Quiz 06: Refactoring

Due Mar 6 at 10pm **Points** 10 **Questions** 5 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides.   You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	43 minutes	0 out of 10 *

* Some questions not yet graded

Score for this quiz: **0** out of 10 *
Submitted Mar 3 at 3:26pm
This attempt took 43 minutes.

Question 1

Not yet graded / 2 pts

What is refactoring?

Your Answer:

Refactoring consists of improving the internal structure of an existing program's code, while preserving its external behavior. It makes it easier to understand the code for the people working in the environment where the single file is used by many making it easier to maintain.

Refactoring favors the emergence of reusable design elements and code modules. Thus it reduces complexity and improves readability.

Question 2

Not yet graded / 2 pts

Describe the two Hats of Software Development.

Your Answer:

The two hats of software development are adding functionalities to the system and refactoring.

Adding functionalities means modifying the existing code by adding new features to the system. The new code is added and new test cases are also added to test the recent changes made in the functions of the system. This increases the user experience.

The refactoring focuses on the structure of the code to make it more readable and understandable without making changes to the code. No additional test cases and functionalities are added for this method unless necessary. The user should not feel that the code has been refactored.

Question 3

Not yet graded / 2 pts

What is technical debt? How is it paid off?

Your Answer:

Technical debt is a concept in agile programming that reflects the extra development work that arises when code that is easy to implement in the short run is used instead of applying the best overall solution.

It is the cost which occurs because of bad design of the code, additional testing and maintenance effort.

We can pay off the technical debt by performing refactoring which makes the code easier to read and understand. Since the code is refactored hence there are not too long methods or class hence the program is divided in modules and modular programs have proven to be easier to work with.

Question 4

Not yet graded / 2 pts

Describe two other reasons (besides technical debt) for refactoring code.

Your Answer:

Refactoring helps to find the bugs easily. As the program is readable and understandable, this makes it easier for us to debug the program and find the bugs easily.

Refactoring makes it easier to program faster since the code is divided into smaller size modules and we do not need to rewrite some of the functionality which is used frequently since refactoring should have made it as a separate function and thus we can reuse the code. So, reusability makes it better and faster for developers to refactor.

Refactoring makes program easier to understand and it also helps us to improve bug solving, feature addition and makes it faster to develop new modules.

Question 5

Not yet graded / 2 pts

Describe two bad smells in code that suggest you should refactor.

Your Answer:

Bad smells aren’t necessarily bugs, but instead is something that isn’t optimal and may suggest a bigger problem such as design flaws and opportunity for improvements.

When there is too much repetition of the code, it is the time to refactor the code as it increases the space and complexity of the program. When a method or class is too long, the program should be refactored. If a class is dependent on the other, we should inherit the class from existing one rather than developing it from the scratch making the functionalities of previous class more usable.

When a method is too long, we should divide the method into smaller modules to make it more readable and usable.

Quiz Score: **0** out of 10