**Other Language report:**

Name: S M Ragib Rezwan

ID: 103172423

Object Oriented Program is a paradigm used by several languages including C#, C++, Java, Rust, Python, etc. But even though they use the same concepts to fulfill the OOP paradigms in their code, they end up doing so in slightly different ways. Thus, in this report, I will look into the different ways C# and Python fulfill the OOP paradigms.

**Comparison between the languages:**

|  |  |  |
| --- | --- | --- |
|  | C# | Python |
| Paradigm it follows | Object Oriented only | Object Oriented and Procedural |
| Unwanted data removal from memory | Automatic garbage collector | Automatic garbage collector |
| Syntax comparison | Need to declare variables before using them  Must end every line with semicolon | No need to declare variables before using them  No need to end every line with semicolon |
| Language type | Statically-typed, compiled (build) | Dynamically interpreted |
| Code reading aspect | Consistent syntax | Human readable, indented (ie the whitespaces) |
| Time taken to learn | Slower | Faster |

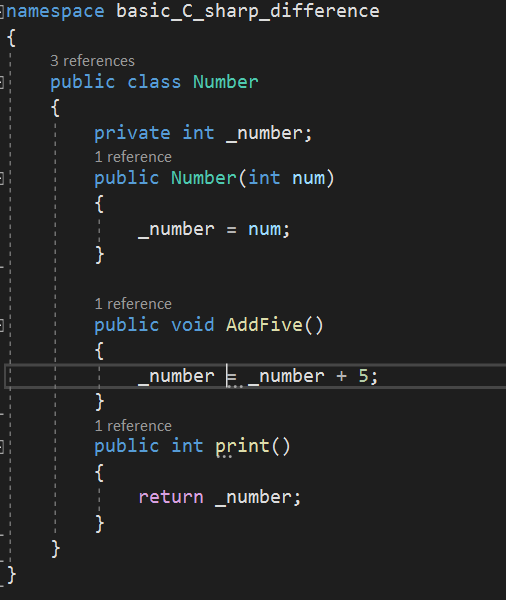
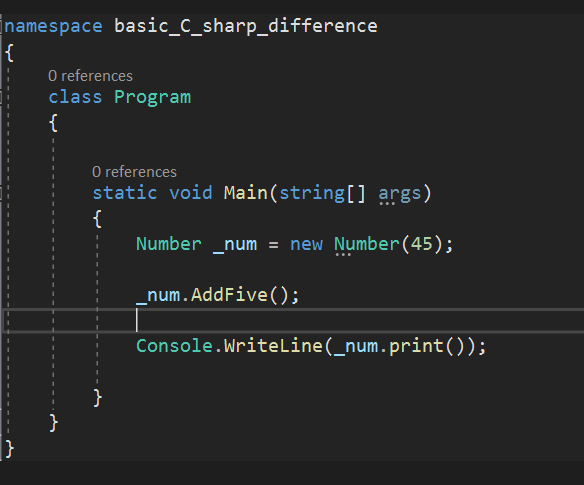
**Key difference between them:**

After considering all the possible differences, the core difference that stands out between them is the type of language they are!

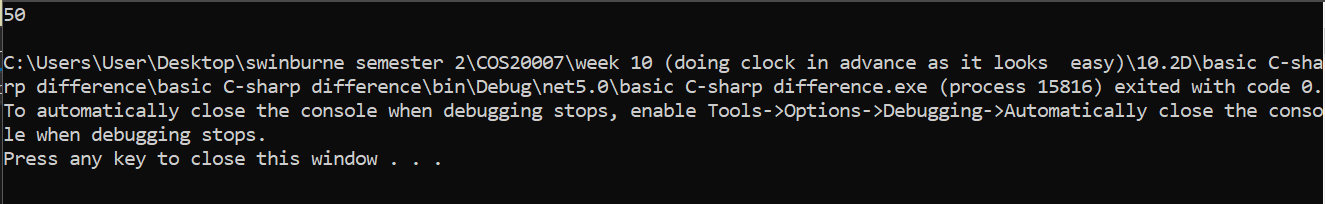
**C#:**

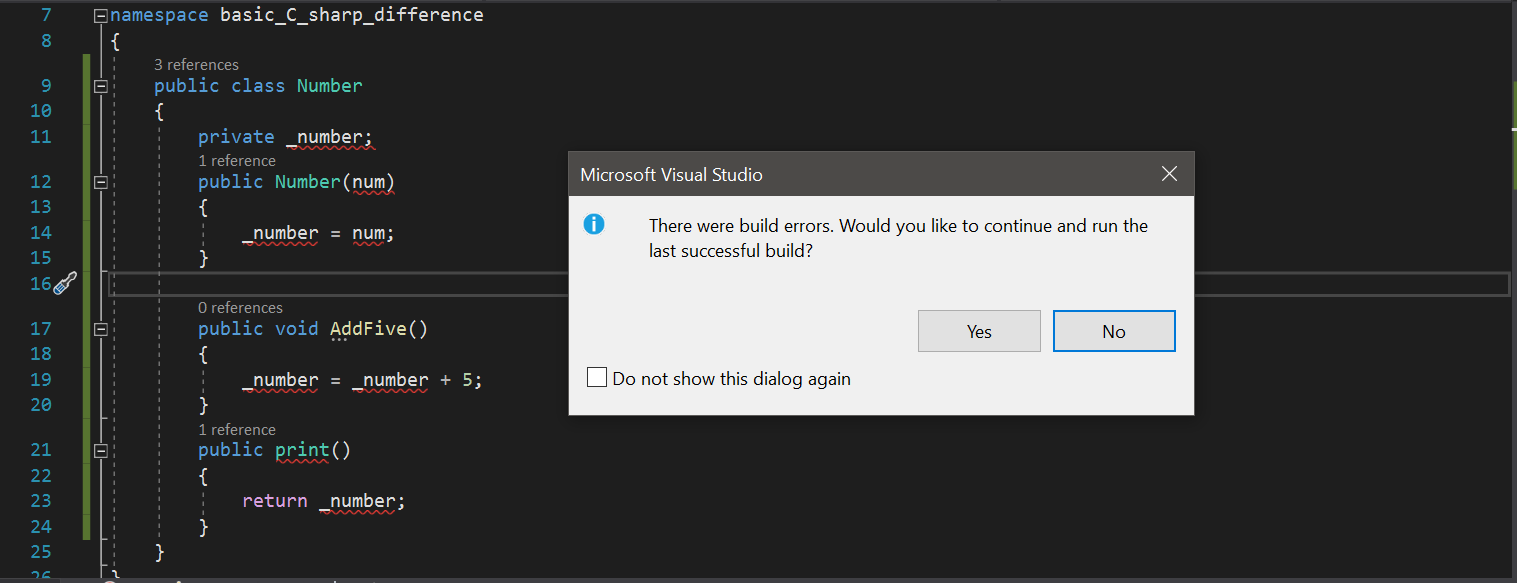
 C# is a statically typed language. This means in order for us to use a variable, we must first declare what type of data it can accept. This variable can be in the form of parameter in the field, a parameter passed to a method or class constructor, a return variable, etc.

Otherwise, the program will return an error at compile time and stop the code from compiling!

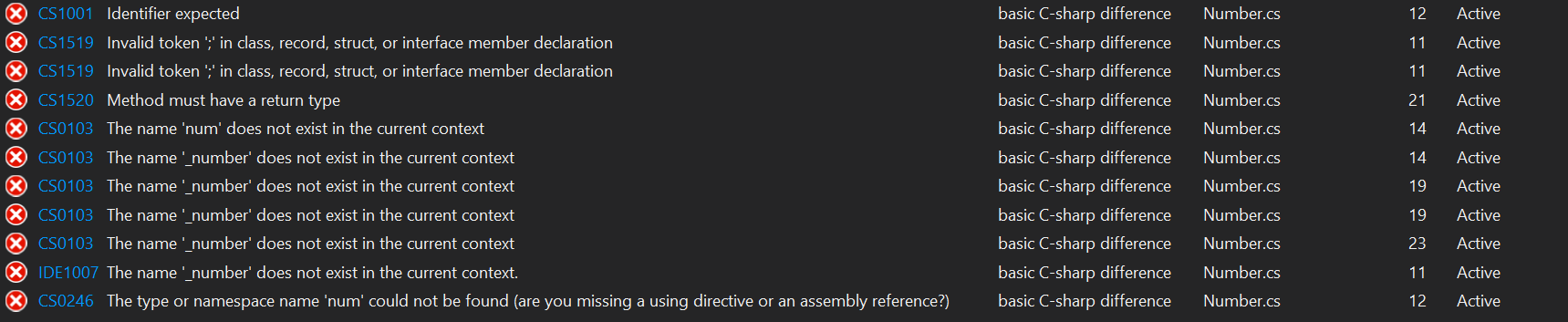


(When we declare the datatype of the variable declared in field, datatype of the variable given as the parameter to the method and constructor, datatype of return variable, etc before using it, it compiles and runs properly)



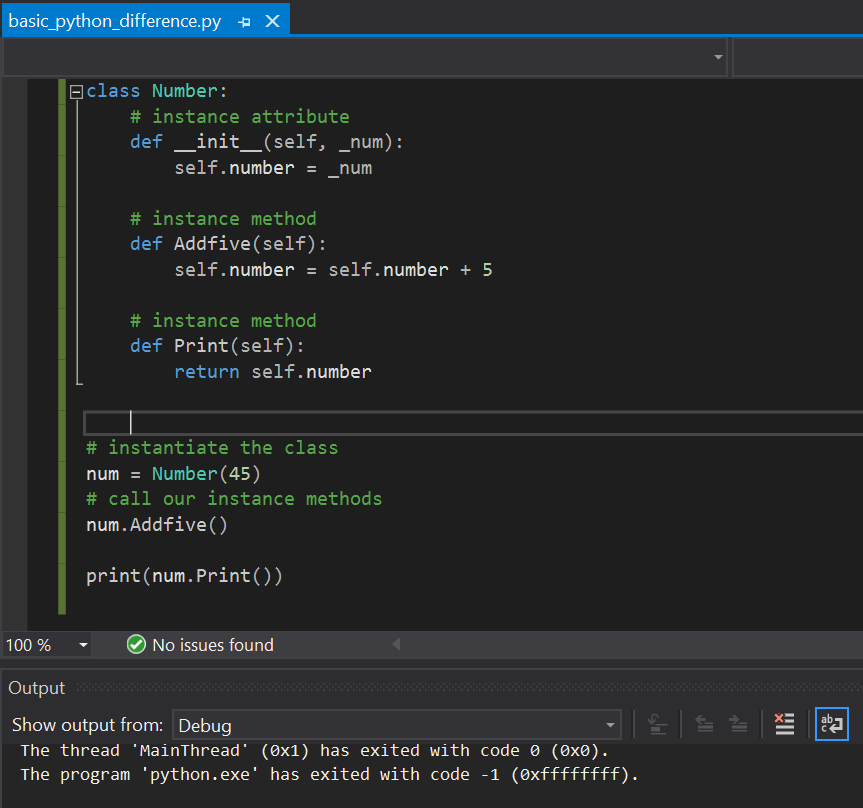
(When we don’t declare the type, the code doesn’t even compile)

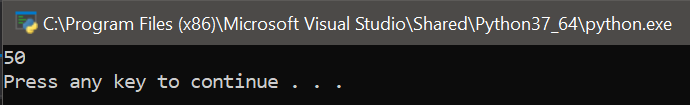
(and instead, it shows the following errors where it doesn’t know what number is and thus can’t do anything using it)

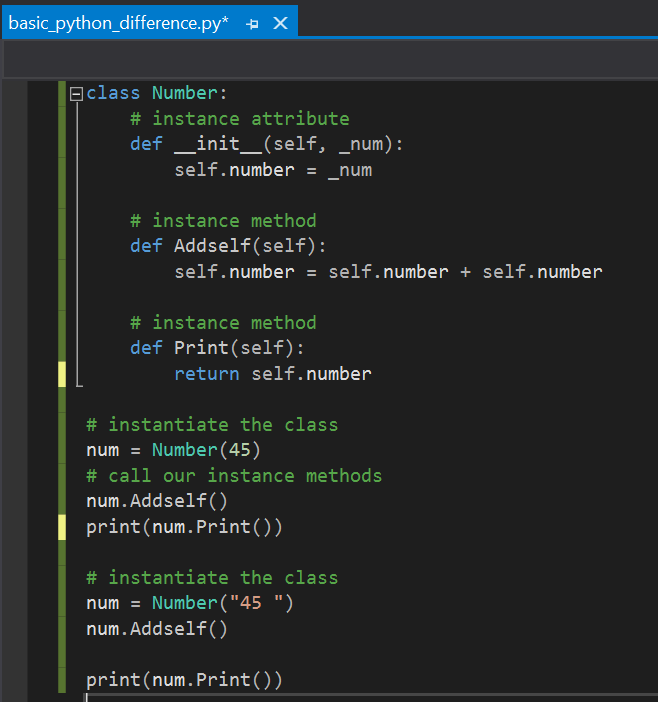


**PYTHON:**

But python is a dynamically typed language. This means we don’t have to declare a variable’s type before using it. Instead we can directly use the variables for our own purposes!

Furthermore, even if there is an error in type, it will still compile the code with the error (as it relies on inbuilt interpreter) and will only detect it during run time!

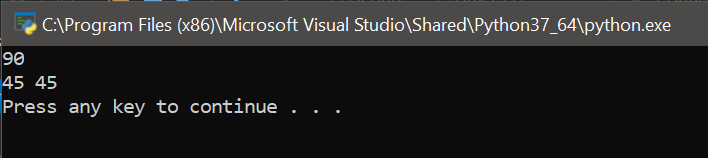
(Although we didn’t mention the datatype anywhere, the code still ran as intended and 45 has been assigned to the number, 5 can be added to it and result can be printed out)

Although this can lead to logical errors in code (as the same variable can be used to store different datatypes), it can also be exploited to create a dynamic class that does different things, depending on the type of data it receives, providing more flexibility to the code!

(here I have altered the code to make it add itself instead of adding 5.

Now in 1st case, it is taking in a number and adding itself making 45 + 45 = 90.

In the 2nd case, it is taking in a string and concatenating itself to it, forming “45 “ + “45 “ which gives “45 45 “ as output in print!)



**Conclusion:**

Although there are several other differences between them, this point is the main one. That is because no matter how big or small, complex or simple the code is; it will have to have at least one variable to store data. Thus, knowing whether or not type needs to be declared for a variable, before assigning data to it, is absolutely crucial!

So, I end the report with a slight modification on Shakespeare’s famous quote:

**To declare or not to declare, that is the question**

Reference:

<https://hackr.io/blog/c-sharp-vs-python>

<https://stackoverflow.com/questions/41259203/does-python-garbage-collection-automatic-in-python>

<https://www.poetryfoundation.org/poems/56965/speech-to-be-or-not-to-be-that-is-the-question>

<https://www.netguru.com/blog/python-vs-c-sharp>