Kingdom of Saudi Arabia
Ministry of Education
University of Jeddah
College of Computer Science and Engineering
Department of Software Engineering

المملكة العربية السعودية
وزارة التعليم
جامعة جدّة
كلية علوم وهندسة الحاسب
قسم هندسة البرمجيات

Course Project
CCSW 413 Software Design Patterns

Total Marks: 15 Marks

Submission of **Stage 1** Deadline: Week 6 (February 22, 2024)

Submission of **Stage 2** Deadline: Week 8 (March 16, 2024)

Submission of **Stage 3** Deadline: Week 11 (April 19, 2024)

Submission of **Stage 4** Deadline: Week 14 (May 09, 2024)

Final Presentation Deadline: Week 15 (May 12, 2024)

Objective:

The goal of this project is to teach students the necessary analytical skills to refactor existing open-source Java projects. Students will choose an open-source Java project, either from GitHub or one they have developed in previous coursework, and apply Java creational, structural, and behavioural design patterns to enhance the project's maintainability, extensibility, and overall code quality. Students must choose a Java open-source project of an acceptable size that is suitable for refactoring. The open-source project has to include **a minimum of 5 unique** classes that are not tied together in an inheritance relationship. The project deliverables are organised into four stages. In the following, the project deliverables for each stage are described with their respective deadlines;

Project Stages:

Stage 1: Project Selection and Problem Statement Definition

- **Students organize into groups of at most 4 members.**
- **Each group consults with the instructor to select an open-source Java project appropriate for refactoring.**

- Each groups submit a PDF report (Phase 1: Project Assignment on Blackboard) containing sections provide the following details:
    1. **Group members' details.**
    2. **Project Problem Statement:** This section defines the problem statement of the open-source Java project.
    3. **Class diagram:** This section includes class diagrams illustrating the detailed structure of the open-source Java project.
    4. **GitHub repository:** This section includes a Link to the group's GitHub repository containing the project source code. Each group must create a GitHub account and upload the source code of their Java project on their GitHub account repository. The repository that includes your project source code should be public and a link to this project GitHub repository has to be included to this section of the report.
- **Deadline:** The deadline for submitting this report is **22/02/2024.**

Stage 2: Applying Creational Design Patterns

- **Each group refactors their project, applying two of the following Java creational design patterns:**
    - **Singleton design pattern**
    - **Prototype design pattern**
    - **Builder design pattern**
    - **Factory design pattern**
    - **Abstract factory design pattern**
- For **stage 2**, each project group must submit on Blackboard under (Project Stage 2 assignment) a pdf report that adds to the report that has been submitted in the first project stage and includes the following sections:
    - **Creational Design Patterns:** This section briefly describes each creational design pattern that has been used to refactor your group's open-source Java project. The description must provide insights on the intent of the used creational design pattern along with advantages gained from refactoring the code to use these creational design patterns.
    - **Class Diagram after Applying Creational Design Patterns:** This section includes class diagrams reflecting the project's structure post-refactoring.
    - **Source Code:** should be updated in the GitHub repository you created in Stage1.
- **Deadline:** The deadline for submitting this report is **16/03/2024**.

Stage 3: Applying Structural Design Patterns

- **Each group further refactors their project, applying <u>two</u> of the following Java structural design patterns:**
    - **Adapter design pattern**
    - **Decorator design pattern**
    - **Composite design pattern**
    - **Proxy design pattern**
    - **Bridge design pattern**
    - **Flyweight design pattern**
- For **<u>stage 3</u>**, each project group must submit on Blackboard under (Project Stage 2 assignment) a pdf report that adds to the report that has been submitted in the second project stage and includes the following new sections:
    - **Structural Design Patterns:** This section includes:
        - Brief descriptions of each applied structural design pattern.
        - Insights into the intent and advantages gained from the refactoring.
    - **Class Diagram after Applying Structural Design Patterns:** This section includes class diagrams reflecting the project's structure post-refactoring.
    - **Source Code:** should be updated in the GitHub repository you created in Stage1.
- **Deadline:** The deadline for submitting this report is **19/04/2024**
-

Stage 4: Applying Behavioural Design Patterns

- Each group continues the refactoring process, applying **<u>two</u>** of the following Java behavioural design patterns:
    - **Observer design pattern**
    - **Strategy design pattern**
    - **Command design pattern**
    - **State design pattern**
    - **Chain of Responsibility design pattern**
- For **<u>stage 4</u>**, each project group must submit on Blackboard under (Project Stage 4 assignment) a pdf report that adds to the report that has been submitted in the third project stage and includes the following new sections:

  - **Behavioural Design Patterns:** This section includes:
    - Brief descriptions of each applied behavioural design pattern.
    - Insights into the intent and advantages gained from the refactoring.
  - **Class Diagram after Applying Behavioural Design Patterns:** This section includes class diagrams reflecting the project's structure post-refactoring.
  - **Source Code:** should be updated in the GitHub repository you created in Stage1.
- **Deadline:** The deadline for submitting this report is **09/05/2024**

Please be aware that a presentation will be required at the end of the semester where you will present you project in a live-demo setup. Please be well prepared and make sure to double check that your laptop and program are functioning properly before your presentation.

Evaluation Sheet:

Grades will be determined according to the following:

Assessment	Grade
CLO 2.1 Select appropriate design patterns for design problems (30 points) ***Note this grade is divided on all four stages**	
Class Organization: Classes should be organized in a way that promotes clarity and ease of understanding. The evaluation is based on the following criteria: • Are class names descriptive and indicative of their purpose? • Are classes grouped based on related functionality? • Are common functionalities encapsulated into reusable components or functions? • Does the class diagram reflect the source code? • Does the class diagram represent each design pattern correctly?	20% (5% each stage)
Class Diagrams: Quality and clarity of both the initial and updated class diagrams.	5% (1.25% each stage)
Use of source control: The project employs version control that contributes to the overall success of the refactoring process. The evaluation is based on the following criteria: • Does the commit history reflect a consistent and steady pace of contributions from all team members? • Are commit messages descriptive enough to understand the context of changes made by each student?	5% (1.25% each stage)
CLO 2.2 Implement solutions based on design patterns (60 points)	

Creational Design Patterns: implement relevant design patterns to enhance code organization and maintainability. The evaluation is based on the following criteria: • Correct application and integration of two creational design patterns. • Students demonstrate understanding and justification of the design choices.	20%
Structural Design Patterns: implement relevant design patterns to enhance code organization and maintainability. The evaluation is based on the following criteria: • Correct application and integration of two Structural design patterns. • Students demonstrate understanding and justification of the design choices.	20%
Behaviour Design Patterns: implement relevant design patterns to enhance code organization and maintainability. The evaluation is based on the following criteria: • Correct application and integration of two Behaviour design patterns. • Students demonstrate understanding and justification of the design choices.	20%
CLO 3.1 (10 points)	
Presentation: • Content is of high quality and presented within allocated time. • Presentation time divided reasonably among students (all participated). • Students demonstrate understanding and justification of the design choices.	10%
Common Issues	
Students missed submission deadline	-20
Student(s) missed the presentation (take points only from students that missed the presentation)	-10
Student did **NOT** follow submission instructions	-5
The project lacks proper size (It is too simple)	-20
The submitted code does **NOT** seem to be the student's own work. (e.g., might be a copy of another student's work or an online premade template)	-20 for each stage