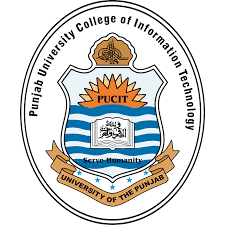
**Faculty COLLEGE OF**

**INFORMATION AND TECHNOLOGY**



**Enterprise application Development**

**Assignment -2**

**Submitted To**: Dr.Fatima Sabir

**Submitted By:**  BCSF10A026-Abdul Aziz

**Session:** 2020-2024

**Date**: October 8, 2023

**Question 1;**

Collections are majorly classified into two types.

1. Generic Collections
2. Non-Generic Collections

**Generic Collections**

A Generic collection provides the type safety without derivation from a basic collection type and the implementation of type-specific members. The Generic Collection classes are found in the name space "System.Collections.Generics." Internally, Generic Collections store elements in arrays of their respective types.

Generic collections are often classified into five types.

### List:

In Generic List, we have to specify a data type to its contents, and all elements will have the same data type.

### Dictionary:

Dictionaries usually store data in key-value pairs, and we have to specify both data types beforehand.

### Stack:

Values are kept in Stack using LIFO (Last In First Out). It offers the Push() and Pop() & Peek() methods to add and retrieve values, respectively. In generic Stack, we have to specify the datatypes of its content beforehand.

### Queue:

Values are kept in a queue in a FIFO fashion (First In, First Out). The sequence in which the values were inserted is preserved. It offers the Enqueue() and [Dequeue()](https://www.simplilearn.com/tutorials/data-structure-tutorial/dequeue-in-data-structure" \o "Dequeue()" \t "https://www.simplilearn.com/tutorials/c-sharp-tutorial/_blank) methods to add and remove values from the collection. In the generic queue, we have to specify the datatypes of its content beforehand.

**Non-Generic Collections**

Non-generic collections are specialized data storage and retrieval classes that handle stacks, queues, lists, and hash tables. The "System.Collections" namespace contains the Non-generic Collection classes. Non-generic collections store elements in object arrays internally, allowing them to hold any data type.

### Array List:

The array's size might change during use since it is dynamic, which implies it is not static. It offers functions that are comparable to those in the generic List class.

### Hash Table:

A hash table data structure is made up of key-value pairs. The hash values of the keys are compared to find the values. It offers functions that are comparable to those in the generic dictionary class.

### Stack:

It's a FIFO (first-in, first-out) list. Hence it works similarly to the Stack class in generic collections.

### Queue:

A first-in, first-out collection of items is represented by it. When you require first-in, first-out access to objects, you utilize it. It offers functions that are comparable to those in the generic Queue collection.

**Question 2:**

For form design where the requirements can vary based on user needs, dynamic collections like lists or dictionaries are suitable. These collections can flexibly adapt to changing requirements by adding or removing elements. They allow for dynamic creation and manipulation of form elements, providing a personalized user experience.

On the other hand, if the form design is fixed and unchanging, using a static collection such as arrays or fixed-size data structures is appropriate. These provide simplicity and consistency in form design and are easier to maintain for unchanging requirements. Fixed collections are ideal when you have predefined, consistent form structures that do not need to adapt to user input.

The choice between dynamic and static collections depends on the specific needs of your application, whether it requires adaptability or remains constant in form design.