

Classes

Classes are building blocks of software which encapsulate data and behavior.

Anatomy: Name, Fields, Methods

Declaration: `<> AccessModifier + class + ClassName`

Constructors: (code snippet: **ctor**)

- Constructors are methods that are called when an instance of a class is instantiated.

- Constructors have no return type.

- Constructors can be overloaded with different signatures.

```
<> public class ClassName() // default constructor
{ }

<> public class ClassName(int parameter) // overload
: this() //calls the default constructor
{ }
```

Functions/Methods

- Methods are the functions of classes.

- Methods can be overloaded with different signatures.

Declaration: `<> Access Parameter + Return Type + (Argument Type argument, ...) {}`

Varying number of parameters:

```
<> public void MethodName(params int[] arguments) {}
```

Calling a function/method:

```
returnValue = FunctionName(arguments)
returnValue = ObjectName.MethodName(arguments)
```

Conventions

PascalCase	classes, methods, properties
camelCase	method parameters
_camelCase	private fields

Access Modifiers

public	accessible everywhere
private	accessible only from inside the class
protected	
internal	
protected internal	

Usable for classes, methods and variables.

Objects

<code>var objectName = new ClassName();</code>	Create an object
<code>objectName.MethodName()</code>	Access public class methods
<code>var x = ObjectName.PropertyName()</code>	Gets an object property.
<code>ObjectName.PropertyName = x</code>	Sets an object property.

