

School of Computer Science – Coursework Issue Sheet

Session	17-18	Semester	Autumn
Module Name	Software Maintenance	Code	G52SWM
Module Convenor(s) (CW Convenor in Bold)	Andrew French, Peer-Olaf Siebers		

Coursework Name	Understanding and improving other people's software	Weight	25%
Deliverable (a brief description of what is to be handed-in; e.g. 'software', 'report', 'presentation', etc.)	1000 word report reflecting on the process of understanding third party software including relevant examples, class diagrams and explanations.		
Format (summary of the technical format of deliverable, e.g. "C source code as zip file", "pdf file, 2000 word max", "ppt file, 10 slides max", etc.)	PDF file, 1000 words max . Word count on front page. This should be in the form of a number of screenshots of examples of poor coding practice (four examples) along with a brief description of any coding problem, and how you would improve it. In addition you should provide one example of good coding practice, by using a screenshot and a brief justification why it is an example of good coding practice.		

Issue Date	w/b 16 Oct
Submission Date	8 Nov, 3pm
Submission Mechanism	Electronic submission of PDF via Moodle
Late Policy (University of Nottingham default will apply, if blank)	Standard policy
Feedback Date	w/b 20 Nov
Feedback Mechanism	Individual comments via grading system on Moodle. General feedback in lecture.

Instructions	<p>You will be given the source code of a Breakout game, which uses the JavaFX Game Library (FXGL) as a basis. We have introduced some poor programming practice to this code - Use the given code for this coursework. Your task is to find and read some material (we will give you an article to start with linked below, but you should find some more material by yourself online) on the topic of "Making Bad Code Good Code" and then apply what you learned from your studies to portions of the given code. Don't forget to reference articles.</p> <p>Note: The code will not compile or run (yet). You do not need to make the code run in this coursework.</p> <p>The expected output of the coursework is a report (1000 words max.) where you describe how you would improve the provided code, based on what you have learnt about good programming practices and object oriented software development to date. This should be in the form of a number of screenshots of problem code (covering four significantly different types of problems) along</p>
---------------------	---

with a brief description of any coding problem, and how you would change it to overcome these problems. In addition you should provide **one substantial** example of good coding practice, by using a screenshot and a brief justification why it is an example of good coding practice.

Some tips:

- You should start with reverse engineering a high level class diagram from parts of the source code provided. Doing this and **providing an explanation** of how you derived it from the source code (e.g. how you identified the type of relationship between classes) will have a positive impact on your mark. Only providing a class diagram without explanations of how you derived it is not sufficient. The class diagram only needs to be high level – i.e. providing class names and relationship but not listing all fields and methods.
- When you pick examples for uncovering bad coding practice you always need to provide an explanation of why your alteration is an improvement to the project. Just providing examples is not enough. Picking the same *kind* of examples will not bring you any benefits. You need to look out for different kinds of examples (tackling different types of problems with the source code)

You should start by reading the article "Make Bad Code Good" (see link below). If you want to have a better grade you should also consider some ideas from additional sources (on the internet).

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

IMPORTANT: Make sure you understand what you are writing in your essay. We reserve the right to briefly interview you if we think that you do not understand what you write about. So, when you write your essay do not simply copy/paste large portions of text from existing articles or other resources – as this does not demonstrate your understanding of the topic. You might also run into issues with plagiarism.

If you use information from other resources please remember to **provide a reference** to the website / book you found the information in. A quick guide for correct referencing can be found here:

<https://www.uhi.ac.uk/en/t4-media/one-web/university/library/how-to/UHI-mini-Student-referencing-guide-en-N A.PDF>

A link to the game code is available on the Moodle website.

Assessment Criteria	<p>Quality and challenge of the code improvements will be assessed. Evidence of further reading and investigation of the code will be particularly rewarded.</p> <p>MARKING:</p> <p>A pass mark requires:</p> <ul style="list-style-type: none"> • Evidence that you read the supplied article and that you have an understanding of some elements of good and bad coding practice <p>A high mark requires</p> <ul style="list-style-type: none"> • Evidence that you read and understood the article, and have done some further reading (and referenced it) • Provision of a wide range of insightful examples from the supplied code, related to recognised concepts, and suggesting improvements • Provision of a high level class diagram (including an explanation of how it was derived) • Demonstration of an understanding of the more complex aspects of OO development applied here, with examples and explanations, will achieve the highest marks • Clear, justified examples and clear descriptions of your improvements to the code.
----------------------------	---