# MODULE CODE CFP 2119
# MODULE TITLE WORKSHOP

| | |
|---|---|
| **SCHOOL(S) INVOLVED** | Computing and Engineering |
| **NAME OF COURSE(S)** | Undergraduate Computing/Software Development Suite |
| **MODULE LEADER** | Paula Sturdy |
| **LOCATION** | Queensgate |
| **MODULE TYPE** | Core |
| **CREDIT RATING** | 20 credits |
| **LEVEL** | Foundation |
| **LEARNING METHODS** | **Workshop/Tutorials/Practicals**: 48 hrs<br>**Unsupervised learning**: 152 hrs |
| **PRE-REQUISITES** | None |
| **RECOMMENDED PRIOR STUDY** | None |
| **CO-REQUISITES** | None |
| **PROFESSIONAL BODY REQUIREMENTS** | N/A |
| **GRADED** | graded |
| **BARRED COMBINATIONS** | None |

**MODULE SYNOPSIS**

This module will serve as a forum for the analysis, discussion, and application of materials presented in the other core modules of the stage. This could include (but is not limited to) the application of computing science theory and practice; modelling and prototyping of computer systems; development of working solutions to problems; and hands-on practical experience of computer hardware and networks. The module will also provide an opportunity for students to develop core skills such as command line unix, file transfer, and basic security considerations including passwords and firewalls. Finally, the modules will be used to provide 'tasters' of the different courses within the undergraduate computing/software development suite.

**OUTLINE SYLLABUS**

Although workshop modules do not normally have lecture programmes associated with them it is envisaged that this module will include a set of keynote lectures firstly to introduce some core material used at the beginning of the module and later to help to draw together material from the other modules studied on the stage. Topics for study on the module include:

- Command line Unix
- Basic computer security
- Tools to support the application of computing science (e.g. Kara)
- Tools to support the modelling and prototyping of software systems (e.g. NetBeans)
- Introductory sessions (“tasters”) on Software Development, Computing Science, Secure and Forensic Computing, and Internet Systems Development.

**LEARNING OUTCOMES**

1.Knowledge and understanding

Upon completion of this module the learner will be able to:

1.1 Recognise the interrelationships in the materials taught in the modules on stage 1 of the course
1.2 Be able to draw upon the appropriate materials from these modules to solve problems and carry out investigations relevant to the course of study.

2. Abilities

On completion of this module the learner will be able to:

2.1 Apply the concepts and theories taught in the modules on stage 1 of the course to the modelling, prototyping, development, and testing of software systems which execute on real hardware and networks.
2.2 Comment critically both on the quality of the artefacts produced for 2.1 above and on the appropriateness and efficacy of the various approaches which can be applied to a given problem.

## ASSESSMENT STRATEGY

### Formative Assessment

The practical exercises and case-studies used throughout the module will provide the learners with formative assessment and will include “dry run” exercises in preparation for elements of the coursework.

### Summative Assessment

One on-going portfolio style assignment set by or in consultation with the leaders of the other modules which make up stage 1 of the pathway. This will involve the modelling, prototyping, development, testing, and evaluation of a software system. The staged deliveries will allow for feedback and interim grades to be provided as the work progresses. This work will address all learning outcomes.

### Assessment Criteria

The portfolio will be assessed using criterion-referencing and will be judged on the quality of the modelling, the efficiency and quality of the software, and the efficacy of the evaluation.

## LEARNING STRATEGY

Appropriate use of practical hands-on work, discovery strategies, and reinforcement of material covered elsewhere. Students will be required to complete a set of increasingly complex exercises and mini-projects requiring the synthesis and practical application of the theoretical concepts taught elsewhere on the stage.

## INDICATIVE REFERENCES

Relevant Learning materials for the various tools used, e.g.

- Kara on-line teaching materials
- SunONE reference guides
- BlueJ reference guides
- Unix command line 'man' pages

http://java.sun.com
http://www.bluej.org