MODULE CODE CFS 2112
MODULE TITLE WRITING SOFTWARE

SCHOOL(S) INVOLVED	Computing and Engineering
NAME OF COURSE(S)	Core (MEng Software Engineering, MEng European Computer Science, BSc Software Development, BSc Software Development with …, BSc Computing Science, BSc Computing, BSc Secure and Forensic Computing, BSc Internet Systems Development, BSc Computer Games Programming)
MODULE LEADER	Paula Sturdy
LOCATION	Queensgate
CREDIT RATING	20 credits
LEVEL	Foundation
LEARNING METHODS	**Lectures**: 24 hrs **Tutorials/Practicals**: 48 hrs **Unsupervised learning**: 128 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 introduce the concepts of structured programming, object oriented programming, algorithm design, data structure design, and program implementation, testing, and documentation.

OUTLINE SYLLABUS

Introduction to computers, language, and syntax.
Pseudocode for representation of algorithms.
Structured Programming (sequence, selection, and iteration).
Data modelling (base types, structured types, user defined types).
Encapsulated types - introduction to objects.
Procedures, functions, parameter passing, variable scope.
Coding standards and documentation.
Program testing, verification and validation.
Further Object Orientation (reuse and inheritance).
Recursion.
Dynamic data structures (lists and trees).
Applied data structures (eg. stacks and queues).

LEARNING OUTCOMES

1.Knowledge and understanding
Upon completion of this module the learner will be able to:

1.1 Outline the concepts of procedural computer programming (sequence, selection, and iteration and base and user defined data types)
1.2 Outline the concepts of Object Oriented programming including encapsulation and inheritance
1.3 Formulate a range of algorithms for manipulating data structures.
1.4 Recognise the concept of correctness and verification and validation in testing computer programs.

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

2.1 Analyse problems and develop algorithms using a design approach (e.g. functional decomposition or object oriented design)
2.2 Implement, test, and debug computer programs using the chosen language.
2.3 Write documentation and software with reference to the standards of the chosen technologies (eg. design documentation, annotated program listings, test strategies, and actual test results)
2.4 Design and develop programs that manipulate dynamic data structures.

ASSESSMENT STRATEGY

Formative Assessment

Formative assessment will be to develop a basic program that demonstrates object interaction

Summative Assessment

Summative assessment will be by means of two courseworks, each contributing 50% of the marks for the module.

The first assignment will involve the development of a portfolio of programming exercises and will address Learning Outcomes 1.1, 1.3, 2.1, 2.2 and 2.3.

The second assignment will involve the development of a program incorporating the manipulation of dynamic data structures and will address all learning outcomes.

Assessment Criteria

Both courseworks will be assessed using criterion-referencing. Each will be judged on the quality and adherence to industry standards of: the computer program, the testing of the program, and the documentation of the code.

LEARNING STRATEGY

A weekly lecture to introduce the theoretical foundations of the module, plus a tutorial to further explore the issues through the use of paper-based exercises and a practical to support the applied aspects of the module and allow application of the material covered. All exercises are also available electronically.

INDICATIVE REFERENCES

Barnes DJ, Kolling M (2006) **Objects First with Java: A Practical Introduction using BlueJ, 3rd Ed.,** Prentice Hall

Bates B, Sierra K (2005) **Head First Java, 2nd Edition**, O'Reilly.

Deitel HM, Deitel PJ (2005) **Java: How to Program, 6th Edition**, Prentice Hall.

Charatan Q, Kans A (2005) **Java in 2 Semesters, 2nd Ed.,** McGraw-Hill.

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