

Exam #1 will consist of questions taken or inspired from this homework, the two previous homework, and from the labs:

- For Section A (Tuesday), you should study the lab up to Week #6 (included), and this homework up to question 16 (included), as well as the problem (which is *difficult*),
- For Section B (Thursday), it is enough to stop at the lab of Week #5 (even if having a look at Week #6 will not hurt), and to stop at question 13 (included) in this homework, as well as the problem (which is *difficult*).

## Part I – Questions

1. What is “an instance of the class”?

2. Fill in the blanks:

“A class asserts that every objects created using it should have \_\_\_\_\_ (i.e., ‘data’) and \_\_\_\_\_ (i.e., ‘operations’).”

3. Give two access modifiers.

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

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

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

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

8. 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.

9. What does the keyword `return` do?

10. Write a get method for a total instance variable of type `int`.

11. Write a getter for an attribute of type `string` named `myName`.

12. Write a setter for an attribute of type `int` named `myAge`.

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

```
public int SetName1(string var){
    name = var;
}
```

14. Draw the UML diagram of a class named “Student” with a single attribute, “name”, of type `string`, and two methods, `SetName` and `GetName`.

15. Briefly describe what a format specifier is. Write a statement that uses one.

16. 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?

17. 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?

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

19. What is the name of a constructor method? What is the return type of a constructor?

20. Write a constructor for a `Soda` class with one `string` attribute called `name`.

21. What is called the “default” constructor? Do we always have the possibility of using it?

22. 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)).

23. Why would one want to define a constructor for a class?



## 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.

## 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  $180^\circ$ . 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.

