Computer Science D01
University of Toronto Scarborough

Project Deliverable #4

---

1 What you will do

For this deliverable, you will design and plan the implementation and testing of a new feature, or a significant bug fix, for `matplotlib`. As with the previous deliverable, this requires you to use your judgment about which feature to select for implementation. There is no "correct" choice: you will be given credit for selecting a feature that can be fully implemented, documented, and tested in the time available.

2 What to submit/present and when

1. Examine the list of requested features/serious bug fixes in `matplotlib` issue list in GitHub. From the issue list, select at least two features/bug fixes to examine further.

   These should not be small fixes: we are looking for **substantial new development** here. The best way to ensure that you are working on acceptable features/fixes it to contact your TA very early for a confirmation. Some guidelines (not exhaustive, of course, and not necessarily sufficient) for selecting what to work on:

   - Label `Difficulty: Hard`, or, possibly `Difficulty: Medium`,
   - Not `Documentation`,
   - Label `API changes`,
   - Issues on improving performance (requested by `matplotlib` developers from our course — issue 6952, for example)

   Your TAs will also make sure that **no two teams work on the same issue**, so the sooner you contact your TA, the better!

2. Briefly and clearly describe the selected features/bug fixes in your report.

3. For each of the selected features/bug fixes, investigate what parts of the existing code base you would need to modify in order to implement the feature. Produce designs for the selected features, describing your plans for the organization of new code, as well as all interactions between new code and existing code. You guessed it: UML diagrams would be very helpful here.

4. Select one of these features/bug fixes for implementation, and briefly explain your decision.

5. For the selected feature/bug fix, produce a suite of acceptance tests that will demonstrate that the feature/bug fix has been implemented correctly. Design these as "customer acceptance" tests: i.e. a description of the steps a user needs to carry out to check that the program works as expected.

6. For the selected feature/bug fix, produce a unit test suite that will demonstrate that the feature/bug fix has been implemented correctly. See mpl guidelines for writing tests.

2.1 Marking

- Issue descriptions, high-level. (6 marks)
  - Issue 1. (3 marks)
  - Issue 2. (3 marks)
- Detailed issue descriptions, including UML diagrams, code traces, implementation plans. (14 marks)
  - Issue 1. (7 marks)
  - Issue 2. (7 marks)
- Acceptance tests for the selected issue. (5 marks)
- Unit tests for the selected issue. (5 marks)
- Presentation and Quality of Writing (5 marks)
  Same guidelines and expectations as before.