Confirmation that this version of the module specification has been approved by the School Learning and Teaching Committee:

…………26th January 2015……………………………………….(date)

**MODULE SPECIFICATION**

1. **Title of the module**

EL560 Microcomputer Engineering

1. **School or partner institution which will be responsible for management of the module**

Engineering and Digital Arts

1. **Start date of the module**

Revised version start date September 2015

1. **The number of students expected to take the module**

50

1. **Modules to be withdrawn on the introduction of this proposed module and consultation with other relevant Schools and Faculties regarding the withdrawal**

None

1. **The level of the module (e.g. Certificate [C], Intermediate [I], Honours [H] or Postgraduate [M])**

I

1. **The number of credits and the ECTS value which the module represents**

15 (7.5 ECTS)

1. **Which term(s) the module is to be taught in (or other teaching pattern)**

Autumn and Spring

1. **Prerequisite and co-requisite modules**

|  |  |
| --- | --- |
|  EL313  | INTRODUCTION TO PROGRAMMING  |
| or CO320  | JAVA  |
| or CO520  | FURTHER OBJECT-ORIENTED PROGRAMMING  |
| or CO324  | COMPUTER SYSTEMS  |

1. **The programmes of study to which the module contributes**

BEng Electronic and Communications Engineering

BEng Electronic and Communications Engineering with a Year in Industry

BEng Computer Systems Engineering

BEng Computer Systems Engineering with a Year in Industry

MEng in Electronic and Communications Engineering

MEng in Electronic and Communications Engineering with a Year in Industry

MEng in Computer Systems Engineering

MEng in Computer Systems Engineering with a Year in Industry

1. **The intended subject specific learning outcomes**

On successful completion of the module, students will have:

1. A working knowledge of the C programming language;

2. Experience of developing microcomputer applications;

3. A working knowledge of software engineering principles.

These outcomes are related to the programme learning outcomes in the appropriate curriculum maps as follows:

ECE/ECEwInd: A3,A8, B5, C2,C4,C5,C7,C8

CSE/CSEwInd: A3,A8, B4, C2,C4,C6,C8,C9

1. **The intended generic learning outcomes**

On successful completion of the module, students will have acquired key skills in

1. Problem solving

2. Information technology.

These outcomes are related to the programme learning outcomes in the ECE/ECEwInd and CSE/CSEwInd curriculum maps as follows: D1, D5-D7.

1. **A synopsis of the curriculum**

Lecture Syllabus

C PROGRAMMING

variables, operators and control structures.

Good programming practice: layout, naming, software documentation. Scope.

Pointers and Arrays.

Data structures. Memory handling.

MICROCOMPUTER ENGINEERING

The structure of a typical micro computer. The PIC: an overview. The programmer’s model, instruction set and addressing mode. The control unit, stack and stack pointers and program counter. The operation of a microcomputer: fetch and execute activities. Translating C into assembler. How to run a C program on the PIC. The efficiency of C programs: memory requirements and operational speed. An introduction to input/output. Accessing interface registers in C. Simple input/output using the switches and lights. Bit testing and bit manipulation. Generating waveforms. Time delays using program loops and timers. Controlling peripherals. Determining peripheral status. Serial ports. The interaction of hardware and software during polled input/output. The principles of input/output using interrupts. Interrupt service routines: how they are instigated and typical activities. The implementation of interrupts in C. Handling several interrupt sources. A Keyboard interface. Polling vs Interrupts. Flow of data and control in a memory reference instruction. The structure of memory

- conceptual, addressing, decoding

- memory components

- multiple memory chips

- chip enable, implementation strategies

SOFTWARE ENGINEERING

Software Engineering Process: lifecycle models. Software requirements engineering: basic concepts and principles, requirements engineering process, requirements elicitation, requirements analysis, requirements validation, requirements management. Software design: basic concepts and principles, software architecture, design notations, design strategies and methods (object-oriented, function-oriented, real-time systems). Software testing: basic concepts and principles, testing process, test planning, testing strategies and techniques.

Coursework

WORKSHOPS

4 two-hour assessed workshops support the C Programming language component.

Workshop 1: Control structures and program flow.

Workshop 2: Pointers and arrays.

Workshop 3: Use of structures for data representation.

Workshop 4: Mini project involving the development of a software solution for a problem according to given specifications.

ASSIGNMENT - INTRODUCTION TO MICROCOMPUTERS

One assessed and five non-assessed lectures.

LABORATORY - MICROCOMPUTER ENGINEERING

This one-day laboratory uses the PCs with PIC microcomputers. Assessed.

EXAMPLES CLASS - SOFTWARE ENGINEERING

One assessed examples class.

1. **Indicative Reading List**

Recommended Reading

Jasio, LD (2011) Programming 16-bit PIC Microcontrollers in C: Learning to Fly the PIC 24, Oxford: Newnes

Background Reading

Van Sickle, T (1994) Programming microcontrollers in C, California: Hightext.

1. **Learning and Teaching Methods, including the nature and number of contact hours and the total study hours which will be expected of students, and how these relate to achievement of the intended module learning outcomes**

There will be 31 hours of lectures, four workshops each of 2 hours duration, a 7 hour laboratory experiment and 1 examples class. The lectures and practical classes address learning outcomes 11.1, 11.2 and 12.1 whilst the software engineering lectures and examples class addresses learning outcomes 11.3. and 12.2

Total Contact Hours: 47

Independent Study Hours: 103

Total Study Hours: 150

1. **Assessment methods and how these relate to testing achievement of the intended module learning outcomes**

Assessment of the module is by examination (60%) and coursework (40%). The examination duration is 3 hours.

The coursework mark is proportioned as follows:

C-programming workshops (35%)

Intro to microcomputers assignment (20%)

Microcomputer Laboratory (30%)

Software engineering assessed examples class (15%)

The coursework assesses learning outcomes 11.1, 11.2 and 11.3 whilst the examination assesses learning outcomes 11.2 and 11.3

The examination, C-programming workshops and examples classes will assess learning outcome 12.1. The programming workshops and microcomputer laboratory will assess learning outcome 12.2.

*In order to obtain credit for this module, the coursework mark and the exam mark must each be greater than or equal to 30% as well as achieving the pass mark for the module.*

*This module will only be considered for compensation if the coursework mark and exam mark are each greater than 30%*

1. **Implications for learning resources, including staff, library, IT and space**

None

1. **The School recognises and has embedded the expectations of current disability equality legislation, and supports students with a declared disability or special educational need in its teaching. Within this module we will make reasonable adjustments wherever necessary, including additional or substitute materials, teaching modes or assessment methods for students who have declared and discussed their learning support needs. Arrangements for students with declared disabilities will be made on an individual basis, in consultation with the University’s disability/dyslexia support service, and specialist support will be provided where needed.**
2. **Campus(es) or Centre(s) where module will be delivered:**

Canterbury