**Exercise 1: Working with List<T>**

**Task**: Create a program that uses a List<T> to store and manage a collection of integers.

1. Create a List<int> to store integer items.
2. Add five different integers to the List<int>.
3. Display all items in the List<int>.
4. Remove the third item from the List<int>.
5. Insert a new integer at the second position.
6. Display the total count of items in the List<int>.
7. Sort the list in ascending order and display the sorted list.

**Exercise 2: Working with Dictionary<TKey, TValue>**

**Task**: Create a program that uses a Dictionary<TKey, TValue> to store and manage a collection of key-value pairs.

1. Create a Dictionary<string, int> to store student names (key) and their ages (value).
2. Add five different students and their ages to the Dictionary<string, int>.
3. Display all key-value pairs in the dictionary.
4. Remove a student by name.
5. Update the age of a specific student.
6. Check if a student exists in the dictionary by name.
7. Display the total count of students in the dictionary.

**Exercise 3: Working with Queue<T>**

**Task**: Create a program that uses a Queue<T> to manage a queue of tasks.

1. Create a Queue<string> to store a list of tasks.
2. Add five different tasks to the Queue<string>.
3. Display all tasks in the queue.
4. Dequeue the first task and display it.
5. Peek at the next task in the queue without dequeuing it.
6. Display the total count of tasks in the queue.

**Exercise 4: Working with Stack<T>**

**Task**: Create a program that uses a Stack<T> to manage a stack of books.

1. Create a Stack<string> to store book titles.
2. Add five different book titles to the Stack<string>.
3. Display all book titles in the stack.
4. Pop the top book from the stack and display it.
5. Peek at the top book in the stack without popping it.
6. Display the total count of books in the stack.

**Exercise 5: Working with SortedList<TKey, TValue>**

**Task**: Create a program that uses a SortedList<TKey, TValue> to store and manage a collection of employee names (key) and their salaries (value).

1. Create a SortedList<string, double> to store employee names (key) and their salaries (value).
2. Add five different employees and their salaries to the SortedList<string, double>.
3. Display all key-value pairs in the sorted list.
4. Remove an employee by name.
5. Update the salary of a specific employee.
6. Display the total count of employees in the sorted list.

**Exercise 6: Working with HashSet<T>**

**Task**: Create a program that uses a HashSet<T> to store and manage a collection of unique integers.

1. Create a HashSet<int> to store integer items.
2. Add ten different integers to the HashSet<int>, including some duplicates.
3. Display all items in the HashSet<int>.
4. Remove a specific integer from the hash set.
5. Check if a specific integer exists in the hash set.
6. Display the total count of unique items in the hash set.