C# Day3

1. **Describe the problem generics address.**

   Generics address the problem of code duplication and type safety by allowing you to create classes, methods, and data structures that can operate on any data type. This helps to create reusable and type-safe code without needing to write multiple versions of the same class or method for different data types.

2. **How would you create a list of strings, using the generic List class?**

   ```
   List<string> stringList = new List<string>();
   ```

3. **How many generic type parameters does the Dictionary class have?**

   The `Dictionary` class has two generic type parameters: `TKey` and `TValue`.

4. **True/False. When a generic class has multiple type parameters, they must all match.**

   False. When a generic class has multiple type parameters, they do not need to match; each parameter can be a different type.

5. **What method is used to add items to a List object?**

   The `Add` method is used to add items to a `List` object.

   ```
   stringList.Add("example");
   ```

6. **Name two methods that cause items to be removed from a List.**

   `Remove` and `RemoveAt`.

   ```
   stringList.Remove("example"); // Removes the first occurrence of the specified item
   stringList.RemoveAt(0); // Removes the item at the specified index
   ```

7. **How do you indicate that a class has a generic type parameter?**

   By using angle brackets `<T>` after the class name.

   ```
   public class MyGenericClass<T>
   {
       // implementations
   }
   ```

8. **True/False. Generic classes can only have one generic type parameter.**

   False. Generic classes can have multiple generic type parameters.

```
public class MyGenericClass<T1, T2>
{
    // implementation
}
```

9. **True/False. Generic type constraints limit what can be used for the generic type.**

   True. Generic type constraints limit what types can be used as the generic type parameter by specifying requirements like implementing a particular interface or inheriting from a specific class.

10. **True/False. Constraints let you use the methods of the thing you are constraining to.**

- True. Constraints allow us to use the methods and properties of the type you are constraining to, providing more functionality and type safety.