Summary of Inf2C-SE

The major change in Inf2C-SE this year was the increase in coursework weight from 25 to 40% and the coursework itself. Coursework in earlier years required students to pick a project from sourceforge.net and make a contribution of their choice (can be fixing bugs, reporting bugs, develop or improve user documentation, add tests, among others).

Description of the coursework offered this year is as follows,

- **Coursework 1** (overall weight 8%) - was focussed on creating a requirements document and use cases for a Cruise Control System (CCS). Templates to write use-cases, and a guide to specifying requirements were provided. The Eclipse Papyrus tool was recommended for drawing use case diagrams. Markers provided detailed feedback of any changes necessary in the requirements/use cases.

- **Coursework 2** (overall weight 12%) - was focussed on object-oriented design of the CCS introduced in coursework 1. The coursework required a detailed design document complete with UML class and sequence diagrams based on the requirements and use cases from coursework 1. Students also had to show how they addressed the coursework 1 feedback from the markers. The students used the Eclipse Papyrus tool or the online drawing tool, draw.io to draw UML class and sequence diagrams. Markers provided detailed feedback of any changes necessary in the design.

- **Coursework 3** (overall weight 20%) - required the students to implement the CCS design from coursework 2 in Java. Implementation had to adhere to coding standards that included descriptive JavaDoc comments. Support code for the system interfaces was provided. The students used Eclipse IDE for their implementation and test. Students had to write requirements-based tests in the form of JUnit tests to test requirements from coursework 1. The students were also required to measure the statement coverage achieved with the tests. A minimum of 80% coverage was required. Coverage was measured with the EclEmma plug-in Eclipse. The students were required to provide a test report (which tests passed and which failed), a coverage report generated with EclEmma and finally a traceability matrix to show which test cases verify which requirements. Version control using SVN was encouraged.

The coursework gave the students a flavour of the major software engineering activities, the discipline involved and hands-on experience with tools for achieving them. The coursework was also meant for students to understand how the activities feed into each other and how mistakes early in the process are more expensive to correct later on. The importance of addressing feedback was also stressed in the coursework. Majority of the students scored well in the coursework.

**Coursework in Groups of 2**

Software development in the industry is almost always performed in groups. In earlier years, coursework was done individually. This year, we encouraged students to form teams of 2 for the coursework. We only had a couple of students who chose to do the coursework individually. Everyone else chose to do the coursework in groups of 2 and we did not encounter any issues with team work.

**Overall impression**

I got the impression that the students found the coursework to be very useful in understanding the major software engineering activities - requirements, design, implementation and test. The Cruise Control System itself was fairly straightforward since the focus was on the different activities. I am pleased with the performance of the students in the coursework, and the outcome of increased coursework weight. I would like to continue to stick to this year's format for the coming years.