**JOHN REY COLORADO SD2B**

ASSIGNMENT no. 1

1. What is Object Oriented Programming?

* The OOP is a programming method that is based on object that is organized by classes. This Class are allowing individual object to be grouped together.

1. Why use OOP?

* One benefit of OOP is Encapsulation where in the objects that are created in OOP are able to hide certain part of code from programmer, it use to prevent the unintentional modification in the code that might cause unwanted outcomes.
* Second one is the code can be reuse in other program.
* Used to become easy to maintain and debug the code using OOP.
* And lastly, the OOP needs to have an extensive design planning to provide design benefits over traditional style.

1. What are the Characteristics of OOP?

* Objects – The instances of a class which are used in real functionality – its variables and operations
* Abstraction – Specifying what to do but not how to do ; a flexible feature for having a overall view of an object’s functionality.
* Encapsulation – Binding data and operations of data together in a single unit – A class adhere this feature
* Inheritance and class hierarchy – Reusability and extension of existing classes
* Polymorphism – Multiple definitions for a single name - functions with same name with different functionality; saves time in investing many function names Operator and Function overloading
* Generic classes – Class definitions for unspecified data. They are known as container classes. They are flexible and reusable.
* Class libraries – Built-in language specific classes
* Message passing – Objects communicates through invoking methods and sending data to them. This feature of sending and receiving information among objects through function parameters is known as Message Passing.

1. Different OOP Languages:

* Java
* C++
* C#
* Python
* PHP
* Ruby
* Perl
* Object Pascal
* Objective-C
* Dart
* Swift
* Scala
* Common Lisp
* Smalltalk

\* SHORT EXAMPLE CODE FOR C#

* SORTING ARRAY (int)

[C#]

* // **sort int array**
* int[] intArray = new int[5] { 8, 10, 2, 6, 3 };
* **Array.Sort**(intArray);
* // write array
* foreach (int i in intArray) Console.Write(i + " "); // output: 2 3 6 8 10

SORTING ARRAY (String)

[C#]

* // **sort string array**
* string[] stringArray = new string[5] { "X", "B", "Z", "Y", "A" };
* **Array.Sort**(stringArray);
* // write array
* foreach (string str in stringArray) Console.Write(str + " "); // output: A B X Y Z

SORTING ARRAY (w/ delegates)

[C#]

* // array of custom type
* User[] users = new User[3] { new User("Betty", 23), // name, age
* new User("Susan", 20),
* new User("Lisa", 25) };

[C#]

* // **sort array by name**
* **Array.Sort**(users, **delegate**(User user1, User user2) **{**
* return user1.**Name**.CompareTo(user2.**Name**);
* **}**);
* // write array (output: **Betty**23 **Lisa**25 **Susan**20)
* foreach (User user in users) Console.Write(user.Name + user.Age + " ");

[C#]

* // **sort array by age**
* **Array.Sort**(users, **delegate**(User user1, User user2) **{**
* return user1.**Age**.CompareTo(user2.**Age**); // (user1.Age - user2.Age)
* **}**);
* // write array (output: Susan**20** Betty**23** Lisa**25**)
* foreach (User user in users) Console.Write(user.Name + user.Age + " ");
* OTHER SAMPLE (1)
* using System;
* using AnotherNameSpace; *// you will add this using statement*
* namespace MyNameSpace
* {
* class HelloWorld
* {
* static void Main(string[] args)
* {
* AnotherClass obj = new AnotherClass();
* obj.Func();
* }
* }
* }

OTHER SAMPLE (2)

* int[] array = new int[10]; *// single-dimensional array of int*
* for (int i = 0; i < array.Length; i++)
* array[i] = i;
* int[,] array2 = new int[5,10]; *// 2-dimensional array of int*
* array2[1,2] = 5;
* int[,,] array3 = new int[5,10,5]; *// 3-dimensional array of int*
* array3[0,2,4] = 9;
* int[][] arrayOfarray = new int[2]; *// Jagged array - array of array of int*
* arrayOfarray[0] = new int[4];
* arrayOfarray[0] = new int[] {1,2,15};

REFERENCES:

* <https://techterms.com/definition/oop>
* <https://owlcation.com/stem/Use-Of-Object-Oriented-Programming>
* <https://www.careerride.com/oops-characteristics.aspx>
* <https://www.quora.com/What-are-the-types-of-OOP-languages>
* <https://www.csharp-examples.net/sort-array/>
* <https://www.codeproject.com/Articles/4300/Quick-C>