Tasks 2020

Emerging Technologies

Due: last commit on or before December 18th, 2020

This document contains the instructions for the tasks for Emerging Technologies in the first semester of the 2020/2021 academic year. This assessment is worth 50% of the total marks for the module. Students should carefully read the document *Using git for assessments* [2] that is available on the Moodle page and applies to this assessment. Students are also reminded to follow the GMIT Quality Assurance Framework including the Code of Student Conduct and the policy on plagiarism [1].

Instructions

You should complete the following four tasks in one jupyter notebook in a single git repository. This repository should be submitted using the correct link on the Moodle page. The tasks should each ideally be completed within two week of them being made available. It is important that you continuously work on the repository throughout the semester.

1. **October 5th, 2020:** Write a Python function called `sqrt2` that prints to the screen the square root of 2 to 100 decimal places. Your code should not depend on any other module, in the standard library or otherwise. You should research the task before attempting it, and include references and a description of your algorithm in a Markdown box above the code.

Marking scheme

The following marking scheme will be used to mark your submission out of 100%, which will then be scaled to 50%. The examiner's overall impression of the assignment may influence marks in each individual component. Also, it is important that your submission provides direct evidence of each of the items listed in each category. For instance, your commit history should demonstrate and provide evidence that you had a pragmatic attitude to completing the assessment. Likewise, your submission should have references in it to demonstrate that you considered the literature and the work of others.

25%	**Research**	Evidence of research performed on topic; submission based on referenced literature, particularly academic literature; evidence of understanding of the documentation for any software or libraries used.
25%	**Development**	Environment can be set up as described; code works without tweaking and as described; code is efficient, clean, and clear; evidence of consideration of standards and conventions appropriate to code of this kind.
25%	**Consistency**	Evidence of planning and project management; pragmatic attitude to work as evidenced by well-considered commit history; commits are of a reasonable size; evidence of consideration.
25%	**Documentation**	Clear documentation of: how to create an environment in which any code will run, how to prepare the code for running, how to run the code including setting any options or flags, what to expect upon running the code. Concise descriptions of any code in comments and README.

References

[1] GMIT. Quality assurance framework.
`https://www.gmit.ie/general/quality-assurance-framework`.

[2] Ian McLoughlin. Using git for assessments.
`https://github.com/ianmcloughlin/using-git-for-assessments/`.