**1.Differentiate between Generic vs non -Generic Collection in C#.**

* **Generic Collections**:

**Type-Safe:** Generic collections are type-safe, meaning they can store and retrieve specific types of elements. For example, List<T> can hold a list of elements of type T.

**Compile-Time Safety:** Since the type is known at compile-time, you get compile-time safety. This helps catch type-related errors early in the development process.

**Reusability:** Generic collections provide better code reusability because the same collection type can be used for different data types

**List<int> integerList = new List<int>();**

* **Non-Generic Collections:**

**Not Type-Safe:** Non-generic collections, like the older ArrayList or Hashtable, store objects of type object. This means you lose type information and must cast elements when retrieving them.

**Runtime Type Checking:** Since the type information is not known at compile time, type-related errors might only be caught at runtime, leading to potential runtime exceptions.

**Less Reusable:** Because they work with object, non-generic collections are less reusable and can lead to more runtime errors.

**ArrayList nonGenericList = new ArrayList();**

**2.What collection is suitable for Form Design entry based on the user needs vs Form design entries that are fixed?**

**Solution:**

**Dynamic Form Entry (User-Defined Fields):**

**Suitable Collection: ExpandoObject**

**Why:** The ExpandoObject class in C# allows you to add properties dynamically. This is perfect for scenarios where users can define their form fields.

**Example:**

dynamic userDefinedForm = new ExpandoObject();

userDefinedForm.ProductName = "Smartphone";

userDefinedForm.Price = 599.99;

userDefinedForm.InStock = true;

**Fixed Form Entry (Static Fields):**

**Suitable Collection: Tuple or ValueTuple**

**Why:** For a fixed form with known fields, tuples provide a concise and expressive way to represent the data.

Example (Tuple):

Tuple<string, int, string> fixedFormTuple = Tuple.Create("Alice", 28, "New York");