**Instructions:**

 **Create a Console Application**:

* Create a new project in Visual Studio of type Console Application and name it OOPLab.

 **Define Classes and Implement Concepts**:

* **Copy Constructor**:
  + Create a class Person with properties Name, Age, and Address.
  + Add a general constructor to this class to set the values for the properties.
  + Add a copy constructor to create a copy of an existing Person object.
  + Write a ToString method to display the person's information in a formatted way.
* **Value Types and Reference Types**:
  + Create a struct ValueTypeExample with a property Value.
  + Create a class ReferenceTypeExample with a property Value.
* **Garbage Collection**:
  + Create a class GCClass with a constructor that prints a message when the object is created and a destructor that prints a message when the object is destroyed.
* **Inheritance**:
  + Create a class Animal with a property Name and a virtual method Move that prints a message.
  + Create a class Bird that inherits from Animal and overrides the Move method.
  + Create a sealed class Eagle that inherits from Bird.
* **Abstract Classes**:
  + Create an abstract class AbstractAnimal with a property Name and an abstract method MakeSound.
  + Create a class Dog that inherits from AbstractAnimal and implements the MakeSound method.

 **Main Program**:

* Write code in the Main method to test the different concepts:
  + Create two objects of the Person class using the copy constructor and display their information.
  + Test the difference between value types and reference types.
  + Create an object of GCClass and invoke garbage collection.
  + Create objects of Animal and Bird to test inheritance and polymorphism.
  + Create an object of Dog to test abstract classes.

 **Run the Application**:

* Ensure the printed results match the expectations for each concept.
* Observe garbage collection messages when GC.Collect() is called.