IT & Computer Science

Instructor Name: Zubaria Noureen

Name: Abuzar Khan **Reg #:** B22F1053SE23

Section: SE (GREEN) **Date:** 16/10/2025

Software Re-Engineering

Assignment 02

Answers

01. Composing Methods

Refactors that improve how code is structured inside methods.

Sub-methods:

- Extract Method
- Inline Method
- Replace Temp with Query
- Introduce Explaining Variable
- Remove Assignments to Parameters
- Substitute Algorithm

Advantages:

- Makes long, messy methods shorter and more readable.
- Improves reusability of code.
- Easier debugging and testing.

When to Apply:

When your method is long, repetitive, or hard to understand.

02. Moving Features Between Objects

Focuses on distributing responsibilities properly between classes.

Sub-methods:

- Move Method
- Move Field
- Extract Class
- Inline Class
- Hide Delegate
- Remove Middleman

Advantages:

- Improves cohesion and reduces coupling.
- Keeps classes focused on one purpose.

When to Apply:

When one class is overloaded with responsibilities or relies too much on another.

03. Organizing Data

Refactors to handle data structures and fields more cleanly.

Sub-methods:

- Rename Field
- Replace Data Value with Object
- Change Value to Reference
- Change Reference to Value
- Replace Array with Object
- Encapsulate Field
- Encapsulate Collection

Advantages:

- Makes data structures more expressive and safer to modify.
- Reduces duplication and inconsistencies.

When to Apply:

When you have confusing, repetitive, or unstructured data fields.

04. Simplifying Conditional Expressions

Targets messy if/else or complex logic.

Sub-methods:

- Decompose Conditional
- Consolidate Conditional Expression
- Consolidate Duplicate Conditional Fragments
- Remove Control Flag
- Replace Nested Conditional with Guard Clauses
- Replace Conditional with Polymorphism

Advantages:

- Makes decision-making logic clear and maintainable.
- Prevents logic errors in deeply nested conditions.

When to Apply:

When conditionals become too complex, repetitive, or hard to follow.

05. Simplifying Method Calls

Makes method invocations more straightforward.

Sub-methods:

- Rename Method
- Add Parameter
- Remove Parameter
- Replace Parameter with Explicit Methods
- Preserve Whole Object
- Replace Parameter with Method
- Introduce Parameter Object
- Hide Method
- Replace Constructor with Factory Method
- Encapsulate Downcast

Advantages:

- Improves method usability and clarity.
- Reduces unnecessary parameters or confusion in API design.

When to Apply:

When method calls are confusing, inconsistent, or frequently misused.

Challenges You Might Mention

To make your report realistic:

- Understanding the difference between refactoring and rewriting code.
- Identifying when refactoring is safe (no behaviour change).
- Finding suitable examples for each technique.
- Maintaining readability while showing screenshots.
- Time constraints in analysing all five categories deeply.

END!