Teaching programming
M Rovatsos, Teaching Committee, 12th March 2014

Background
Various discussions over at least this and the previous year with staff and students indicate that our provision of teaching programming falls short of student, staff, and employer expectations. Even though there is no hard, conclusive evidence, feedback from all these groups suggests that not all Informatics graduates have sufficient programming skills, that many of them lack the level of skill required by top employers in the software industry, and that many students feel they are insufficiently prepared for and guided through the process of acquiring programming skills through our courses.

Purpose
The purpose of this item is to gather staff opinion on possible future modifications to our programming curriculum, based on an initial list of objectives and possible interventions. These lists are by no means exclusive, and should only be taken as a starting point for discussion. A positive outcome would be to refine/extend these lists and get an initial feel for interventions that seem worth developing further.

(a) List of relevant objectives to guide measures
The following is a list of possible objectives for programming teaching, does not imply the School pursues or necessarily agrees on all of these objectives, but they are possible goals that would help set our priorities.

<table>
<thead>
<tr>
<th>Objective Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>Every Informatics graduate has reasonable programming abilities</td>
</tr>
<tr>
<td>O2</td>
<td>Every student is able to build working programmes from scratch by end of Y2</td>
</tr>
<tr>
<td>O3</td>
<td>All CS and SE students have advanced programming skills</td>
</tr>
<tr>
<td>O4</td>
<td>Any interested student has opportunities to become an expert programmer</td>
</tr>
<tr>
<td>O5</td>
<td>Every student has working experience with several programming paradigms</td>
</tr>
<tr>
<td>O6</td>
<td>Every student has experience with languages widely used in their subject area</td>
</tr>
<tr>
<td>O7</td>
<td>Every student has skills to pick up new programming languages with ease</td>
</tr>
<tr>
<td>O8</td>
<td>Programming teaching addresses algorithmic problem solving</td>
</tr>
<tr>
<td>O9</td>
<td>Programming teaching addresses “soft” programming skills</td>
</tr>
<tr>
<td>O10</td>
<td>Programming teaching provision is aligned to the skills of different students</td>
</tr>
</tbody>
</table>

From discussions with students and staff, it would seem that we currently achieve O4, O5, O7, but that we fall short in some of the others. O1 and O2 seem particularly critical both for our overall graduate profile and for students to do well on our courses.

(b) Possible interventions
With lists of objectives they support (and, in brackets, those they potentially support), and effort/resource rating on a $ to $$$ scale.

1. Review teaching of Inf1-OP (lab-based/online delivery, syllabus, streaming)
   O1, (O2, O10) $

2. Introduce “Inf1-OP++” for students with programming background
   O10, (O3, O4) $$

3. Pre-sessional courses for UG/PG students without programming background
   O10, (O2) $$$

4. Strengthen programming element of the Inf2C-SE curriculum
   O3, O9, (O6) $

5. Introduce a Y1 C/C++ course for Informatics students
   O5, O6, (O3, O4, O1, O7) $$

6. Introduce a portfolio-based “Improving Programming Skills” Y4 course
   O1, O10, (O4, O3, O8, O9) $$
7. Introduce an optional “Advanced Programming” Y3/Y4 course
   \(\text{08, 09, (03, 04) $$$}\)

8. Introduce a “Programming Skills” Y3 course compulsory for all students
   \(\text{01, (03, 04, 08, 09, 07) $$$}\)

9. Augment Individual Practicals by more programming teaching elements
   \(\text{01, (04, 010) $}\)

10. Offer (peer-assisted?) extra-curricular programming workshops at all levels
    \(\text{01, 04, 010, (08, 09) $$$}\)