**03 Object-Oriented Programming**  
Test your knowledge  
1. What are the six combinations of access modifier keywords and what do they do?

Public: accessible from outside class and derived class

Private: cannot be accessed from outside class and derived class

Protected: can be accessed by derived class only

Internal: create public and protected accessibility

Protected internal: combination of protected and internal, accessed from derived class and namespace

Private protected: combination of private and protected, accessed within same class and derived class.

2.What is the difference between the static, const, and readonly keywords when applied to a type member?

Static: only can be used in class, methods, properties, etc. but must be static

Const (compile time): must assign a value and sure that value would not change. Can have all access modifiers

Readonly(run time): initialized at time of declaration or constructor.

3. What does a constructor do?

Create instances. It will share same name across classes share same name.  
4. Why is the partial keyword useful?

Useful when working on a large project, spread classes over different files.   
5. What is a tuple?

Tuple is an immutable list. values subject not change (ex. Password)  
6. What does the C# record keyword do?

Used for reference variable, when you need to make comparison but don’t want to copy values, and want to use reference variable.  
7. What does overloading and overriding mean?

Overloading (happens in parent class)is when you have same function name but different parameter.

Overriding (virtual/abstract in parent class, happen in child class) is when you have same function name and parameter, return different value  
8. What is the difference between a field and a property?

Field is the variables declared first

Property or attribute is declared publicly after the field so variable can be used properly.

Field: string name; int age;

Property: string name {get;set;} int age{get;set;}  
9. How do you make a method parameter optional?

1. make parameter default. If no overwrite parameter in method then it will return default value

2. overloading: same function name but different parameter list (1 parameter or 3 parameters) 🡪 not the best way

3. param keyword: this allows to create many parameters as long as the datatype is matched 🡪 not the best way  
10. What is an interface and how is it different from abstract class?

Interface: only can define functionality, not implement it. only method and properties.

Abstract: It is designed to be inherited by subclass to implement or override method (in virtual), can have constructors.   
11. What accessibility level are members of an interface?

Public by default  
12. True/False. Polymorphism allows derived classes to provide different implementations of the same method. True  
13. True/False. The override keyword is used to indicate that a method in a derived class is providing its own implementation of a method. True  
14. True/False. The new keyword is used to indicate that a method in a derived class is providing its own implementation of a method. False  
15. True/False. Abstract methods can be used in a normal (non-abstract) class. False

**16.**True/False. Normal (non-abstract) methods can be used in an abstract class. True

17. True/False. Derived classes can override methods that were virtual in the base class. True

18. True/False. Derived classes can override methods that were abstract in the base class. False

19. True/False. In a derived class, you can override a method that was neither virtual non abstract in the base class. False  
**20.** True/False. A class that implements an interface does not have to provide an implementation for all of the members of the interface. True  
**21.** True/False. A class that implements an interface is allowed to have other members that aren’t defined in the interface. True  
**22.** True/False. A class can have more than one base class. True  
**23.** True/False. A class can implement more than one interface. True

**Designing and Building Classes using object-oriented principles**  
1. Write a program that that demonstrates use of four basic principles of  
object-oriented programming /Abstraction/, /Encapsulation/, /Inheritance/ and /Polymorphism/.

2. Use /Abstraction/ to define different classes for each person type such as Student and Instructor. These classes should have behavior for that type of person  
3. Use /Encapsulation/ to keep many details private in each class.   
4. Use /Inheritance/ by leveraging the implementation already created in the Person class to save code in Student and Instructor classes.  
5. Use /Polymorphism/ to create virtual methods that derived classes could override to create specific behavior such as salary calculations.  
6. Make sure to create appropriate /interfaces/ such as ICourseService, IStudentService, IInstructorService, IDepartmentService, IPersonService, IPersonService (should have person specific methods).

IStudentService, IInstructorService should inherit from IPersonService.  
Person  
 Calculate Age of the Person  
 Calculate the Salary of the person, Use decimal for salary  
 Salary cannot be negative number  
 Can have multiple Addresses, should have method to get addresses  
Instructor

Belongs to one Department and he can be Head of the Department  
 Instructor will have added bonus salary based on his experience, calculate his years of experience based on Join Date  
Student  
 Can take multiple courses  
 Calculate student GPA based on grades for courses  
 Each course will have grade from A to F  
Course  
 Will have list of enrolled students  
Department  
 Will have one Instructor as head  
 Will have Budget for school year (start and end Date Time)  
 Will offer list of courses

Explore following topics  
Fields  
Access modifiers  
Enumeration types  
Constructors  
Methods  
Properties  
Inheritance  
Interfaces  
Polymorphism