**PE10 – Introduction to OOP**

**Due Thursday 17-Sept-2019 by 11:59pm**

* Which of the following are real levels of accessibility in OOP?

friend: NO

public: Yes

secure : No

private: Yes

protected: Yes

loose: No

wildcard: No

* "We must call the destructor of an object manually, or it will waste memory." True or False?

True

* Do you need to create an object in order to call a static method of its class?

- No

* Using the yuml extension in Visual Studio Code, generate a UML diagram similar to the ones shown in this chapter for the following classes and interface:
* An abstract class called HotDrink that has the methods Drink(), AddMilk(), and AddSugar (), and the properties Milk, and Sugar.
* An interface called ICup that has the methods Refill() and Wash(), and the properties Color

and Volume.

* A class called CupOfCoffee that derives from HotDrink, supports the ICup interface, and has the additional property BeanType.
* A class called CupOfTea that derives from HotDrink, supports the ICup interface, and has the additional property LeafType.
* Write some code for a function that would accept either of the two cup objects in the above example as a parameter. The function should call the AddMilk(), Drink(), and Wash() methods for any cup object it is passed.
* *Define the following terms.*

class: A blueprint like data structure in which many objects can be created using it. It sets rules, functions, and parameters which an object or child class recieved.

object: An instance of a class. The object will be used in the main class/program as an individual piece of data from which it can be manipulated or manipulate other code.

constructor: The constructor is a function that sets up a class by defining variables that will be used throughout the class. It can accept methods and there can be multiple constructors in one class.

field: A variable specific to a class that is usually defined in a constructor function.

method: A function that has a specific return type. It might return an integer, string, or a boolean value.

dot notation/dot syntax:

encapsulation:

inheritance: Classes that are derived from a parent class and all share parent variables and methods unless stated otherwise.

polymorphism: When child classes derived from a parent class share the same methods as the parent class but modify something about it such as a data type. It could be seen as many versions of the same method/class.

* *True or False? (Green means I selected True and red means I selected False)*

T F You can create an object without a corresponding class.

T F A field is a variable belonging to a class.

T F A constructor is a special type of function that can only initialize a class’s fields. Constructors cannot contain any other code to do anything else.

T F Class declarations do not need the class keyword.

T F Classes are data types, just like integer and Boolean.

T F There are other flavors of C# such as Ccheddar, Cswiss and Cmozzarella

* What does it mean to have an overloaded method?

An overloaded method when you use the same method more than once with different parameters/ data types. For example, int thisMethod(int num1, int num2); and float thisMethod(float num1, float num2);

* Briefly explain how polymorphism can help store related objects in a collection.

Polymorphism can help store related objects in a collection because all of the objects

can have shared methods. Since all of the objects are derived from a parent class,

we can access these shared methods once in an array for example.

* What does it mean to override a method?

A derived class/ child class uses the same method as its parent class, however it passes in different parameters or the method is slightly modified. The compiler will run the method of the child class rather than the parent's.

***Submission***

Upload this completed document and the .SVG file for #4 to the corresponding MyCourses dropbox.