Module Type		Taught		
Level		4	Credit Value	20
Programme(s) in which module is available		BEng: Electrical and Electronic Engineering; Communication Engineering;		
Involvement of Other Schools		None		
Resource Split		N/A		
Name of Module Co-ordinator		Vladimir Mezentsev		
Related Modules	Pre-requisites	None; students are expected to have a scientific background but not all will have A-level standard Mathematics.		
	Co-requisites	None		
	Prohibited Combinations	None		

Aims of the Module

- To examine the issues involved in the design of software: for solving engineering problems and to provide the functionality of programmable electronic systems.
- To develop an appreciation and understanding of the process and techniques of software design, implementation and testing
- To introduce the principal control flow structures and data structures used in the construction of computer programs, and the software engineering principles required to produce sound software, fit-for-purpose.
- To expose students to 'C' as a programming language to implement these design aspirations.

Summary of Content

- Role of software for computing engineering problems.
- The key constructs of 'C':
- Language basics: Getting started
  - Functions
  - Arrays and strings
  - Variables and arithmetic expressions
  - Control flow: conditionals, loops, switches
  - Functions and Program structure Pointers and Arrays
  - Input/output
- Basics of Matlab
- Unix system interface
- Basic principles of software engineering
- Examples of common data structures
- Interaction with the real-world

Summary of Methods and Frequency of Teaching

Contact hours spread over teaching periods 1 and 2:	Lectures:	30 x1 hour
	Tutorials:	6 x 1 hour
	Design exercises (lab sessions):	18 x 2 hours
	Private study:	128 hours

Summary of Methods of Assessment and Feedback (including formative feedback)

The module will be assessed by a 1.5 hour exam (30%) and continuous assessment, comprising design exercises (50%) and class tests (20%).
Feedback will be provided according to School guidelines. 1-to-1 feedback will be provided orally in the laboratory, in presentations and demonstrations, and by written comments on written reports. Generic feedback will be provided orally in class and via Blackboard.

Module Outcomes – what the student should gain from successful completion of the module:	Learning and Teaching and Assessment Strategies to enable outcomes to be achieved and demonstrated

	Learning and Teaching Methods	Assessment Methods
Knowledge and Understanding of • Software requirements analysis, design and construction • Software testing	Lectures, recommended reading, design exercises	Examination, class test and design reports and documentation
Intellectual Skills • Requirements analysis • Problem solving using computer programs.	Lectures, design exercises, tutorial problems, recommended reading	Examination, class test and design reports and documentation
Professional/Subject Specific Skills • Plan and construct computer programs. • Use appropriate software design, implementation and testing techniques. • Prepare a technical report. .	Lectures, design exercises, tutorial problems	Examination, class test and design reports and documentation
Transferable Skills • Problem solving analytical skills. • Self-study. • Management of time and resources. • Documentation writing.	Lectures, design exercises, tutorial problems	Design reports and documentation

Introductory Learning Resources	Module information, tutorial material and generic feedback will be available on Blackboard
Core Texts, reading lists	Extensive lecture notes available on Blackboard “Practical C Programming”, Steve Oualline, 3rd Edition, O'Reilly, 1997,
Specification completed by: Date	Geof Carpenter 28.7.09
Date module approved by Teaching Committee(s)	Original version June 2007