

Author: Revature Engineering Associate James Weight
(Being) 'Revature Ready' ← {Said to Coley on Jan.. 29)
Date: Attributed to class. [2021 – Feb. 20]

Team

_ Leader: Revature Corporate Trainor Nick Escalona
Purpose: Fundamentals with C# and Guidance to help other Team Members
Regards: .Net Full Stack Batch 2102-Feb08-Escalona

Preface:

Engineering is a Discipline.

This perception can be implemented through a '**Technical Approach**' or a '**Behavioral One**' as a way to provide solutions and resolve issues within projects or situations regarding endeavors in life.

The Technical Approach:

Building systems of integral parts that allow technologies from different manufacturers to work together in some form.

The knowledge to reuse existing parts for new purposes and applications.

The ability to build new components as a facilitated process.

The concepts of rapid deployment and re-usability to maintain or repair existing components as needed.

As a Full Stack VB.net Developer to provide support for 'Business Entities' with 'Projects of Scope' and endeavors that pertain to growth and chains of locations as the business expands in goods or services.

C# Constructions of Code

Note: < > as default if none specified or as in a declaration.

Memory Types: Value Type – Data corresponds directly to the variable.

Reference Type – Data is stored internally as designated by the IDE 'Engine' in physical memory as allocated and the variable contains address of.

Pointer Type – Referent type* as un-managed by the IDE 'Engine' and does not ('Primitive') inherent nor does conversion occurs as typical of the other types.

A static variable has the accessibility of a 'Global Scope' and is not an aspect of C#.

Shorthand notation: UserKeyword<type, type> (value, value);

Name-Space: A segment of block(s) of code that contains 'Classes' or 'Structures' as members.

Structure (Struct): 'Value Type' that is encapsulated by data and related functionality. <internal>

Enumerators: (Enums) are a 'Value Type' set of constants.

Revature Intellectual Property as denoted by being a company employee.

Class: 'Reference Type' that is a section as a block of code which can have these members.

- 1) Fields as a variable 'Value Type' in memory,
Can be a 'Constant' declaration (const).
- 2) Properties as a specialized 'Field' with an 'Accessor Method'.
- 3) Methods as a function 'Reference Type' in memory to implement an algorithm.

Modifiers: A modification.

1) Class Type: <internal>

abstract – Can not be 'Instantiated' on its own and intended as a 'Base-Class'.

Is 'Instantiated' with an 'Inheritance Chain'.

sealed – Can not be a 'Base-Class' as derived from through Inheritance.

internal – Accessible within the class blocks of its name-space as an assembly.

new – [nested classes] → Hides an inherited member of a Name-Space
with the same name as this Class.

private – Enclosed Class in which access is only allowed within itself.

public – Access allowed to all other classes within its Name-Space or another.

void – Declaration that there is no return command written in.

static – Can not be 'Instantiated' for 'Objects' as 'new Class-Name()';.

Can not be used with 'Indexers' or 'Finalizers'.

2) Access Member Type: (Accessibility levels)

public – Access within its own assembly 'Name-Space' or another.

private – Access only within its block of code as defined by {}.

protected – Access only within its class {} & a class inherited from this.

internal – Access only within its Assembly {}s and not allowed from
another.

protected internal – Access within its own Assembly {}s and a derived class
requesting inheritance from another assembly.

private protected – Access as protected and its assembly {}s are not accessible
by other assemblies.

'Root Classes' or 'Root Structures' within a 'Name-Space' without being nested can
have an access member type of internal or public.

'Struct Members' as 'Root' or nested with a 'Class' or 'Struct' can have internal,
private, or public access and all members within can utilize the same access.

Nested 'Class' or nested 'Struct' have <private> by default as an over-ride.

'Class' members as a 'Root' with 'Nested' classes or structures can have any access
member type and all members within can utilize the same access.

'Members' of a 'Class' can have any of the 6 types except anything protected due to
structures do not support inheritance.

The child members access levels inside the parent member can not have greater
accessibility that its parent.

'Over-Loading' requires access as public with static.

Revature Intellectual Property as denoted by being a company employee.

- 3) Reference Types: class, interface, delegate, record, dynamic, object, string
- 4) params valueType keyWord: To use a parameter method to pass a single dimension array of compatible items as arguments in a 'Method Declaration' concept to the header of a method.

Recommended

Further Research of Concepts: Finalizer,

Indexer,

Extension Methods,

Abstract Method,

Virtual Method,

Collection Functionality,

Layer-Specific Functionality,

Interfaces,

Delegates,

Lambda Expression as =>,

Attributes (above class as []),

Reflections,

Nullable and Non-Nullable Types

Tuples,

Boxing Data to Reference Type,

Un-Boxing Data to Value Type,

Built-In Constructors → Object

Built-In Methods → Equals(Object), Finalize(), GetHashCode(),

GetType(), MemberWiseClone(),

ReferenceEquals(Object, Object), ToString()

Revature Intellectual Property as denoted by being a company employee.