Exam #1, on September 20, 2019, will consist of questions taken or inspired from this homework, the two previous homework, and from the labs.

Part I – Questions

Q. 1 - What is "an instance of the class"?

Q. 2 - Fill in the blanks:
"A class asserts that every objects created using it should have _____ (i.e., 'data') and _____ (i.e., 'operations')."

Q. 3 - Give two access modifiers.

Q. 4 - What, if any, is the difference between a parameter and an argument?

Q. 5 - Write a statement that creates a new object from the `Rectangle` class.

Q. 6 - What is the purpose of the keyword `new`?

Q. 7 - Do different objects from the same class share their instance variable?

Q. 8 - What does it mean to say that instance variables have a default initial value? How is that different from the variables we have been manipulating in the `Main` method?

Q. 9 - Suppose we have a `Circle` class containing

```
public void SetRadius(double RadiusArgument)
{
        radius = RadiusArgument;
}
```

Write a statement that create a `Circle` object, and one statement that sets its radius to 3.5.

Q. 10 - What does the keyword `return` do?

Q. 11 - Consider the following UML diagram:

Circle
- radius : float
+ setRadius(radiusParam : float) : void + getRadius(): float + getArea(): float

What is the name of the class, what are the methods and attributes of the class?

Q. 12 - Write a getter for an attribute of type `string` named `myName`.

Q. 13 - Write a setter for an attribute of type `int` named `myAge`.

Q. 14 - Is it possible to have more than one constructor defined for a class? If yes, how can C# know which one is called?

Q. 15 - What is the name of a constructor method? What is the return type of a constructor?

Q. 16 - Write a constructor for a `Soda` class with one `string` attribute called `name`.

Q. 17 - What is called the "default" constructor? Do we always have the possibility of using it?

Q. 18 - Consider the following partial class definition:

```
1  public class Book
2  {
3          private string title;
4          private string author;
5          private string publisher;
6          private int copiesSold;
7  }
```

1. Write a statement that would create a `Book` object.
2. Write a "getter" and a "setter" for the `title` attribute.
3. Write a constructor for the `Book` class taking at least one argument (you're free to decide which one(s)).

Q. 19 - Why would one want to define a constructor for a class?

Q. 20 - Write a get method for a `total` instance variable of type `int`.

Q. 21 - Assuming `name` is a `string` instance variable, there is problem with the following setter. Fix it.

```
public int SetNamel(string var){
        name = var;
}
```

Q. 22 - Draw the UML diagram of a class named "Student" with a single attribute, "name", of type `string`, and two methods, `SetName` and `GetName`.

Q. 23 - Briefly describe what a format specifier is. Write a statement that uses one.

Part II – Problems

There is only one problem this time, and it is harder than what you'll be asked to do during the exam. Being able to solve it is an excellent sign that you are ready.

In the meantime, focus on lab 9 and 10, that are challenging, and make sure you understand all the concepts we studied so far.

Problem 1

You are going to design a class named `Triangle`. A triangle has three angles, but knowing the value of only two angles is sufficient to determine the value of the third, since they always add up to 180°. Hence, it is sufficient to have only two `double` attributes, `angle1` and `angle2`. We want to define several methods:

- a no-arg constructor that sets the value of `angle1` to 60.0 and the value of `angle2` to 60.0,
- another constructor, that takes two arguments, and assigns to `angle1` the value of the first argument, and assigns to `angle2` the value of the second argument,
- getters for `angle1` and `angle2`,
- a method that computes and returns the value of the third angle, that we name `ComputeAngle3`,
- a method that rotate the triangle: the value of the first angle should become the value of the second angle, and the value of the second angle should become the value of the third angle.

1. Write the UML diagram for the `Triangle` class.
2. Write the full, compilable implementation of the `Triangle` class.