<table>
<thead>
<tr>
<th>Module Code</th>
<th>CS2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Name</td>
<td>Systems Programming I</td>
</tr>
<tr>
<td>Module Short Title</td>
<td>Systems Programming I</td>
</tr>
<tr>
<td>ECTS weighting</td>
<td>5</td>
</tr>
<tr>
<td>Semester/term taught</td>
<td>Michaelmas Term</td>
</tr>
<tr>
<td>Contact Hours</td>
<td>22 lectures, 11 labs. Total 33 hours</td>
</tr>
<tr>
<td>Module Personnel</td>
<td>Dr David Gregg</td>
</tr>
</tbody>
</table>

### Learning Outcomes
When students have successfully completed this module they should be able to:
- design and construct clear, maintainable, correctly structured programs in C to solve a variety of practical problems;
- develop programs using UNIX development tools such as vi, regular expressions, gcc, gdb and make;
- appraise the quality of programming style in existing programs;
- distinguish between good, questionable and bad programming style;
- assess alternative program design decisions;

### Module Learning Aims
Students taking this module have already successfully completed courses in object-oriented Java programming and ARM assembly language programming. This module starts with a new perspective on program construction, with structured programming in C. This part of the module deals with the features of C, sound design principles for structured programming, and the necessary self-discipline required to program in a low-level language like C. Students also learn the basics of how programs are commonly implemented, with special emphasis on the layout of program data in memory. In parallel to learning C programming, the students learn the basics of the Unix Operating System.

### Module Content
- The C programming language
- Pointers and dynamic memory allocation/de-allocation
- Principles of structured programming
- Abstract data types in structured programming
- The UNIX operating system
### Module Descriptor 2013/14
School of Computer Science and Statistics.

**Recommended Reading List**

* UNIX programming tools  
* Programming style appreciation and criticism

**Recommended Reading List**

* The C Programming Language, Brian Kernighan and Dennis Ritchie  
* The Practice of Programming, Brian Kernighan and Rob Pike  
* Programming Pearls, Jon Bentley

**Module Pre Requisite**

CS1011, CS1012, CS1021, CS1022

**Module Co Requisite**

CS2011

**Assessment Details**

Assessment is by written examination (contributing 80% to the overall mark) and continuous assessment (contributing 20% to the overall mark). To pass the module, students must achieve an overall mark of 40%.

The course consists of a mixture of lectures and laboratory work. The course is fundamentally practical in outlook. Students learn solid engineering design principles of using simplicity and abstraction to solve practical programming problems. During lectures programs are constructed on the board, through a combination of student participation and prompting from the lecturer. Students are strongly encouraged to evaluate and criticise alternative design decisions, to learn to distinguish between simple, elegant solutions, and complex, difficult and error-prone ones.

In laboratory sessions students put principles into practice in the weekly programming assignment using C and UNIX. Students design, implement and test their solutions using UNIX development tools.

The core of this course is the weekly lab assignments. The students complete a small programming assignment each week. This assignment is started in the weekly lab, and completed in the student’s own time. Around half of these programming assignments are submitted for grading. Students are required to demonstrate and explain the internals of their program as part of the marking process, in order to ensure that the work is their own. The results of these 5-6 assignments are totalled and count for 20% of the total mark for the year. The remaining 80% of the marks for the course are for the annual exam.

In the supplemental examinations, assessment is by written examination only, which contributes 100% of the overall mark.

**Module approval date**

N/A

**Approved By**

N/A
<table>
<thead>
<tr>
<th>Academic Start Year</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Academic Year of Data</td>
<td>N/A</td>
</tr>
</tbody>
</table>