159251 Software Design and Construction 2018
Assignment 1

August 15, 2018

Deadline and penalties

You must submit your final work using the stream submission system no later than [see deadline on stream]. The penalty is 10% deducted from the total possible mark for every day delay in submission (one day late — out of 90%, two days late 80% ...).
You are required to use a private repository for this assignment. GitLab and Bitbucket are services that offer private repositories. GitHub private repositories are available with their free student developer pack.

Contribution to Final Grade

19%

Overview

You are to work in self-selected pairs to create the scripts/program defined below, using Git to manage source code contribution and integration between the two developers. All project issues and changes should be tracked using an issue tracker.
Both members of the group will receive the same mark unless it is clear that the work is predominantly that of a single person. These will be sorted out on a case-by-case basis. The breakdown of the work is entirely up to you and your project partner. Part of this assignment is to become familiar with using git for version control. You can either use Github (https://github.com) or Bitbucket (http://www.bitbucket.com) – both provide a free git hosting service. However, Bitbucket provides free private repositories. For Github, you can request an academic licence, which will allow you to create private repositories. Make sure that your repository is private in any case. Both tools come with default issue tracking features. IMPORTANT: Choose meaningful names for your repository (e.g. 251-A1-Smith-Jones)

How to submit

All submission is to be done using Stream:
Share your program on your private git repository (GitHub or Bitbucket) with us by sending a share invitation (READ-ONLY access is best) to S.Rasheed@massey.ac.nz and
jacobstringer@windowslive.com
This is to track commits on your git repository.
Include a `README.md` file in the top level of the project.
The `README.md` is a text file in Markdown format that contains: 1. the names and IDs of BOTH MEMBERS of the group.
2. If there are any other folders, what they contain.
3. for each student, a couple of git commit IDs that show the work of each individual member of the group.
4. any other features you feel worth mentioning.

Who submits what?

Only one member of a group should submit a complete project, the other just submits the `README.md` file: member A: submit (through Stream) a single compressed (e.g., zip or tar) file that contain the assignment and name the compressed file with both members' FirstName_LastName and ID numbers (e.g. James_Smith-87817172- Susan_Jones 01002023.zip) member B : submit just the `README.md` file containing your name and that of the partner who is submitting the zip/tar file. This is so Stream knows that you've submitted something, otherwise it won't let a mark be entered.

Tasks

1. Shell and scripting

A team of developers manage todo lists in their local directories as markdown files. Each markdown file represents a project and they contain a list of todos. Markdown format:

```
# Project title
## Tasks
[ ] (A) Todo 1 due:2010-01-23
[ ] (C) Todo 2 due:2010-01-21
[X] (B) Todo 3 due:2010-01-10
```

Write a script that:
1. Checks that the project title matches the file name, moves files that don't match to a directory named `./error_files` (8/45)
2. Copies todos without a due date to directory named `./no_duedates` (5/45)
3. Combines all project todos (todos without errors) in a single file (2/45)

2. Software development

You are provided with the source code for a basic web application that can be used to manage todo lists. The application uses Apache Wicket (a Java web framework.) You are required to implement additional functionality:
1. Clear completed tasks from the list (2/45)
2. A filter to show either completed, active or all tasks (5/45)
3. Show the total number of incomplete tasks (5/45)
4. Add a due date, project title field in the add task form (3/45)
5. Add persistence (e.g. XML-based serialisation, SQLite database) support for the application (cleared tasks must be archived) (8/45)
6. Import the tasks from the todo list text file into your application (5/45)
7. Add an indicator in the webpage so that it is easy to distinguish the environment it is running in (development or production) (2/45)

Development environment

You will need an installation of Apache Maven for building and executing the web application, and Git for version control on your computer. More instructions are included in the `README.md` file for the application.

Assessment

Your assessment will be based on the following criteria

- Correct implementation of the functions - 45%
- Use of issue tracking from the start for tracking issues/changes and features - 25%
- Appropriate use of Git - 25%
- Overall code quality (comments, error handling) - 5%

Plagiarism

We will check submissions for plagiarism. Please read the Massey guidelines on plagiarism and dishonesty for details: https://goo.gl/S3tn18 .