

# COSC2511 Introduction to Programming

## Assignment, Semester 1, 2017 STAGE 2

A final product **or** continued prototype development can be submitted (through Grade centre).

The reason for continued development is that coding projects often need to be extended. Finishing on time with an inferior product is not recommended.

This is also why you need to develop technical confidence in being upfront with issues, often to justify why something has occurred, and further, a strategy for solving it.

A component marking scheme similar to the first stage is being used.

The previous stage marking is compared with the current marking. You **do not** have a marking category marked twice in both stages. However, if there is reasonable differences in modification and improvement, where for example a feature is improved, another marking is possible. Ideas and code development is encouraged and rewarded in this marking scheme.

Your stage 1 code is compared directly with your stage 2 code.

The assessment is a difference between your stage 1 and stage 2 submissions.

The exception to component markings requiring unique status is 'Program logic 2' and 'Functions and methods 2' where the same categories appear in stage 1. This takes a holistic view of the code and compares with the previous code version. However, if there is no improvement from the previous code submission, or no further development, *no marks will be awarded*. So it is possible to have a logical program that receives no marks in stage 2 for logic as the code has not been improved (this has happened). However, you can obtain marks through implementing features which in theory is unlimited.

The projects code should be an involved consideration of the problem in more depth than your class exercises. Stage 2 is a continuation of the stage 1 project.

Stage 2 due Week 13- Week 18

Requires:

- Presentation/interview (in-class or pre-arranged)
- communication of solo or group work and members
- competed in-class interview (needs to be arranged)
- 1 page report (*demonstrated in interview*)
- Original data (the problem should have its own associated data)
- working Java code (*demonstrate in interview*)
- + optionally other documents

All documents submitted through gradecentre before the interview. If non-attendance of interview, no marks for the report stage as no verification of work.

Approximate marking guide, though adding components increases the mark. Factors: demonstration of technical skill (code); organisation of data and problem specification.

Dates: Week 18 starts 12th-June

Resources: the class examples provide many coding resources

If group work, all members share the same mark.

Different projects will have different marking components and common marking components. While the nature of marking is subjective, by considering marking components of the project, a tally of the components (compared with stage 1) is used to determine the mark.

A maximum of 100 marks is possible.

| Marking Criteria   | Marks allocated     |
|--|---------------------|
| Problem communication (technical – support what you claim with reason), include honesty such as other references, technical and design issues. | 10                  |
| Javadoc documentation (generally single line comments in all code attributes, functions and classes)   | 10                  |
| Menus and user interaction   | 15                  |
| Storing test data in the application   | 10                  |
| Program logic 2  | 20                  |
| Non-trivial data structures (e.g. classes, arrays)   | 10                  |
| Classes, attributes, functions/methods 2 (not monolithic or badly coded)   | 20                  |
| Minor feature  | +2.5                |
| Moderate feature   | +5                  |
| Major feature  | +10                 |
| Testing  | Variable 2.5-5.0    |
| Documents (some projects can be assessed   | Variable on project |

No double marking: reminder of two kinds of section marks:

fixed sections and component sections - doing the same pattern (repeating the same idea) is not considered as separate component - no double marking.

- additional features are separate coding features and hence not subject to double marking
- you are rewarded for additional coding features

No double marking: if get marks in one component, the marks are not awarded in another component.