1. **Create a Student Class:**
   * Create a class named Student with the following:
     + A **property** named Name of type string to represent the student's name.
     + A **property** named Age of type int to represent the student's age.
     + A **property** named StudentID of type int to represent the student's ID.
     + Add an **auto property** named Email to represent the student’s email.
2. **Add Constants:**
   * Add a **constant** named MinAge to represent the minimum allowed age for a student (18 years).
   * Add a **constant** named MaxAge to represent the maximum allowed age for a student (40 years).
3. **Add a Method:**
   * Add a **method** named GetDetails that displays a message containing the student’s name, age, and ID.
4. **Create Objects:**
   * Create an **object** from the Student class with the following values:
     + Name: "Ahmad"
     + Age: 22
     + Student ID: 1001
   * Call the GetDetails method for the object.

**Add these Concepts**

1. **Add a Constructor:**
   * Add a **constructor** to the Student class that:
     + Accepts the name, age, and student ID as parameters.
     + Updates the class fields with the given values.
2. **Add a Destructor:**
   * Add a **destructor** to the Student class that prints a message when the object is deleted.
3. **Implement Encapsulation:**
   * Make all fields in the Student class **private**. (if we have field )
   * Add **Getters** and **Setters** to control access to these fields.
4. **The Four OOP Principles:**
   * Demonstrate **Encapsulation** by making student data private and using Getters and Setters.

**Additional Instructions:**

1. **Examples:**
   * Create two new objects of the Student class.
   * Display their details using the GetDetails method.
2. **Code Comments:**
   * Add comments to the code to explain each step.

**Questions** :

1. What is **class**
2. What is **object**
3. Difference between class && object
4. Mention the OOP Principles
5. What is property
6. What is field
7. What is **constructor**
8. What is **encapsulation**