# Basic Generic Class

**The goal for this exercise** is try creating a simple, generic class. For this exercise, you’ll be operating at a mostly ‘syntactic’ level.

Your job is to fill in the implementation of the BasicGeneric class so that the class can store a single object (be it a simple type or a reference type), print out that object (remember that *everything*, both simple & reference, has a ToString method).

As a hint, the code that you'll write should be really short – probably about 10-20 lines of code, including blank lines. The objective here is to get you up-and-running with generics, in a way that's suggestive of how they're typically used to store 'Collections' of things. The objective is not to have you doing incredibly intricate work, yet ☺

**What you need to do for this exercise:**

1. In the starter solution, in the **PCE\_Starter** project you'll find a class named BasicGeneric. You need to implement this class, as described below.

|  |
| --- |
| **BasicGeneric Data & Methods** |
| Data Field Name | Type | Description: |
| storedItem | Generic type;Assumed to be a simple/value type | **storedItem** will be the name of the **one,** **single** thing that this class holds. The thing that the class holds is assumed to be a simple type (a value type) – such as an int, or a double,etc.Yes, this is really contrived – you would almost never, in real life, create a class to hold a copy of single int (well, in most cases, in real life, anyways….) – the objective here isn't to store integers (and doubles, and floats, etc) , it's to get acquainted with what generics are. |
| Note: all data fields should be marked **private** |
| Method Name | Returns | Description/Parameters: |
| SetItem | Nothing | Parameters:1. An instance of the type specified for the overall class (this it the <T>, in the sample code)

Takes the parameter, and stores that value into the **storedItem** variable. This shouldn't need to return anything, since it should be impossible for it to fail (again, assuming that we're storing only value/simple types) |
| GetItem | storedItem | Parameters: NoneThis will return storedItem, whatever that value happens to be. |
| Print | Nothing | Parameters: NoneThis will call Console.WriteLine on **storedItem** . For this exercise, you're allowed to assume that Console.WriteLine will do something reasonable ; don't worry about the format of the output otherwise. |
| Note: all methods should be marked **public** |

1. There’s some test code provided to you in the Basic\_Generic\_Test\_Code class. Your code should work with the provided test code (i.e., the test code should compile and run with your completed code). Your code should cause the provided test code to produce the following output:

|  |
| --- |
| INT: Stored 100, got back:100DOUBLE: Stored 3.14159, got back:3.14159Stored "fun text here", got back:fun text here |