ARTICULATE THE PRINCIPLES OF PROGRAMMING WITH CLASSES.

- **Abstraction**:

Abstraction means focusing on what something does without worrying about how it does it. It allows us to focus on high-level concepts rather than implementation details. In our project, we focused on what videos and comments do (like storing information and displaying it), without worrying about the complex stuff happening behind the scenes on YouTube. Also, if YouTube changes its video structure, we only need to update the **Video** and **Comment** classes, keeping the rest of the code unchanged.

- **Encapsulation**:

Encapsulation means putting related things together in a class and keeping them separate from other classes. By hiding internal details of classes, changes can be made to one part of the code without affecting others. In our project, we kept information about products, customers, and orders separate in different classes, making it easier to manage and understand our code. Also, altering the way orders are processed won't impact the structure of other classes in the online ordering system.

- **Inheritance**:

Inheritance allows one class to inherit traits (like properties and methods) from another class. It promotes code reuse and organization. In our project, we had different types of events, like lectures and receptions, that inherited common traits from a base event class. This helped us avoid repeating the same code for each event type. Furthermore, adding a new event type to the event planning system only requires creating a new subclass, leveraging existing functionality from the base class.

- **Polymorphism**:

Polymorphism allows objects to be treated as instances of their parent class. It allows objects to be treated uniformly, enabling easy addition of new types without modifying existing code. This means that even though we have different types of activities in our project (like running and swimming), we can still use them in the same way. Introducing a new activity type in the exercise tracking system involves simply creating a new subclass with overridden methods. This makes our code more flexible and easier to work with.