Generic in C# Programming

- Generics allow you to write a class or method that can work with any data type
- Generics allow you to define a class with placeholders for the type of its fields, methods, parameters, etc. Generics replace these placeholders with some specific type at compile time it helps you to maximize code reuse, type safety, and performance.
- You can create your own generic classes methods and delegates.
- You may get information on the types used in a generic data type at runtime.
- A generic class or method can be defined using angle brackets < >

Advantage:

- ✓ Increases the reusability of the code.
- ✓ Generic are type safe. You get compile time errors if you try to use a different type of data than the one specified in the definition.

Generic Method:

- Generic methods process values whose data types are known only when accessing the variables that store these values.
- A generic method is declared with the generic type parameter list enclosed within angular brackets.
- Defining methods with type parameters allows you to call the method with a different type every time.
- You can declare a generic method within generic or non-generic class declaration.
- Generic methods can be declared with the following keywords virtual, override and abstract.

Generic Class:

- Generics classes define functionalities that can be used for any data type and are declared with a class declaration followed by a type parameter enclosed within angular brackets.
- Generics allow you to define a class with placeholders for the type of its fields, methods, parameters, etc. Generics replace these placeholders with some specific type at compile time