C# Fundamentals 2

CSF2 –
Course Objectives

- Use and understand C# complex datatypes.
- Generate and use random numbers.
- Create custom methods and understand method signatures.
- Create custom classes and understand class signatures.
- Understand and use more collection types.
- Create and use enums.
- Understand and use inheritance to simplify class creation.
- Understand and explain polymorphism at a basic level.
- Use Try / Catch blocks to contain potentially dangerous code.
- Understand and explain value vs. reference types.

Module 1:
Intro to Complex Data Types

Complex Data Types – Objectives

- How to Use the DateTime Data Type.
- Review the Method Overloads for Creating a DateTime Object.
- Discuss When It Is Appropriate to Use Each Overload

C# Review!

Homework Review!

CODE ALONG!

Complex Data Types –
Key Terms

- **Complex Datatype** – A type of variable that requires the “new” keyword to call the constructor for the object.

STOP

END MODULE 1

Homework:

1. Quizlet Vocabulary

- How to Use the DateTime Data Type.
- Review the Method Overloads for Creating a DateTime Object.
- Discuss When It Is Appropriate to Use Each Overload

Module 2:
Intro to Random

Random – Objectives

- How to Use the Random Class Data Type.
- Discuss How the Random Class Works in the Framework Class Library

CODE ALONG!

LAB!

Random –
Key Terms

- **Instance Method** – A method that is called from an instance of some variable or other object created within the application. Example: name.ToUpper()
- **Static Method** – A method that is called from a class name. Example: Console.WriteLine()

END MODULE 2

- How to Use the Random Class Data Type.
- Discuss How the Random Class Works in the Framework Class Library

Homework:

1. Quizlet Vocabulary
2. Magic 8 Ball*

Module 3:
Intro to Custom Methods

Custom Methods – Objectives

- Discuss What Methods Are Within C#.
- Implement Actions and Methods Not Built into the Framework Class Library.
- Review the Parts of the Method Signature

Custom Methods – Method Signatures

Access Modifier
private
public
*internal

Options/Keywords
*instance
static

Return Type
int
string
DateTime
bool
void (nothing)

Method Name
PascalCase
CanBeAnything

Parameters
Optional
(datatype paramName, datatype paramName)
camelCase

*Default – Not Actually Typed Out

CODE ALONG!

Custom Methods –
Key Terms

- **Method** – A group of code statements that together perform an action. Methods can be considered the verbs of the C# language.
- **Reference** – A pointer to look at another namespace (aka Project) and bring in the files that exist in it. A reference MUST be added to be able to access files that exist outside of the FCL and outside of the namespace where they will be used.
- **Using Statement**- An optional that references the namespace for files that exist outside of the namespace. Used with a reference, it allows a namespace to use files that exist in another namespace within the solution without having to reference the namespace with every call that occurs to that namespace.
- **XML** – Extensible Markup Language. XML is a metalanguage used to create other languages and define what the information that is being represented means. In C#, XML is useful for adding comments before classes and methods that help a developer implementing those classes and methods with using them when calling them from other files.

END MODULE 3

- Discuss What Methods Are Within C#.
- Implement Actions and Methods Not Built into the Framework Class Library.
- Review the Parts of the Method Signature

Homework:

1. Quizlet Vocabulary
2. Complete any Unfinished Labs