School of Computer Science and Statistics
ECTS Module Descriptor

<table>
<thead>
<tr>
<th>Academic Year</th>
<th>2016-2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Code</td>
<td>CS1022</td>
</tr>
<tr>
<td>Module Title</td>
<td>Introduction to Computing II</td>
</tr>
<tr>
<td>Pre-requisites</td>
<td>CS1021 (Introduction to Computing I)</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Chief Examiner</td>
<td>Dr Adam Taylor</td>
</tr>
<tr>
<td>Teaching Staff</td>
<td>Dr Adam Taylor</td>
</tr>
</tbody>
</table>

### Delivery

<table>
<thead>
<tr>
<th>Lecture hours</th>
<th>Lab hours (per student)</th>
<th>Tutorial hours (per student)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>11</td>
<td>11</td>
<td>44</td>
</tr>
</tbody>
</table>

**Comments:** Attendance at all lectures, labs and tutorials is compulsory.

### Aims

This module continues directly from CS1021 (which is a prerequisite) and examines the structure and behaviour of computer systems in greater depth. In particular, this module introduces students to the implementation of simple data structures (stacks, multi-dimensional arrays, composite data types), subroutines, exceptions, interrupts and basic I/O at the machine level.

The relationship between high-level programming language constructs and their realisation as executed instructions is explored further by developing pseudo-code solutions to programming problems in the first instance, before translating those solutions into assembly language programs. Particular attention is paid to the implementation of subroutines, the system stack and parameter passing conventions.

Students are given opportunities throughout the module to reinforce their problem solving, programming and written communication skills by designing, implementing, documenting and testing solutions to programming problems of increasing complexity. Problem decomposition is strongly encouraged.

### Learning Outcomes

When students have successfully completed this module they should be able to:

- Describe the characteristics, structure and operation of a computer system, including the execution of subroutines and the interface between the processor and external devices;
- Translate between high-level programming language constructs, including fundamental data structures and subroutines, and their assembly language equivalents;
- Design, construct, document and test assembly language programs to solve small-scale problems of moderate complexity by decomposing the problems into smaller parts and implementing solutions consisting of one or more assembly language subroutines;
### Syllabus

Specific topics addressed in this module include:

- Stacks;
- Subroutines;
- Parameter passing conventions;
- Single- and multi-dimensional arrays;
- Floating-point number representation;
- Exceptions and interrupts;
- Basic I/O using memory-mapped peripherals.

### Assessment

Assessment is by written examination (70%) and continuous assessment (30%). Continuous assessment usually consists of a single assignment and four smaller lab exercises, although this is subject to change.

### Bibliography

There is no required text for this module. The following are suggested additional reading:


### Website

See mymodule.tcd.ie (Blackboard)