



Introduction to **PROGRAMMING CONCEPTS**

WHAT IS INTERFACE

- An interface is a contract that defines a set of methods and properties without implementation.
- Classes that implement the interface must provide the implementation for all its members.
- Interfaces are used to achieve abstraction and multiple inheritance in C# (C# does not allow multiple class inheritance, but allows multiple interfaces).



RULES OF MAKING AN INTERFACE

- Use the keyword interface to declare an interface.
- All members are implicitly public and abstract.

```
/ public interface ICustomer
{
    1 reference
    public string AddData();
    1 reference
    public string GetData ();
    1 reference
    public string searchData();
    1 reference
    public string UpdateData();
}
```

RULES OF MAKING AN INTERFACE

- Class implementing an interface must implement all members.
- Interface cannot have constructors.

```
public class Customer : ICustomer
{
    1 reference
    string ICustomer.AddData(){throw new NotImplementedException();}

    1 reference
    string ICustomer.GetData(){throw new NotImplementedException();}

    1 reference
    string ICustomer.SearchData(){throw new NotImplementedException();}

    1 reference
    string ICustomer.UpdateData() { throw new NotImplementedException(); }
}
```

USE OF INTERFACE IN MULTIPLE INHERITENCE

```
1 reference
interface Buyer
{
    1 reference
    void buy();
}

1 reference
interface Seller
{
    1 reference
    void sell();
}

0 references
class Customer :Buyer, Seller {
    1 reference
    public void buy(){
        Console.WriteLine("Customer is buying products");
    }
    1 reference
    public void sell()
    {
        Console.WriteLine("Customer is selling products");
    }
}
```

EXCEPTION HANDLING

- Exception: An unexpected event or error during program execution.
- Exception Handling: Mechanism to catch and manage errors to prevent program crashes.
- Provides robust and maintainable code.

Types of Exceptions

- System-defined Exceptions – Provided by .NET, e.g.: DivideByZeroException
- User-defined Exceptions – Custom exceptions created by the programmer.

COLLECTION

Definition:

- A collection is a framework for storing, managing, and manipulating multiple objects in C#.
- More flexible than arrays for dynamic data storage.

Key Points:

- Can store multiple items dynamically.
- Provides built-in methods like Add(), Remove(), Sort().

GENERIC

Definition:

- Generics allow you to define classes, methods, or collections that work with any type, providing type safety.

Key Points:

- Use <T> to define the type.
- Avoids casting and runtime errors.
- Generics allow you to write a class or method once and reuse it with multiple data types.

LINQ IN C#

Definition:

- LINQ = Language-Integrated Query
- Used to filter, select, and sort data from collections like arrays and lists.

Key Points:

- Makes working with data easy and readable.
- Works with arrays, lists, XML, databases, etc.
- Common methods: Where(), Select(), OrderBy().

HOME

ABOUT

MORE

THANK YOU

www.reallygreatsite.com