Module 1: Computer Programming

<table>
<thead>
<tr>
<th>Stage</th>
<th>Semester</th>
<th>Module Title</th>
<th>Module Number/Reference</th>
<th>Module Status (Mandatory/Elective)</th>
<th>Module ECTS credit</th>
<th>Module NFQ level (only if applicable)</th>
<th>Pre-requisite Module Titles</th>
<th>Co-requisite Module Titles</th>
<th>Is this a capstone module? (Yes or No)</th>
<th>List of Module Teaching Personnel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>Computer Programming</td>
<td>1</td>
<td>Mandatory</td>
<td>10</td>
<td>8</td>
<td>None</td>
<td>None</td>
<td>No</td>
<td>Mr Eoin Carroll Mr John Hannon</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contact Hours</th>
<th>Non-contact Hours</th>
<th>Total Effort (Hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>Practical</td>
<td>Tutorial</td>
</tr>
<tr>
<td>24</td>
<td>36</td>
<td>60</td>
</tr>
</tbody>
</table>

Allocation of Marks (Within the Module)

<table>
<thead>
<tr>
<th>Percentage contribution</th>
<th>Continuous Assessment</th>
<th>Project</th>
<th>Practical</th>
<th>Final Examination</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Intended Module Learning Outcomes

On successful completion of this module learners will be able to:

1. solve programming problems of modest complexity in a systematic, well-organised way
2. specify precisely the syntax and semantics of a programming language construct
3. document accurately the design of a program on-the-fly
4. determine the basic efficiency of an algorithm
5. design and develop a range of standard algorithms

6. analyse and modify a program written by someone else

7. develop a program using an integrated development environment

Module Objectives

This module teaches the learner how to design high-quality programs in a systematic way. All the relevant concepts and techniques are explained and exemplified in the clearest, simplest language. The module aims to introduce the learner to the concepts of programming and problem solving.

Module Curriculum

Introduction to problem solving
- How do you complete a task?
- Identifying sub-components on larger task
- Defining order of subcomponents
- Creation of algorithms
- Stepwise design of programs

Introduction to programming
- Expressions and statements
- Basic arithmetic
- Comments
- Variables and assignment (Integers, doubles, booleans, characters)
- Boolean expressions and logic
- Conditional statements
- Iteration statements
- User input
- Output
- String manipulation
- Sub-routines (Parameters, Signature, Procedures, Copy rule.)

Professional Practice
- Developing a good coding style
- Using comments effectively
- Naming conventions
- Indentation
- Code structure

Reading lists and other learning materials

Secondary reading

McGettrick, A D. Graded Problems in Computer Science, 1983


Web Resources

http://docs.oracle.com/javase/tutorial/tutorialLearningPaths.html

Module Learning Environment

Accommodation
Lectures are carried out in classrooms / lecture halls in the College. Lab tutorials are carried out in computer labs throughout the Campus. All have the language software required to deliver the programme.

Library
All learners have access to an extensive range of physical and electronic (remotely accessible) library resources. The library monitors and updates its resources on an on-going basis, in line with the College’s Library Acquisition Policy. Lecturers update reading lists for this course on an annual basis as is the norm with all courses run by Griffith College.

Module Teaching and Learning Strategy

The module is delivered through a combination of lectures, tutorials and practical lab programming sessions. The learners complete a series of worksheets throughout the module, which build on the learning in lectures. The emphasis is on developing practical programming skills based on sound theoretical knowledge.

Module Assessment Strategy

The module assessment consists of a series of continuous assignments and a final examination.

<table>
<thead>
<tr>
<th>Element No</th>
<th>Weighting</th>
<th>Type</th>
<th>Description</th>
<th>Learning Outcome assessed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50%</td>
<td>Weekly Work Submission</td>
<td>A series of weekly worksheets covering the design and development of computer programs</td>
<td>1-7</td>
</tr>
<tr>
<td>2</td>
<td>50%</td>
<td>Closed Book Examination</td>
<td>End of module examination</td>
<td>1-6</td>
</tr>
</tbody>
</table>