# PE10 – Introduction to OOP

1. Which of the following are real levels of accessibility in OOP?

friend

public

secure

private

protected

loose

wildcard

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

False. This happens automatically during garbage collection.

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

No, you do not need to instantiate a class to call a static method.

1. 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:
2. An abstract class called HotDrink that has the methods Drink(), AddMilk(), and AddSugar (), and the properties Milk, and Sugar.
3. An interface called ICup that has the methods Refill() and Wash(), and the properties Color

and Volume.

1. A class called CupOfCoffee that derives from HotDrink, supports the ICup interface, and has the additional property BeanType.
2. A class called CupOfTea that derives from HotDrink, supports the ICup interface, and has the additional property LeafType.
3. 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.

static void DrinkFunction(HotDrink drink)

{

ICup myICup = (ICup)drink;

drink.AddMilk();

drink.Drink();

myICup.Wash();

}

1. *Define the following terms.*

class: A blueprint from which objects are created.

object: a dynamically created instance of a class

constructor: A method in a class which is called automatically upon an object’s creation. Initializes the data members.

field: A variable of a class or struct that is declared directly in the class or struct.

method: a code block containing a series of statements that can be called and executed.

dot notation/dot syntax: a method of accessing a property of an object that involves writing the object’s name followed by a dot and then the property name.

encapsulation: the process of grouping member functions and data members into a single package or unit. Hides data that is not necessary from a user.

inheritance: a feature that allows a base class to define specific functionality that classes derived from it will inherit.

polymorphism: the ability of a class to have multiple implementations with the same name.

1. *True or 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

1. What does it mean to have an overloaded method?

An overloaded method means that there are multiple methods with the same name, but they have different method signatures.

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

Polymorphism is useful because it allows different objects related through a class to have similar functionality using minimum code.

1. What does it mean to override a method?

Overriding a method means using a unique implementation of a method instead of the inherited default using the override keyword.

## Submission

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