**Name: Session:**

**Programming II**

**Lab Exercise 4/20/2023**

**Introduction to Object Oriented Programming**

In this exercise, you will learn about OOP. OOP is just another way to organize your programs. The first thing we will do is define a class. A class is known as an abstract data type. An abstract data type contains not only data (properties) but also functions that operate on that data (methods). When you design a class, you will be designing a new data type with all of the methods that we determine go along with that data type.

As our first example, we will design a simple class Name. This class will have three properties (first, middle, and last) and two methods (showName and changeLastName). We could have more properties and methods if we wished but for now we will stick with that.

class Name

{

private string first;

private string middle;

private string last;

public Name(string f, string m, string l)

{

first = f;

middle = m;

last = l;

}

public void showName()

{

Console.WriteLine(first + " " + middle + " " + last);

}

public void changeLastName(string l)

{

last = l;

}

}

Notice that the three properties are “private”. This merely means that they can not be accessed from outside the class. So you might ask, where should you place your class code? The answer for now is place it anywhere but do not place it inside another class. As you will later learn, we can have inner classes but for now put that aside.

There are actually three methods. I said that we were going to have two methods. The first method is a very special method called New. It is known as a constructor method that construct an object from the class.

Notice that all methods are public. That is they may be accessed from outside the class (e.g. form Sub Main()). The showName method is used to access the private properties of the object. The changeLastName method is used to change the value.

Now you might ask, how do you use a class. Here is a simple example:

static void Main(string[] args)

{

Name joe = new Name("Joseph", "Alan", "Jones");

Name mary = new Name("Mary", "Alice", "Rogers");

joe.showName();

mary.showName();

Console.WriteLine("Joe and Mary got married");

mary.changeLastName("Jones");

mary.showName();

}

The first two lines of Main() are to call the constructor method to create two objects (joe and mary). We then call methods to display their names.

**Task 1**

* Create a Student class that contains private properties firstName, lastName, age, and GPA using the appropriate data types.
* Create a constructor method that sets all of the object properties.
* Write a public method to display the student information.
* Write a getGPA public method that retrieves the GPA of the student
* Write a getAge public method that retrieves the age of the student
* Write a Main() that creates three student objects, displays the information for each student and calculates the average age and GPA of each student.

**Task 2**

* Design a Fraction class that contains private properties numerator, denominator, and decimalEquivalent using the appropriate data types
* Create a constructor method that sets all object properties
* Write public method that displays the Fraction
* Write a public method reduce that will reduce the fraction and is called by the constructor
* Write a private method gcd that is called by reduce to the fraction
* Write a Main() that creates three Fraction objects and displays them. Be sure that at least one of your Fraction objects requires reduction

**Task 3**

* Modify your Fraction class to add an add, subtract, multiply, and divide methods which allow adding another Fraction object to your current Fraction object and returns a Fraction object that is the result of that operation.

**When you have completed your tasks, submit your source code and a sample output of the running program.**