<table>
<thead>
<tr>
<th>Academic Year</th>
<th>2014-2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Code</td>
<td>CS1021</td>
</tr>
<tr>
<td>Module Title</td>
<td>Introduction to Computing I</td>
</tr>
<tr>
<td>Pre-requisites</td>
<td>None</td>
</tr>
<tr>
<td>ECTS</td>
<td>5</td>
</tr>
<tr>
<td>Chief Examiner</td>
<td>Dr Jonathan Dukes</td>
</tr>
<tr>
<td>Teaching Staff</td>
<td>Dr Jonathan Dukes</td>
</tr>
<tr>
<td>Delivery</td>
<td></td>
</tr>
<tr>
<td>Lecture hours</td>
<td>22</td>
</tr>
<tr>
<td>Lab hours (per student)</td>
<td>11</td>
</tr>
<tr>
<td>Tutorial hours (per student)</td>
<td>11</td>
</tr>
<tr>
<td>Total</td>
<td>44</td>
</tr>
</tbody>
</table>

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

**Aims**

This module provides students with an introduction to the basic structure and operation of a computer system, focussing on the processor (CPU) and memory.

Students gain an insight into the execution of programs on a computer system by designing, implementing and executing simple assembly language programs. Students are also introduced to concepts that are fundamental to the study of Computer Science, including the binary numeral system and the representation of basic information such as signed integers and strings (text).

Students are encouraged to consider the relationship between high-level programming language constructs – from simple assignments and arithmetic expressions to conditional (if) and iterative (while) execution – and the realisation of these constructs as sequences of machine instructions.

Students are also be given opportunities to develop their problem solving, programming and written communication skills by designing solutions to programming problems, implementing those solutions, first in the form of high-level pseudo-code programs and then as assembly language programs, which must be documented and tested.

**Learning Outcomes**

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

- Describe the basic characteristics, structure and operation of a computer system;
- Represent and interpret basic information (integers, text) in binary form;
- Translate between simple high-level programming language constructs and their assembly language equivalents;
- Design, construct, document and test small-scale assembly language programs to solve simple problems;
- Reason about the cost of executing instructions and the efficiency of simple programs;
### Syllabus

Specific topics addressed in this module include:

- Number systems, memory and data representation;
- Basic computer architecture (CPU, memory, registers, fetch-decode-execute loop);
- Assembly language and machine code;
- Binary arithmetic and bit-wise operations;
- Program flow control using branch instructions;
- Memory accesses (using load and store instructions).

### Assessment

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

Supplemental assessment is by 100% examination.

### Bibliography

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


### Website

See [mymodule.tcd.ie](http://mymodule.tcd.ie) (Blackboard)