

Systems Development: Object Oriented Programming

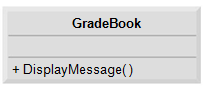
(H172 35)

Introducing Classes

GradeBook Walkthrough

Step 1 – declaring a class with a method and instantiating an object of a class

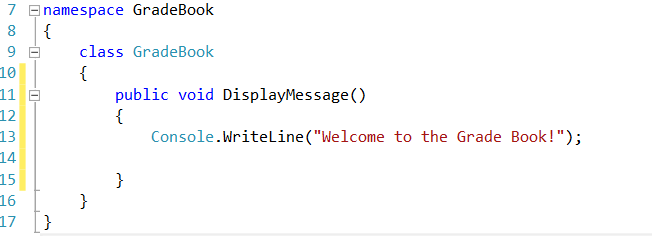
We are going to create a class named GradeBook (declared in file GradeBook.cs) which will be used to display a message on the screen welcoming the user to the GradeBook application Class. The UML diagram for the class is as follows:



Note: there are no attributes for this class so the middle section of the class UML diagram is empty. The bottom section contains the behaviour/operations of the class which correspond to methods in C#. The + sign in front of the operation name indicates that DisplayMessage is a public operation.

We are also going to create a GradeBookTest class (declared in the file GradeBookTest.cs), which is a testing class in which the Main method will create and use an object of class GradeBook. By convention, we declare classes GradeBook and GradeBookTest in separate files, such that each file’s name matches the name of the class it contains.

To start, select File > New Project... to open the New Project dialog, then create a GradeBook Console Application. Right-click on the project in Solution Explorer and Add->Class, naming the file GradeBook.cs. Update the GradeBook.cs code to the following:



Declaration of class GradeBook

class declaration begins on line 9, always contains keyword *class* followed by a name as a minimum.

Contains a DisplayMessage method, remember we need to create an object (instance) of this class and call its method to get line 13 to execute and display the message.

Declaration of method DisplayMessage

Method begins with keyword *public* to indicate the method is available “to the public” i.e. it can be called from outside the class declaration’s body by methods of other classes.

Keyword *void* is the method’s return type which indicates that this method will not return (i.e. give back) any information to its calling method when it completes its task.

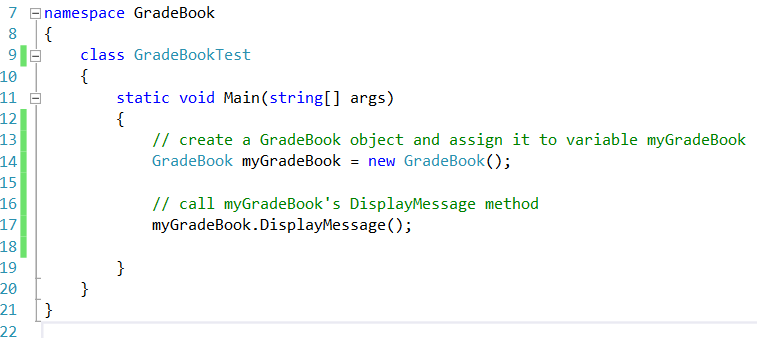
The method name follows the return type, generally methods are named as verbs while classes are named as nouns. The parentheses after the method name indicate that this is a method. An empty set of parentheses indicates that the method does not require additional information to perform its task.

Line 11 is commonly referred to as the method header.

Using class GradeBook:

Now we are ready to use class GradeBook in an application. For Console Applications, *Main* begins the execution and this can be found in the Program.cs file automatically created when the project is set up. Rename the Program.cs file to GradeBookTest.cs. Any class that contains a Main method can be used to execute a Console application.

Update the code in the GradeBookTest.cs file to the following:



Main Method

Main is where the Console Application begins execution. The *static* keyword indicates that Main is a static method. A static method is special because it can be called without first creating an object of the class (in this case, GradeBookTest) in which the method is declared.

Creating a GradeBook object

We would like to call class GradeBook’s DisplayMessage method to display the welcome message in the console window, but typically you cannot call a method that belongs to another class until you create an object of that class – line 14.

Each new class you create becomes a new *type* in C# that can be used to declare variables and create objects. New class types will be accessible to all classes in the same project.

Variable myGradeBook (line 14) is initialised with the result of the *object creation expression* newGradeBook(). The new operator created a new object of the class specified and attached parentheses represent a call to a constructor, which is similar to a method, but is used only at the time an object is created to initialise the object’s data (data can be placed inside the parentheses to specify initial values).

Calling the GradeBook object’s DisplayMessage method

We can now use myGradeBook to call its method DisplayMessage, line 17. This is done by using the variable, myGradeBook followed by the member access (.) operator, then the method name, DisplayMessage (with an empty set of parentheses as we are not required to pass any information). This causes DisplayMessage to perform its task.