

# C# Class Layout

The order of members in a C# class should be in order of instantiation and/or initialization generally, as follows:

## Constants and Fields

- Constants should be all uppercase, with each word separated by an underscore
- Fields should be private and camelCase. While fields are sometimes necessary, using Properties instead is preferable. A property with only a getter is preferable to a readonly field.

```
const string SOME_STRING = "this string";
```

```
private string someString;
```

## Constructors and Destructors

- Constructors should be ordered by number of parameters
- We should chain constructors where appropriate
- Destructors, if needed, should be come below all the constructors

```
public MyObject() {}  
public MyObject(int id){  
    myId = id;  
}  
public MyObject(int id, string name): this(id) {  
    myName = name;  
}  
  
~MyObject {  
    //destructor code  
}
```

## Properties

- Should be PascalCase and should have getters and setters as appropriate
- Should have the necessary access level, private only if necessary

```
public MyClass(){  
    □ MyGuid = Guid.NewGuid();  
}  
  
public string MyString { get; set; }  
public Guid MyGuid { get; }  
public int Number1 { get;set; }  
public double Number2 { get; } = 5;  
public double MyRatio => Number1/Number2;  
public List<int> SomeNumbers { get; } = new List<int>();
```

## Methods

- Methods should be in PascalCase
- Can be one line return methods using => is applicable

```
public string MakeAString(string a, string b)  
{  
    □string c = $"{a}:{b}";  
  
    return c;  
}  
  
public string ANewString(string a, string b) => $"{a}/{b}";
```

## Event Handlers and other, as applicable

---

Revision #1

Created 3 November 2022 14:27:52 by Bryce Holloway

Updated 24 June 2024 05:55:04 by Bryce Holloway