Quiz 06: Refactoring

Due Oct 12 at 10pm **Points** 10 **Questions** 6 **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	5,560 minutes	0 out of 10 *

* Some questions not yet graded

Score for this quiz: **0** out of 10 *
Submitted Oct 12 at 9:20pm
This attempt took 5,560 minutes.

Question 1

Not yet graded / 2 pts

Describe the Litter Pickup refactoring workflow

Your Answer:

As the name suggests "Litter Pickup refactoring", it works based on the idea of make the final code better compare to earliest code rather than ignoring the problems and assume that someone else pick it up and fix that bug or error. As a developer, he/she find a bad smell in the code and fix it at the same time like littler should be picked up. Bad code should be improved, no matter what your abilities or you have a different schedule.

Question 2

Not yet graded / 2 pts

What is technical debt? How is it paid off?

Your Answer:

Technical debt is caused due to bad design of task or code and choose an easy solution instead of right and best approach to eliminate the problems. Technical debt is concept that reflects the implied cost of .extra development, testing and maintenance effort
.It is paid off by Refactoring to fix bad smell code

Question 3

Not yet graded / 2 pts

Describe the TDD refactoring workflow

Your Answer:

TDD refactoring starting with writing a test code before any code is written. After that, if the tests become fail, rewrite the code and make sure that it will pass all tests and run it again. Same as, continuously write the code and and run it until it will pass all tests. Once it is completed, and we are ready to move further on next step, repeat the cycle again from the starting point, which is writing test code. Hence, at one time only few amount of code is there, which is unchecked or untested, which leaves less possibility for errors or bugs.

Question 4

Not yet graded / 2 pts

Describe two benefits of refactoring

Your Answer:

One of the most important benefit of refactoring is, it makes code or program easy to understand and simple to read, by modifying the repetitive code. It is easy for someone who has never seen the code before, to understand that code easily and rapidly with good accuracy.

Another key benefit of refactoring is, it helps to find a bugs or an errors, which are hidden behind smelly code and not give proper outcome. It helps to found these type of bugs, which are not found before and give best appropriate solution to fit it and get perfect outcome.

Question 5

Not yet graded / 2 pts

What is refactoring?

Your Answer:

> As the name suggests "refactoring" means simplify the code without changing it's behavior. It is defined as, internal structure of the software changed to make code more simpler, easy to understand, and cheaper to modify without affecting it's observable behavior.

Question 6

0 / 0 pts

“I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.”

Correct!

● True

○ False

Quiz Score: **0** out of 10