**All the answers should be written inside Program.cs.**

* **Short-answer (theory) questions → should be written as comments (// or /\* ... \*/).**
* **oop questions → all classes should be placed in the same Program.cs**

**OOP Q&A**

1. What are the six combinations of access modifier keywords and what do they do?
2. What is the difference between the static, const, and readonly keywords when applied to a type member?
3. What does a constructor do?
4. Why is the partial keyword useful?
5. What is a tuple?
6. What does the C# record keyword do?
7. What does overloading and overriding mean?
8. What is the difference between a field and a property?
9. How do you make a method parameter optional?
10. What is an interface and how is it different from an abstract class?
11. What accessibility level are members of an interface by default?
12. True/False: Polymorphism allows derived classes to provide different implementations of the same method.
13. True/False: The override keyword is used to indicate that a method in a derived class is providing its own implementation.
14. True/False: The new keyword is used to indicate that a method in a derived class is providing its own implementation.
15. True/False: Abstract methods can be used in a normal (non-abstract) class.
16. True/False: Normal (non-abstract) methods can be used in an abstract class.
17. True/False: Derived classes can override methods that were virtual in the base class.
18. True/False: Derived classes can override methods that were abstract in the base class.
19. True/False: Derived classes must override the abstract methods from the base class.
20. True/False: In a derived class, you can override a method that was neither virtual nor abstract in the base class.
21. True/False: A class that implements an interface does not have to provide an implementation for all of the members of the interface.
22. True/False: A class that implements an interface is allowed to have other members in addition to the interface members.
23. True/False: A class can inherit from more than one base class.
24. True/False: A class can implement more than one interface.

Create 3 classes in Program.cs:

**a. Person class**

* Create an **abstract class Person** with the following members:  
    
  + An Id property (int).
  + A private field salary with a public property Salary that only accepts positive values; throw an exception if a negative value is assigned.
  + A DateOfBirth property (DateTime).
  + An Address property (List of strings).

**b. Instructor class**

* Create a class Instructor that **inherits from Person**.  
    
  + Add a DepartmentId property (int).

**c. Student class**

* Create a class Student that **inherits from Person**.  
    
  + Add a property SelectedCourses, which is a list of Course objects.