Explain Programming with Classes

Abstraction:

Abstraction refers to revealing only the necessary features of an item, and hiding everything else. I abstracted in my final program by having a base Goal class with shared members (like name, description, points) and common methods (GetDetailsString()), so the rest of the program could interact with goals without needing to know implementation details of those goals. This made things nice for future changes, since now I can add a new goal type (say TimedGoal) without rewriting the entire menu system, just as long as my new class also implements an abstract methods with the “Goal” interface.

Encapsulation:

Encapsulation is about hiding data and exposing it via methods. I also utilized encapsulation by saving things _points, isComplete or _amountCompleted as fields on the goal classes then calling methods like RecordEvent() rather than editing those variables directly. This makes it flexible, because if at some point I decided to change how the completion was performed (ie add validation, or require bonus rules) I could update that logic within class without breaking other code.

Inheritance:

Conceptually, inheritance is a way to form new classes (instances of which we’ll call objects) using classes that have already been defined — this allows us to reuse (or inherit) attributes and methods from an existing class. SimpleGoal, EternalGoal and ChecklistGoal are siblings subclasses of Goal, implementing inheritance. This also made my program easier to lift because common code is kept in Goal and new types of goals can reuse it rather than copy logic.

Polymorphism:

By polymorphism I mean that the same base-type variable can refer to two different derived types and at runtime runs the appropriate overridden (derived) method. I've utilized a polymorphism pattern where I stored various types of goals in a List, and made calls on each goal such as RecordEvent() or GetDetailsString(). This made the program flexible since the goal manager doesn’t have to include an if statement for each type of goal — new types can be introduced and would work as long as they implement the required methods.