymorphism, inheritance etc..
4. Basic concepts of programs: Format of related procedural and object oriented languages, basic syntax, IDE
5. Variables and constants: variable types (static and non-static) and declaration, data types, constants, global and local.
6. Objects, classes, constructors and methods (inc. virtual/abstract methods)
7. Arithmetic Operators: division and modulus, order of precedence, use of parentheses, compound assignments.
8. Logical and Bitwise Operators and their precedence
9. Control structures: if..else, switch and loops
10. Arrays and pointers
11. Standard libraries and import mechanisms: functions, passing variables and returning values, recursion.
12. File processing: create, read, write and upload files; I/O operations
13. Modelling: breaking down problems, prototypes, UML

14. Program testing (unit, white box and black box) and documentation: use of debug facilities, comments, layout, style issues

19. Learning Outcomes

Knowledge and Understanding Outcomes

On completion of this module students will be able to:

K1. Understand procedural and object-oriented programming approaches and be able to apply them in real-world problem-solving

K2. Apply and implement required operations of related procedural and object oriented languages such as C and C++

K3. Outline and define how to plan software using modelling techniques, test software using debug techniques and document software using comments

Ability Outcomes

On completion of this modules students will be able to:

A1. Create a well-written and well documented computer program from a detailed design specification

A2. Employ a recognised software development method to design and implement software which meets a specified requirement

A3. Communicate the results of their study accurately and reliably and with structured and coherent arguments

20. Assessment Strategy

20.1 *Formative assessment*

Formative assessment will be by means of weekly observation of performance, with appropriate feedback from the tutor.

20.2 *Summative Assessment*

Assessment tasks (including assessment weightings)

Assess ment	LO’s to be met	Type of assessment	Weighting	Duration (if timed teacher assessments)	Word count or equiv. if appropriate
1	K1 K2 K3 A2 A3	Lab assignment	20%		30 hours
2	K1 K2 K3 A2 A3	Lab assignment	20%		30 hours
3	K1 K2 K3 A1 A2 A3	Final Coursework	60%		90 hours

Assessment Criteria

In order to pass the programme students must achieve a minimum of 40% overall in the summative assessments.

All assessments are open to tutor re-assessment and are subject to anonymity where practical.

21. Learning Strategy

Taught aspects of this programme are presented in formal and tutor-led fashion**.** Students will be encouraged to learn individually and in small groups in interactive and laboratory sessions designed to develop their knowledge and skills.

22. Indicative Reading (Latest Editions)

Deitel, P, Deitel, H.M. *C (2011) International Version: Java-How to Program*, Pearson Education Ltd
Duncan, T. (1997) *Electronics for Today and Tomorrow,* Hodder Education.
Gaddis, T. (2014) *C++: From Control Structures through Objects (8th Edition*), Addison-Wesley Professional
Kernighan; Dennis Ritchie (March 1988). The C Programming Language (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.
Liang Y.D. (2010) *Introduction to Java Programming, Comprehensive: International Version*, Pearson Education Ltd
Savitch, W. (2012) *Problem Solving with C++,* Addison-Wesley
Storey, N. (2009) *Electronics: A System Approach (4th Edition),* Prentice Hall.
Stroustrup, B. (2014) *Programming: Principles and Practice using C++ (2nd Edition),* Addison-Wesley Professional