Pinellas Technical College Clearwater November/25/2021

.NET Application Development & Programming

.NET Programmer OCPD.1000: Intermediate .Net Concepts

Student Alexander Sanchez Professor Adam Seebeck

**Chapter 3 Assignment (pg 73-77)**

Top of Form

Bottom of Form

**Textbook Chapter 3**: "Objects and Types"

**Read the following: (**pages 73-77)

- Create a Word Document called Chapter 3. Answer the following questions.

1. What is a data type, and what is a custom data type?

A data type describes the format and size of (amount of memory occupy by) a data item and defines what types of operations can be performed with the item. It is a classification of data which tells the compiler or interpreter how the programmer intends to use the data. Most programming languages support various types of data, including integer, real, character or string, and Boolean. A custom data type (CDT) is **a designer-defined data structure that represents a logical grouping of related data**, such as Employee and Contract. ... The life cycle of a CDT depends on where the authoritative data structure for your application is designed.

1. What is the difference between a class and a structure?

Classes on struct are essentially templates from which you can create objects. Structs differ from classes because they do not need to be allocated on the heap (classes are reference types and are always allocated on the heap) structs are value types and are usually stored on the stack. Also, structs cannot derive from a base struct

1. List a few members of a class and give a brief description (page 75)

* **Fields.** A field is a data member of a class. It is a variable off a type that is a member of a class.
* **Constants.** Constants are associated with the class (also they do not have the static modifier). the compiler replaces constants everywhere they are used with the real value.
* **Constructors.** Constructors are special functions that are called automatically when an object is instantiated. they must have the same name as the class to which they belong and cannot have a return type. constructors are useful for initialization.

1. Explain the difference between static and instance

Static classes cannot be instantiated. Static classes can only have static methods. Instance methods must be called on the instances of the class, not the class itself. Static methods must be called on the class itself, not on the instances of the class.

1. Explain access modifiers on data types (example public, private, etc.)

Access Modifiers

|  |  |
| --- | --- |
| **Modifier** | **Description** |
| public | The code is accessible for all classes |
| private | The code is only accessible within the same class |
| protected | The code is accessible within the same class, or in a class that is inherited from that class. You will learn more about inheritance in a later chapter |

**Save your assignment. You will submit it later.**

========================================================

**Textbook Chapter 3**: "Objects and Types"

**Read the following: (**pages 78-90)

- Open your Word Document called Chapter 3. Answer the following questions.

1. What are the benefits of using methods?

Ability to **reuse the same code** which ultimately **saves the excessive use of memory**, acts as a time saver and more importantly, it provides a **better readability of code**

1. What is the difference between a static and instance method?

Static method means which will exist as a single copy for a class. But instance methods exist as multiple copies depending on the number of instances created for that class. Static methods can be invoked by using class reference. Instance or non-static methods are invoked by using object reference.

1. Give an example of a method signature.

A method signature is the combination of the **method name and the parameter list**.

**Example** DoSomething (int, int)

1. Describe the purpose of overloading methods

Method overloading **increases the readability of the program**. This provides **flexibility to programmers** so that they can call the same method for different types of data. This **makes the code look clean**. This **reduces the execution time** because the binding is done in compilation time itself.

1. Give an example of a default argument in a method.

Public static void WelcomeMessage**(string name=”Guest”)**

{

ConsoleWriteLine($Hello,{name}”);

{

1. How could a default argument effect an overloaded method?

Default arguments are a convenience, as function overloading is a convenience. Both features allow you to use a single function name in different situations. The difference is that with default arguments the compiler is substituting arguments when you don't want to put them in yourself, but **in general they won’t conflict if the argument lists are different.**

1. Explain the impact of using different access modifiers on methods.

Access modifiers in C# are used to **specify the scope of accessibility** of a member of a class or type of the class itself. For example, a public class is accessible to everyone without any restrictions, while an internal class may be accessible to the assembly only.

**Save your assignment. You will submit it later.**

**Textbook Chapter 3**: "Objects and Types"

**Read the following: (**pages 90-101)

- Open your Word Document called Chapter 3. Answer the following questions.

* 1. What is a struct and how is it different from a class?

A struct is like a class and is useful for lightweight objects, such as point, rectangle and color. Lightweight objects can also be created with classes, but is often more memory efficient to use a struct. Unlike classes, structs are value types and not reference types. Structs are value types and classes are reference types. In .NET, there are two categories of types, reference types and value types. Both Classes and Structs are compound data types typically used to contain a few variables that have some logical relationship, can contain methods and events, and can support interfaces

* 1. What is an enumeration? Give an example.

An enumeration is a **value type that contains a list of named constants** such as the color type shown here. The enumeration type is defined by using the **enum** keyword (code file EnumSample/ Color.cs):

public enum Color

{

Red,

Green,

Blue,

}

**Submit your assignment.**