VB notes P4:

OOP:

Variables inside class: attributes

Functions/Procedures inside class: methods

Class containing both variables(attributes) and functions/Procedures (methods) is called encapsulation.

In main when a variable of type class is declared, it is called an object

Public: the variables and functions that can be used in the class or outside the class or anywhere. Child, grandchild, main, other procedures, and functions, everyone can use.

Example:

Class A

Public name

End Class

Class B: Inherits A

Public age

End Class

Just to make you understand:

Reality of what B looks like:

Class B

Public name

Public age

End Class

Class B inherits name perfectly fine as name is public

Sub main()

Dim Book as New B

Console.writeline(Book.name && Book.age) ‘Works perfectly fine

End Sub

Private: These variables and functions can only be used inside the class. But outside the class they can’t be accessed. Even child classes cant use them, main cant use them

Class A

Private name

End Class

Class B: Inherits A

Public age

End Class

Just to make you understand:

Reality of what B looks like:

Class B

Public age

End Class

Class B can’t inherit name as name is private. Only A can use name

Sub main()

Dim Book as New B

Console.writeline(Book.name) ‘Gives error as name is a private attribute

End Sub

Maybe this is part of syllabus, maybe not:

Protected: These variables and functions cannot be used directly in main. But they can be used by a child class (but not grandchild class).

Class A

Protected name

End Class

Class B: Inherits A

Public age

End Class

Class C: Inherits B

Public Date

End Class

Just to make you understand:

Reality of what B looks like:

Class B

Public age

Private name

End Class

Just to make you understand:

Reality of what C looks like:

Class B

Public age

Public date

End Class

Class C can’t inherit name as name is protected in A, hence it is private in B. Only class A and class B can use name

Sub main()

Dim Book as New C

Console.writeline(Book.name) ‘Gives error, as name is a protected attribute in class A

End Sub

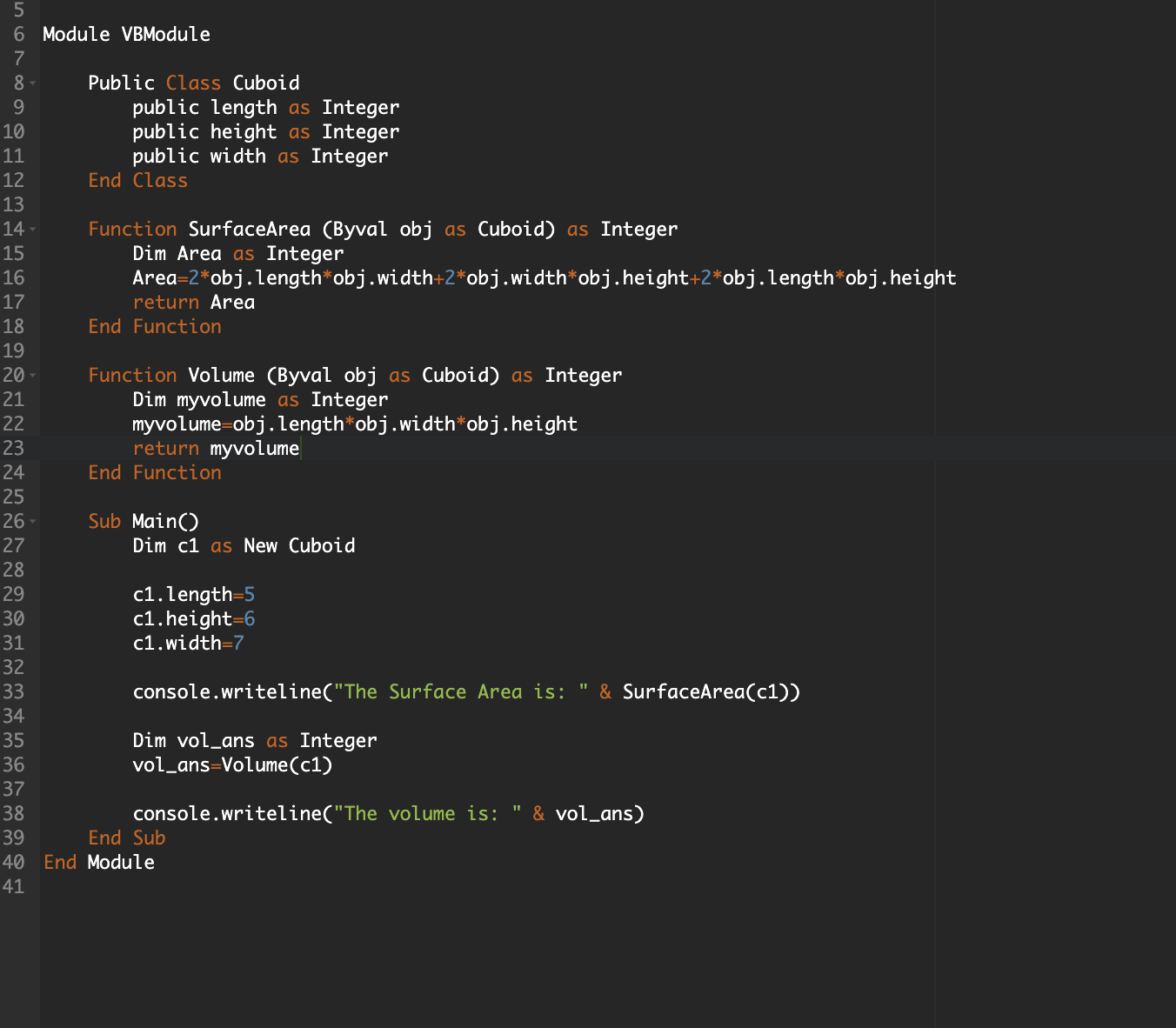
Class A:

Public name

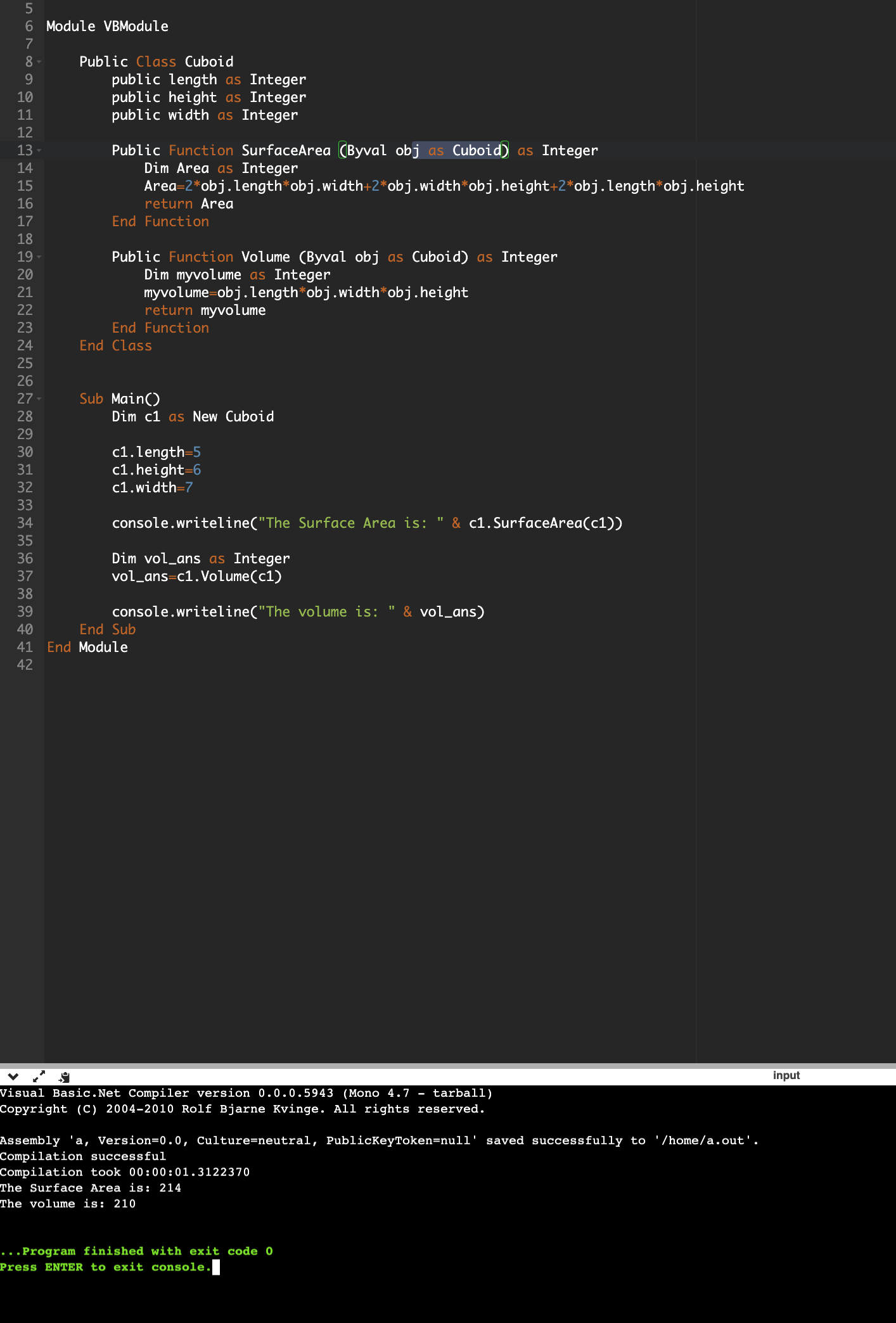
Class

Methods:

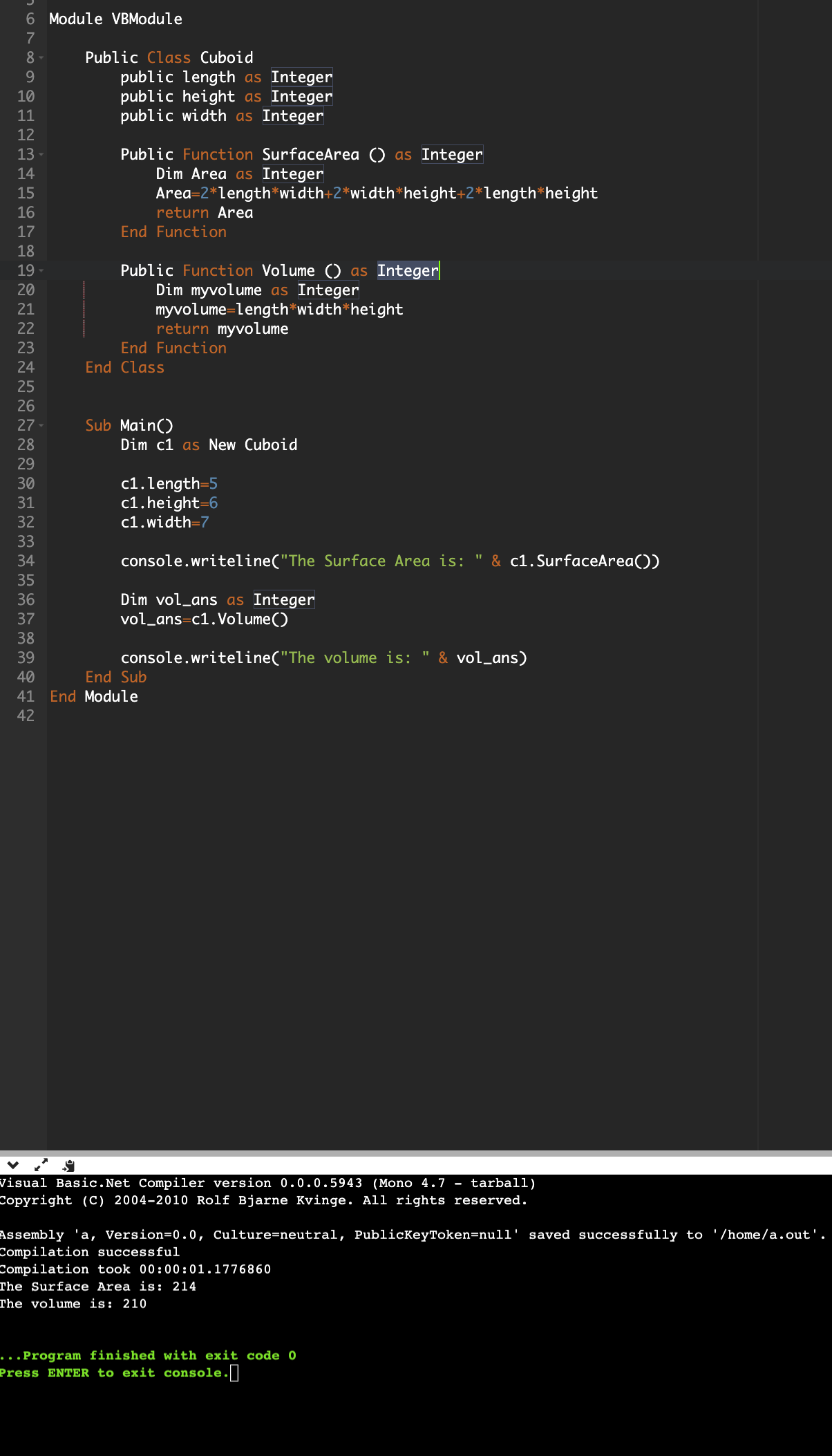
Functions outside the class (no methods)



Functions brought inside the class (methods): though this is not good practice



Next step: Best practice and you don’t have to do c1 a lot of times like “c1.SurfaceArea(c1)”



Constructors: A type of function/Procedure (method) inside the class that:

When you make an object (class variable). A constructor runs immediately.

A constructor runs at the start of the lifetime of an object basically

A constructor can be used to declare the object and initialize the attributes of the object.

So instead of

c1.length=5

c1.height=6

c1.width=7

to initialise

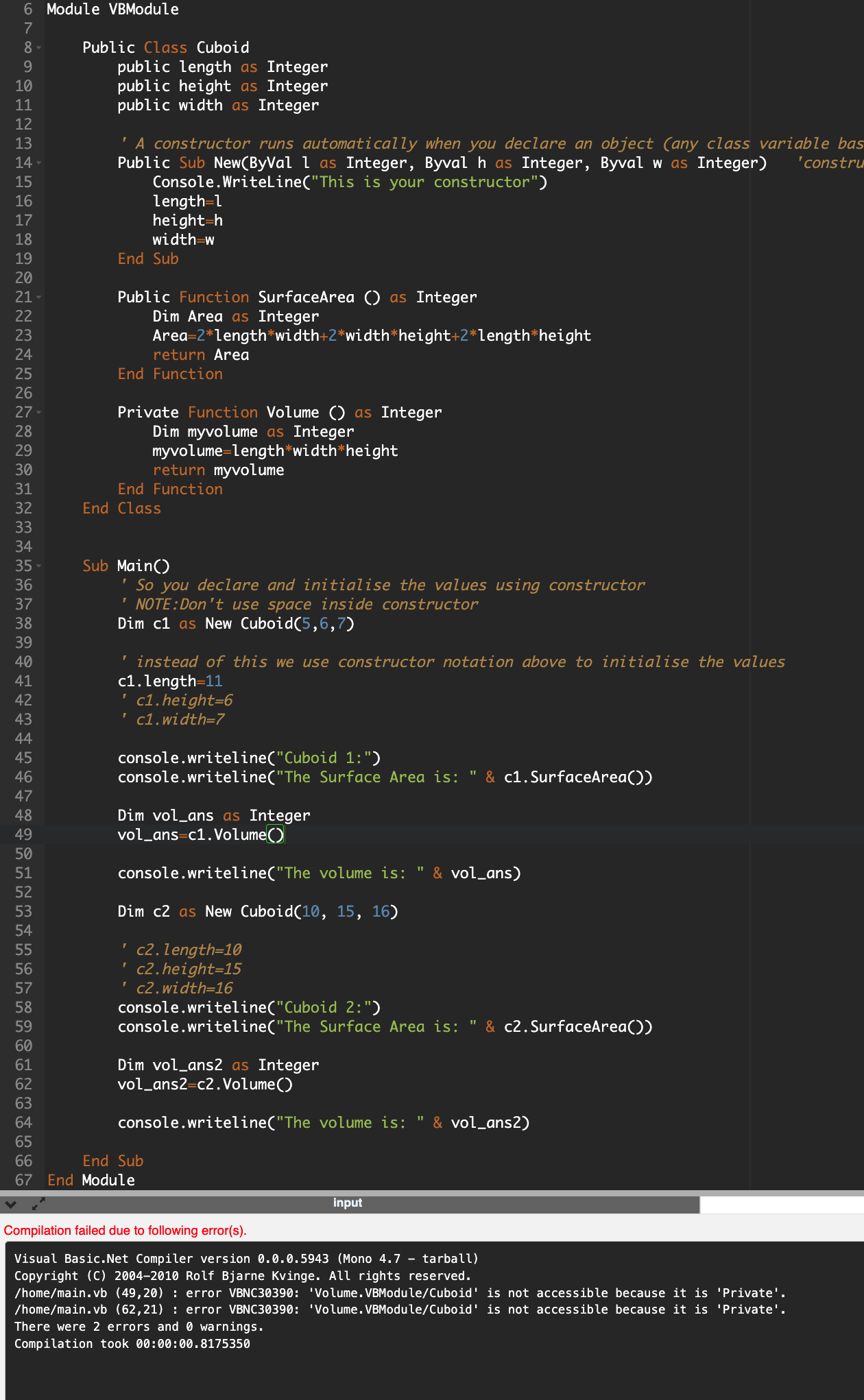
we can do

Dim c1 as New Cuboid(5,6,7)

You can find the rest of the code about constructor in the OOP constructor.vb file in the codes folder.

Notes on Public and private.

If we make the volume function in OOP constructor.vb as private. It cannot be used in main as it is private and it gives an error



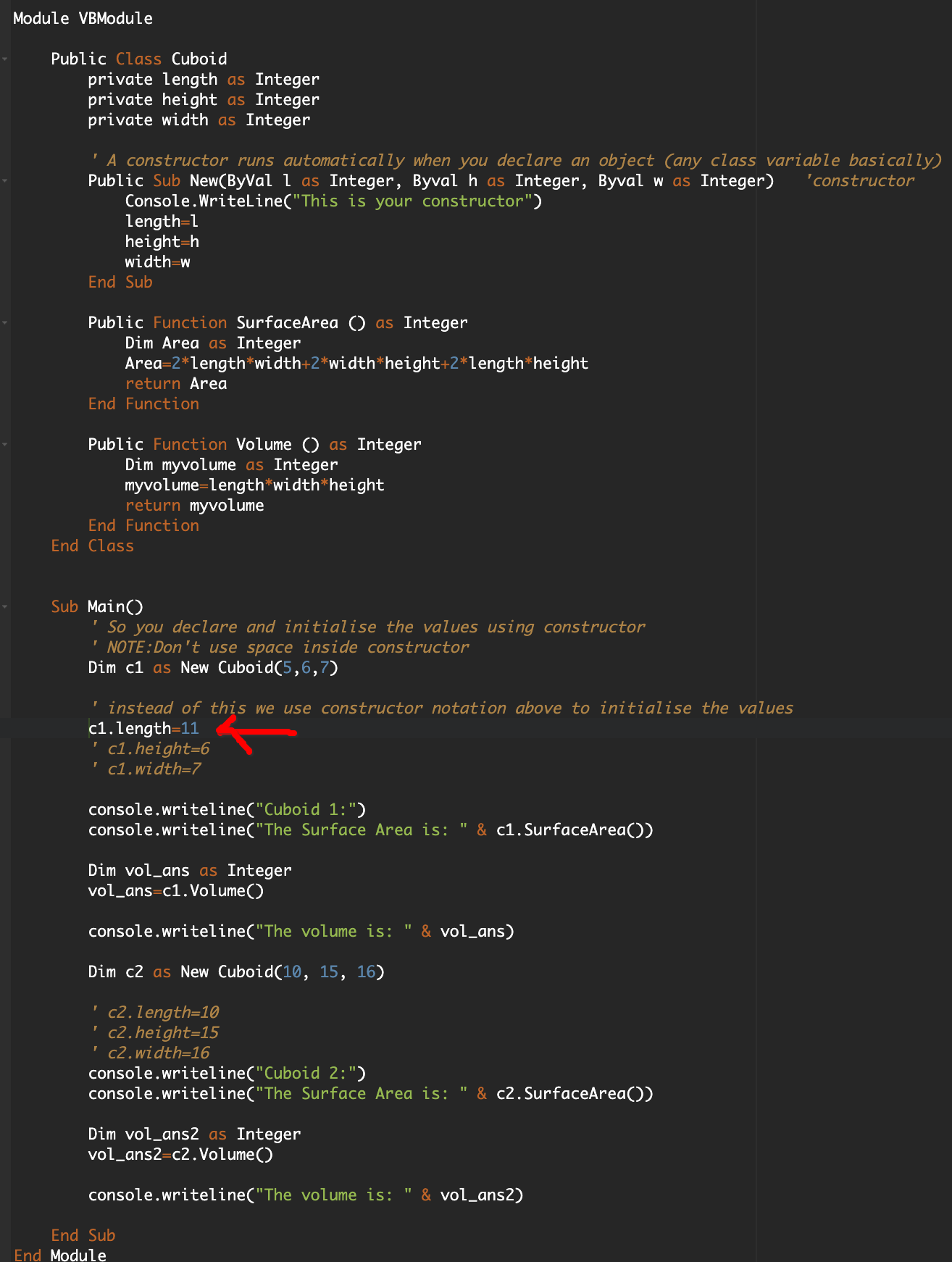
If we make the attributes as private



As the function is public, and the function is inside the class, that’s why the function can use the private variables

However

We can’t direct access private variables like this



This will give error:

Text

Description automatically generated

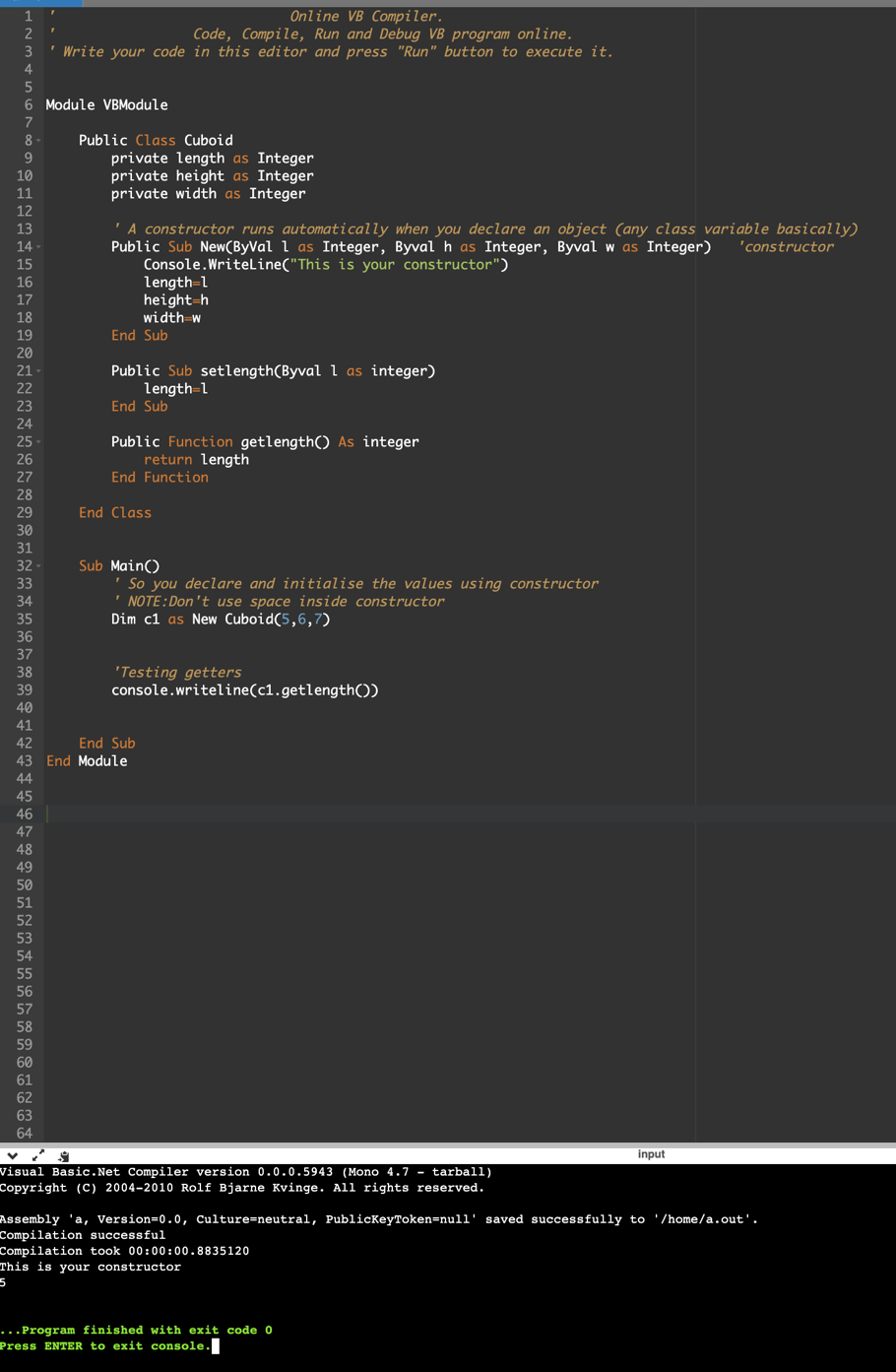
We can only access private variables indirectly (like in the case of the example or like in setters and getters)

Setters and getters

(can be used for both private and public attributes, but it makes most sense to use them for private attributes)

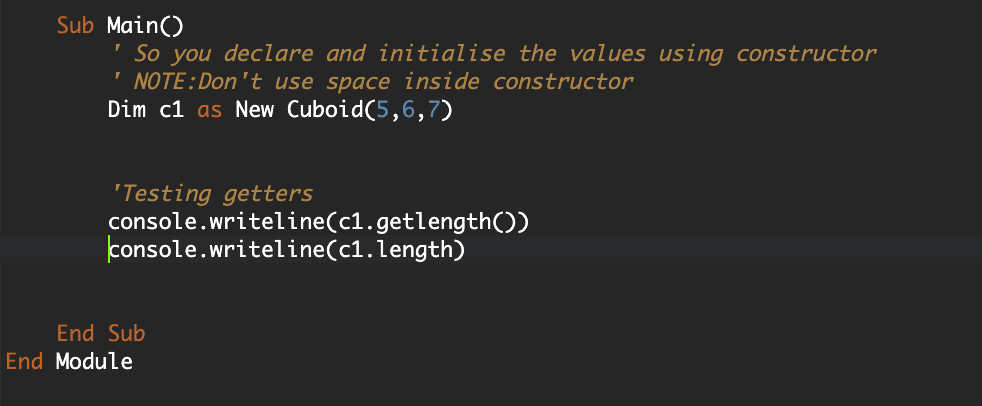
Getter:

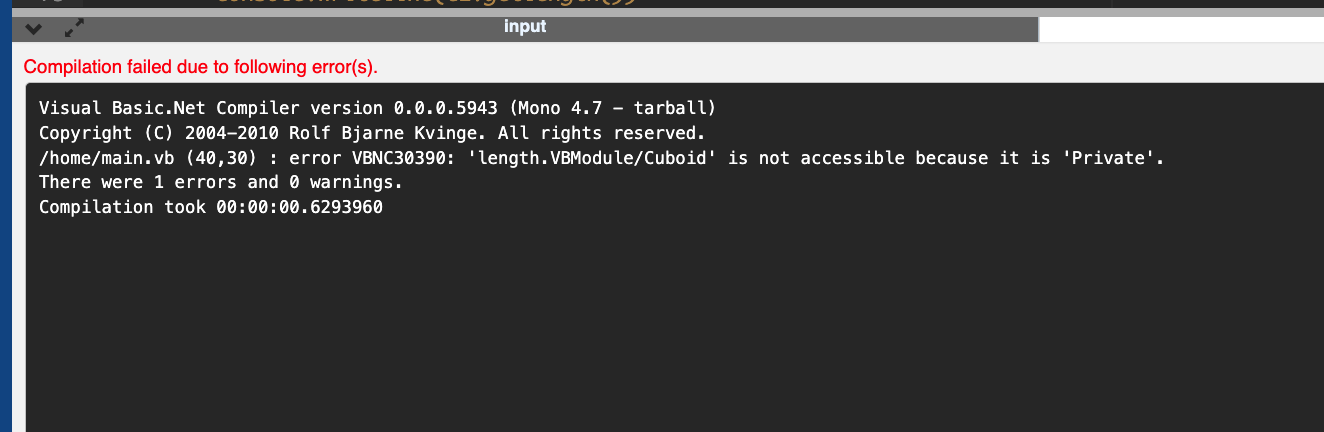
It is a function that returns the attribute



You can directly print the length using a getter even if the attribute is private

If we change main like the following and output c1.length aswell (even though it is the same as outputting the getter method). It will give error as we are accessing length directly and length is private.

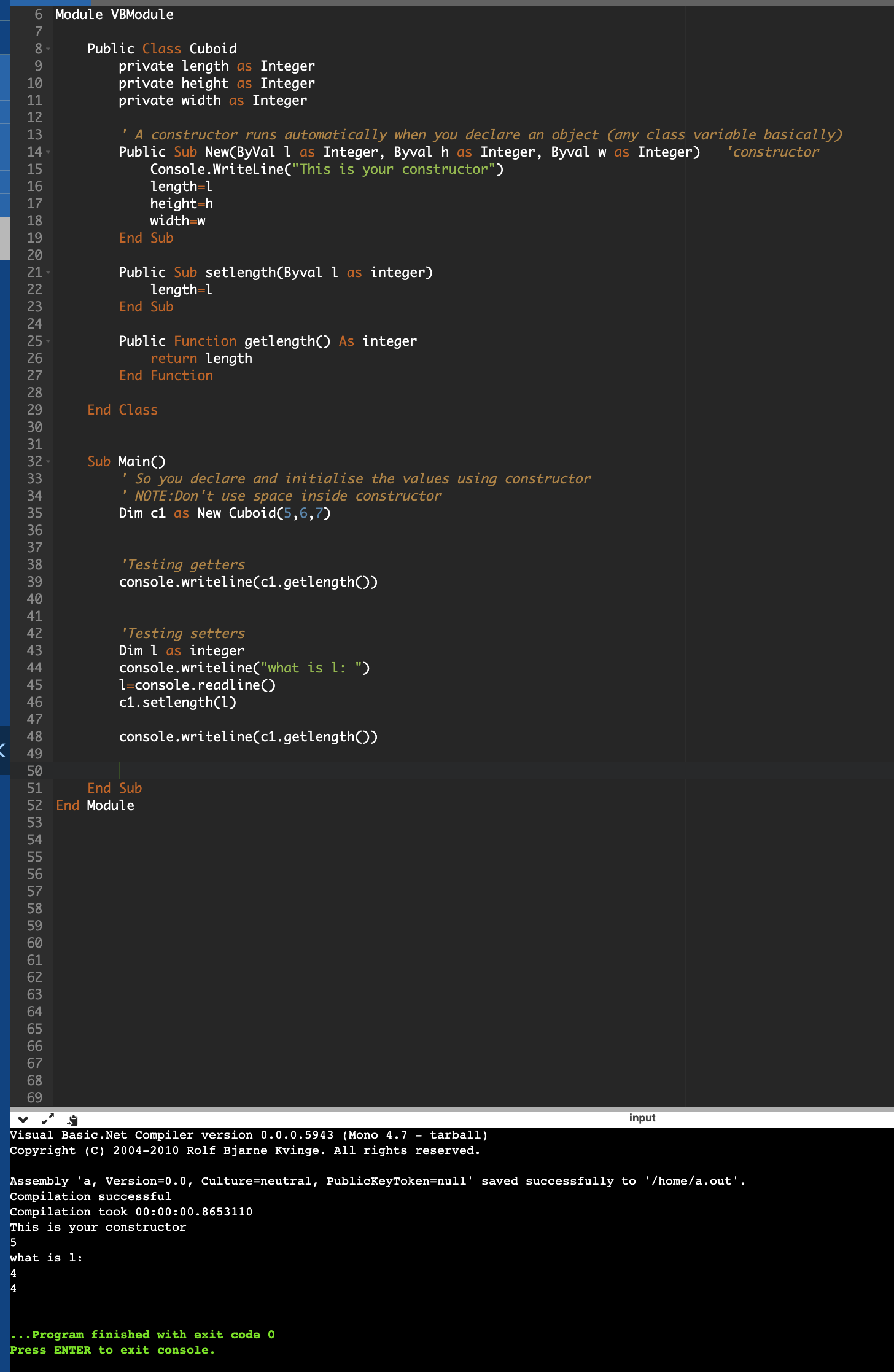




Setter:

It is a procedure that assigns the value to an attribute.

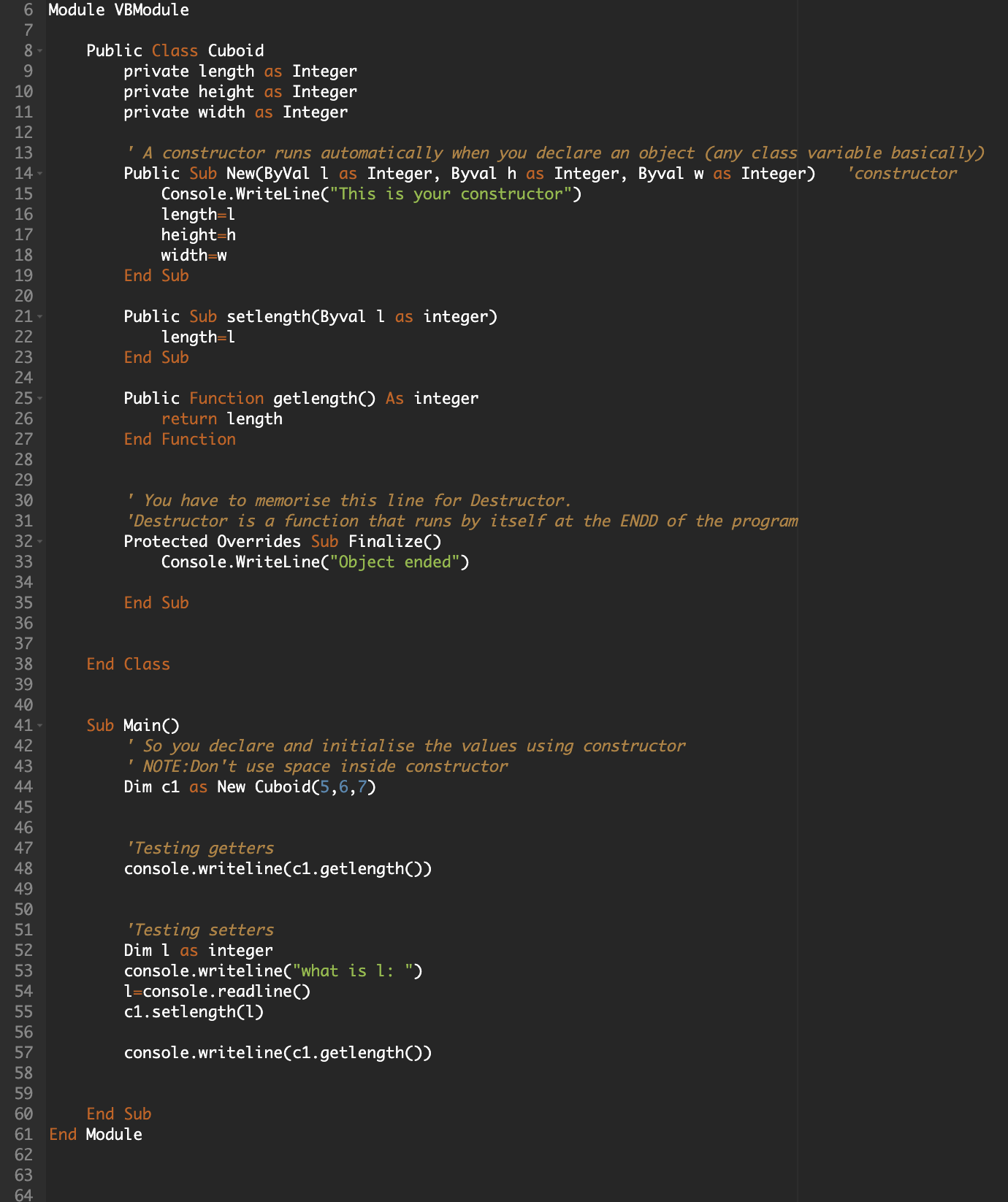
It can be used to change value of an attribute or change value of a private attribute.



As we see here we change length from 5 to 4 using a setter. (Even though length is private, this works)

Destructors:

These are a function that run at the very end of the program



As you can see, “Object ended” comes at the very end of the program

Text

Description automatically generated

Lets say you have 2 classes with destructors.

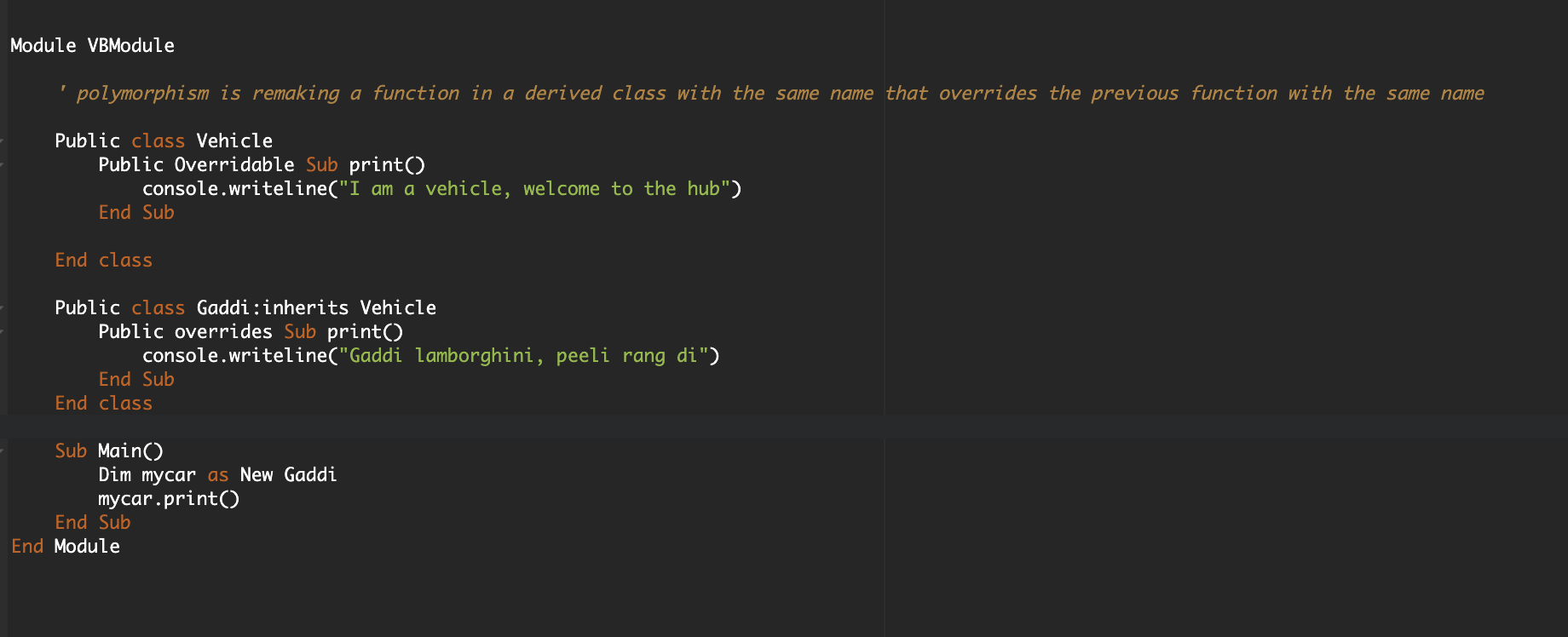
The object declared first runs its destructor at the end. The object declared second runs first

You can see the code in the destructor.vb file. It has the following output

Text

Description automatically generated

Polymorphism: Refer to note below



The child class’s function/procedure runs instead of the parent class as it is overwritten.

Also refer to Polymorphism.vb

Overloading: same name function/Procedure in one class. Depends on parameters

Refer to overloading.vb

Pointers: (P3 topic):not part of p4

