Maintainable Code Reflection Portfolio

Deadline: 5 Nov, 3pm

Synopsis:

Part of understanding how to improve as a software developer is learning how to reflect on new concepts and on your coding performance. In this coursework you will be asked to write about several aspects of the practical sessions of this course. Each week you will receive a new task which is related to either what you have implemented in the lab sessions or what you have understood from the guest lectures.

You must choose FOUR out of the six tasks outlined for you. You must answer the question using examples of what you have achieved in the practical sessions or from your own personal study of the related course material. Each task will contain a question for you to answer.

The aim of this coursework is to ensure you have understood the principles involved in creating and developing maintainable software. To get started, read the following blog article on writing good, clean, clear and understandable code.

https://www.javaworld.com/article/2075129/testing-debugging/make-bad-code-good.html

Your answers to the tasks should be written in the style of a blog article. Please feel free to enhance your answers with snippets of commented code. Each task should be completed in no more than 250 words maximum.

Deliverables:

You must write up your answers to your FOUR out of   ng. We will not accept submissions which answer all six (our first four tasks only)   will have an associated word count which you must also not exceed, which is 250   ns and supporting code snippets.

Your submission will be a PDF file submitted through Moodle on or before the given deadline. This single file must contain your answers to the FOUR tasks. We will not accept submissions containing multiple files. Check nearer the submission date for the live link to the Moodle submission system. Furthermore, we might refuse to continue reading answers to individual questions beyond 250 words (although we will not stop in the middle of a sentence; but try to stick to the limit). Please add the overall number of words you wrote for all FOUR tasks to the front page.

To mark your work, we may wish to print off a copy, therefore please add your full name and student ID to each numbered page of your completed portfolio. Also please use your surname, first name, and student ID as the filename for your submission, e.g Greensmith_Julie_40135203.pdf. Failure to adhere to these rules may result in our refusal to mark your coursework.

Note:

You are gently reminded that we are at liberty to use plagiarism detection software on your submission. Plagiarism will not be tolerated, and academic offences will be dealt with in accordance with University policy and as detailed in the student handbook. This means you may informally discuss the coursework with other students but your submission must be your own work. Please also note that it is not permitted for you to copy and paste text from another source without correct referencing. If you are unclear about this process please discuss with the module convenors during one of our lab sessions or at the end of a teaching session.

Assessment Details:

This coursework is worth 25% of this module and as such is marked out of 25. Each task is worth a maximum of 5 marks. A further 5 marks will be awarded for the quality of the writing and clear communication skills. Each of the five marks refers to a degree classification as outlined in Table 1

Mark	Classification	Marking Schema
0	0	No task was submitted, is missing, or is nonsensical
1	Fail	Poor answering of the question, no reflective content, incorrect code snippets or did not answer the question. Also, very poor presentation.
2	3rd Class	Basic completion of the task with partially correct evidence but no linkage with lab tasks or guest lectures. Also, adequate presentation but with inconsistencies and with some errors.
3	Lower Second Class	Good completion of the task with correct evidence but some problems either with commenting, minor inconsistencies, an attempt at reflective content, basic linkage with lab tasks or guest lectures. Also good presentation, mostly correct and consistent with some minor errors.
4	Upper Second Class	Very good completion of the task complete with a strong link to the lab or guest lecture content. Examples provided and correct and clear he task provided. Also, great clear use of code, style and
5	First Class	Excellent completion o lecture content, with e classes, labs and gues flawless standard of pr nkage with lab or guest d those described in the eferenced sources. Also xternally published.

Table 1: Marking Classification (Rubric) for Tasks and Presentation

Tasks are detailed below, of which you must complete FOUR out of six, each task to be completed in a maximum of 250 words (excluding diagrams, illustration and supporting code snippet examples). You also have to highlight 5-6 keywords/phrases that you think highlight the idea behind your answer.

Task 1: Interfaces

In relation to your implementation of the ZooApp system in Lab 2, describe the term interfaces and information hiding in this context and reflect on how an interface relates to packages, libraries and APIs, giving example code snippets from your completed lab assignment.

Task 2: Object Oriented Design Group Task

In relation to your completed work on the group task in Lab 3, reflect on your considerations for how to design software with lower maintenance effort in the future giving examples from this object oriented design lab class and relating your reflection to your understanding of building maintainable code.

Task 3: Regression Testing

In relation to your completion of the testing process tasks in Lab 4, describe the process or regression testing and discuss the differences between debugging and testing. Illustrate your answer with a discussion of the regression testing process you trialled in the lab and discuss the importance of regression testing when updating legacy versions of maintained software.

Task 4: Branching and Merging in Source Control

In relation to your knowledge acquired of practical version control using Git based systems in Lab 5, reflect on how you will utilise source control in your future work at the University and beyond, including how this would apply to your second year group project. Emphasise how you would manage the process of software development in a team using source control and use command line code samples to substantiate your answer.

Task 5: Reflections on Dr. Sameh Zackary's Guest Lecture (University of Nottingham)

Attend the guest lecture* given by Dr. Sameh Zackary which describes the software maintenance process in research grade software maintained over a legacy period. Write a reflection on how the central messages of this guest lecture can apply to your own ability to write maintainable code and how this can be applied to your second year group project.

Task 6: Reflections on Yas Poptani's Guest Lecture (Coventry Building Society)

Attend the guest lecture* given by Yas Poptani and colleagues which describes the industrial software maintenance process an   ce. Write a reflection on how the central messages of   maintainable code and how this can be applied to yo

*please note that guest lectures will not be recorded, so it   se, if you want to write about them.

Questions?

Dr. Greensmith and Dr. Siebers will of course be happy to answer questions and give high level interim formative feedback on your assignment during the lab sessions.

To help yourself to complete this coursework it is highly recommended that you do the lab worksheets during the lab session where you can make use of course leaders and demonstrators for feedback on your work. Ideally you should complete one task per week to avoid 'coursework congestion' near the deadline - this is a much more mature way of completing assignments. Leave it to the last minute at your peril as some of the lab work is very difficult to complete outside of the assigned teaching sessions!

Good luck, have fun, code well.