Computer Science D01
University of Toronto Scarborough

Project Deliverable #1

---

1 Learning the code base and planning your work

In this part of the project you will familiarise yourself with the existing codebase and decide on a software development process you will use for your next deliverable.

- Check it out of the repository, configure, and run.
- Read through the documentation, learn to use it.
- Investigate available tools that will help you with the assignment. Among others, you will need a UML drawing tool (simple lightweight tools will work fine for this), and likely a reverse engineering tool.
- Browse through the repository, familiarise yourself with the structure.
- Use a reverse engineering tool to generate a UML class diagram representing the classes/modules, subclass relationships, and associations in the source code. You will need to edit the generated model to capture information missed by the tool, to remove unnecessary detail, etc.
- Draw a higher-level diagram to show the overall architecture of the system. Use any appropriate UML notation (e.g. packages, components, interfaces, etc). Be sure to show clearly where the classes belong in this architecture, and what external packages the system interacts with. As always, the exact UML notation that you use is much less important than the modelling decisions you make: what is important enough to include, what to omit, and how to structure your diagrams.
- Decide on the software development process you will use for your next deliverable (you will need to fix three bugs for your project). Make sure you weigh the pros and cons of each process and explain your choices clearly.

  You can choose/set up any software development process we have covered in class **except** for what you did in your CSCC01 project (a more or less modified Scrum process).

2 What to submit?

- A commentary on the architecture of the system, highlighting any interesting aspects of the design (e.g. architectural style, degree of coupling, etc), discussing the quality of the architecture used in the system, and suggesting possible improvements where appropriate. Use UML diagrams to illustrate the points you wish to make.
- Any other UML models you generated, as appropriate. Be sure to include at least enough views so that you show how everything fits into the overall system structure.
- A thorough explanation of the software development process you will follow for your next deliverable, including the reasons for your choice and any modifications you made to customise it for your team.

2.1 Marking

- Overall architecture
  - It is clear to the reader that the team correctly understands the architecture of the project. (5 marks)
  - A reader, initially unfamiliar with the architecture of the project, has a good overview of the project after reading the report. (5 marks)
  - UML diagrams are correct, clear, and helpful. (5 marks)
- Software Development Process
  - The process is described clearly and thoroughly, all aspects are mentioned. (7 marks)
  - The arguments for the choice of the process and any modifications are well-informed and well explained. The pros and cons of different processes are carefully considered. (8 marks)
- Presentation and Quality of Writing (10 marks)
  It is up to you to choose the format of all write-ups in the course: you can produce either `pdf` documents or collections of `html` files. Whatever format you choose, make sure it looks professional and is very easy to read: the TAs will not have much time and you need to convince them you did an excellent job!
  The following will always be considered when grading your work in this course:
  - Presentation: the report is well formatted, easy to read, and easy to navigate.
  - Quality of writing: language, grammar, clarity, professionalism.