Marking Rubric for Problem Solving Tasks (PSTs) - Programming Concepts (Python)

July 22, 2025

FIT1056 - Introduction to Software Engineering

Important Assessment Guidelines for PSTs

- Completing only the **minimum requirements** of the PSTs will result in a **pass-level mark only**. Higher marks require students to show **initiative, completeness, correctness**, and high-quality work.
- Students must **demonstrate their understanding** of the submitted PSTs during the in-class applied session. Tutors may ask questions or request code modifications to verify understanding.
- If a student **fails to attend the session** or is **unable to explain their work**, they will receive **zero marks**, even if the submission is complete and correct.
- **Demonstration is a critical part of PST assessment**. Attend on time, be prepared, and engage actively in discussions.

1 Code Functionality and Quality (95%)

- **Working code with appropriate commenting (90-100%)**
  - The program runs flawlessly.
  - All functionalities, as per the requirements, are implemented and work correctly.
  - Code is accompanied by insightful and necessary comments, explaining complex or crucial parts of the code.
- **Minor errors but majority functionality is correct (70-89%)**
  - The program runs and delivers most of the functionalities as per the requirements.
  - Minor logical errors or small sections of missing functionality.
  - Code has useful comments, but might lack in certain sections or could be clearer.
- **Significant errors with partial functionality (50-69%)**
  - The program runs but encounters significant errors or does not deliver a substantial portion of the required functionalities.
  - Some parts of the code may be commented well, but crucial sections lack appropriate commenting or clarity.

- **Code runs with very limited functionality (30-49%)**
  - The program runs but offers only a limited part of the required functionalities.
  - Commenting is sparse and does not aid much in understanding the code's workings.
- **Code does not compile but structure is visible (10-29%)**
  - The program does not compile due to errors.
  - There is a discernible attempt to structure the code as per the requirements.
  - Some sections might have comments, but they're not consistent or particularly helpful.
- **Code does not compile, no clear structure (0-9%)**
  - The program does not compile and lacks a clear structure or approach towards the requirements.
  - Commenting is either missing or not helpful in understanding the code.

2 GitHub Submission (5%)

- All PSTs must be **committed and pushed to GitHub** as instructed. The deadline for committing and pushing is the same as the Moodle deadline. Delayed commits and pushes will incur late penalties.
- Failure to submit via GitHub (in addition to submitting your work to Moodle) will result in a loss of this **5%** component.

3 Submission and Demonstration

- **No demonstration (0%)**: The student fails to demonstrate their work during the session or as required.
- **No submission (0%)**: The student fails to submit their PST work within the stipulated deadline or as required.
- **Demonstration Understanding**: Tutors will ask students to demonstrate their understanding of the code or answers they have provided. They may sometimes ask students to make minor tweaks and modify the code accordingly to further test understanding. Failing to effectively demonstrate understanding will result in 0 marks for the entire work.

4 Penalties

- **Late Submission**: A late penalty will be applied at 5% of the total mark per day. Any submissions after 7 days from the deadline will not be considered.