CS 2340 Week 8 - Group Activity - SOLID & GRASP: Team 32

Participating Members: Rafay Hadi, Muneeb Ali Asif, Mitchell Ford, Arish Virani, Rayan Castilla, Saibalaji Nagarajan

We focused on adhering to SOLID and GRASP principles by clearly defining class responsibilities and using interfaces effectively. For example, the Single Responsibility Principle (SRP) is evident as each class, like Developer, TeamManager, and Tester, encapsulates specific functionalities related to their roles, making the code easier to maintain and understand. We also respected the Open/Closed Principle (OCP) by allowing the introduction of new task types, such as HighPriority and RecurringTask, without modifying existing classes. The use of interfaces like IDeveloper and ITeamMember promotes the Dependency Inversion Principle (DIP), ensuring high-level modules remain independent of low-level implementations. These layers of abstraction ensure that all our high-level modules don't require a specific low level implementation to run. Additionally, we applied GRASP's Information Expert principle by assigning responsibilities to classes that have the necessary information, like Project managing its own tasks and team members. The Creator pattern is reflected in the System class, which oversees project creation and deletion. We also followed Liskov's substitution principle, which can be seen in Task (Task can be replaced by both RecurringTask and HighPriority). Overall, our design fosters modularity and scalability, aligning with best practices in software design.

SOLID/GRASP design principles mentioned in writeup:

1. Single Responsibility Principle
2. Open/Closed Principle
3. Dependency Inversion Principle
4. Information Expert
5. Creator Pattern
6. Liskov's Substitution Principle