Assignment\_4\_OOP

1. Describe the problem generics address.

* Generics solve the problem of writing type-safe and reusable code for different data types. They allow developers to define classes, methods, and interfaces with placeholders for types, eliminating the need for multiple versions of the same code or unsafe type casting. By enforcing compile-time type checking, generics reduce errors, improve code clarity, and can enhance performance.

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 in C# has two generic type parameters:
  1. TKey — the type of keys in the dictionary
  2. TValue — the type of values associated with those keys

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

* False, When a generic class has multiple type parameters, each type parameter can be different. They do not have to match.

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

* The method used to add items to a List object in C# is Add().

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

* Remove(item) – Removes the first occurrence of the specified item.
* RemoveAt(index) – Removes the item at the specified index.

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

* In C#, you indicate that a class has a generic type parameter by adding angle brackets (<>) with a type placeholder after the class name. For instance, public class MyClass<T>.

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

* False, Generic classes in C# can have one or more generic type parameters.

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

* True, Generic type constraints in C# limit the types that can be used as a generic type parameter. They allow the compiler to enforce certain requirements, such as requiring the type to be a reference type, a value type, have a parameterless constructor, or implement a specific interface.

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

* True, In C#, generic constraints let the compiler know that the type parameter supports certain members or behaviors, so you can safely use those methods or properties inside the generic class or method.