C# vs Python

In Object Oriented Programming

Distinction Task 10.2

2021

Written by Marella Morad

Student ID: 103076428

Introduction

Programming languages are basically the way that we communicate with computers, giving them instructions to how to do a task and then asking them to complete it. Just like the languages we speak, there aren't two languages that are the exact same, there are always difference, some could be minor, and others major differences. This report will focus on two programming languages, C#, since it's the core language in this course, and a language that has some key differences with C#. This language is Python. There will be a comparison between the two languages in relation to main language features, but more specifically, into how the two languages apply object-oriented programming principles.

Main Language Features

C#

C-Sharp is a **statically typed** language, developed by Microsoft, Inc. It first appeared in the year 2000 and is now one of the most known object-oriented programming languages. This is mainly because when C# was first developed, it was based on OOP principles, which make applying these principles using C# considerably easy.

C# supports language **interoperability**, which means that it can access code written in any .NET compliant language and can also inherit classes written in such languages, since C# itself is a .NET language (C# Corner 2021). In addition, C# is a **structured language**, allowing the developer to break code down into functions, which improves the efficiency of the program given that code duplication can be basically replaced by a call of a function. It also features a **rich library**, providing the developer with many built-in functions, also increasing the speed of the program (JavaTPoint 2011).

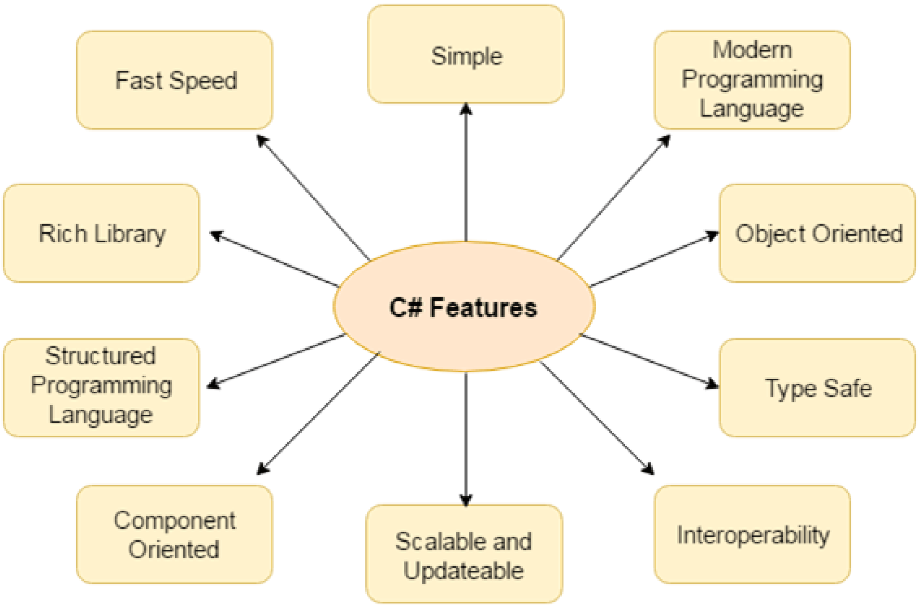

Figure 1 Top 10 C# language features (JavaTPoint 2011)

Python

Python is a dynamically typed, interpreted, general-purpose language, which supports Object-Oriented programming. It developed by Python Software Foundation in early 1991. However, in 2000, Python Consortium members signed an agreement for an Open-Source License, allowing developers to modify/improve on the language (Python.org 2020).

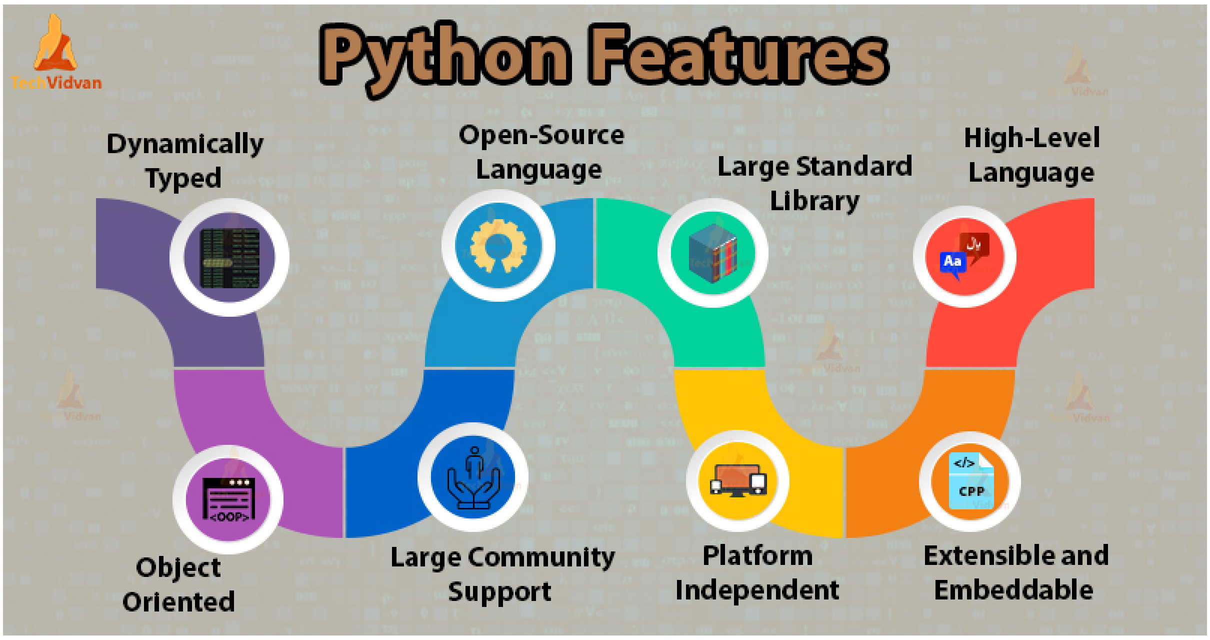

Figure 2 Python Language Features (TechVidvan 2019)

Python is an **Extensible and Embeddable** language, which means it can implement code from other languages, and can also be embedded into other language's code. A feature that sets Python aside is being an **open-source** language. Being an open source, free language, gives it the advantage of being constantly improved by the community and not only relying on improvements/bug fixes from the original developer, like Microsoft in C#'s case. Moreover, Python is a **platform independent** language, meaning that the same program will run on multiple platforms such as Windows, Linux, and Mac OS. It features a **large standard library** providing developers with many build-in functions.

Key Differences

After the brief introduction to each of the two languages, now we will discuss some key differences that are related to OOP implementation in both languages.

Static vs Dynamic Languages

The first main difference between Python and C# is that Python is a dynamically typed language, while C# is statically typed. The main differences arise as Statically typed languages are better with error handling, since the code is first compiled, and error checked before it runs. C# specifically has a built-in error detector that points out the specific lines that caused an error and will not compile in case there are errors in the code. Compiler errors include either not specifying the data type of a variable, or incorrectly specifying the data type (for example, declaring an int variable and assigning it to a string). On the other hand, dynamically typed languages, can run multiple functions simultaneously, resulting in a more difficult approach to locating and fixing errors, but a simpler way of writing the actual code (Friedman 2019). As mentioned earlier, Python is an interpreted language, and not a compiled language, which is why detecting errors before runtime does not happen in Python. Also, variable types are determined during runtime, no need for type declaration in Python (Simon Bonello 2021).

Syntax

The first obvious syntax difference between the two languages is that C# implements namespaces, classes, and functions to run the code. Any C# program has a Main() function which is where the order of the code run is defined, only functions/methods called in Main will run when the program is executed. Looking at a simple program in both languages in Table 1, we can see that python is much more efficient in terms of lines of code.

Table 1 A simple program in C# vs Python

C#	Python

using System;
class Program
{
  public static void Main (string[]
args)
  {
    int age = 21;
    string name = "Marella";
    Console.WriteLine(name + " is " +
age + " years old");
  }
}
	age = 21
	name = "Marella"
	print(name + " is " + str(age)
	+ " years old")

Another main difference is that in Python, indentations are a must, while in C# they are only used for clarity, and are replaced by curly brackets. Additionally, C# and Python are very different in terms of syntax. As mentioned earlier, in Python, there is no need to declare a variable before using it, however, in C#, you must declare the variable correctly first, only then you'll be able to use it (see Table 1).

OOP Differences

The syntax differences discussed in the previous section will have a huge impact on the implementation of OOP principle in the two languages. In this report, I will only cover encapsulation differences.

Encapsulation

One of the main OOP principles is **encapsulation**, which means that the user can only access and change the fields that are relevant to their action (Schults 2018). For example, setting the name of a character in a game is only done once, and the user does not need to change it. Therefore, the name property of the character class should be a read-only (getter) property, that will only return the character's name. In C#, encapsulation is done very easily through **properties** that can be read-only, write-only, or even read and write properties (see Table 2). These properties set rules for user's interactions with the class's private fields. On the other hand, Python has a different implementation of properties, which function more like a method than a property compared to C# syntax. As shown in Table 2, the user can access the private field __name, by calling the name method. Unlike C#, we cannot have a getter and setter (read and write) property in Python, they need to be two different methods (see `age_getter(self)` and `age_setter(self, age)` in Table 2).

Sample Code Illustrating The Difference

Table 2 Encapsulation in C# vs Python

C#	Python

using System;
 	
class Program
{
  public static void Main (string[]
args)
  {
    Dog buddy = new Dog("Buddy", 9);
    Console.WriteLine(buddy.Name);
    Console.WriteLine(buddy.Age);
    buddy.Age = 15;
    Console.WriteLine(buddy.Age);
  }
}
 	
class Dog
	class Dog:
	  def __init__(self, name, age):
	    self.__name = name
	    self.__age = age
	
	  def name(self):
	    return self.__name
	
	  def age_getter(self):
	    return self.__age
	
	  def age_setter(self, age):
	    self.__age = age
	
	buddy = Dog("Buddy", 9)
	print(buddy.name())

```
{
  private string _name;
  private int _age;

  public Dog(string name, int age)
  {
    _name = name;
    _age = age;
  }

  public string Name
  {
    get
    {
      return _name;
    }
  }

  public int Age
  {
    get
    {
      return _age;
    }
    set
    {
      _age = value;
    }
  }
}
```

```
print(buddy.age_getter())
buddy.age_setter(15)
print(buddy.age_getter())
```

Reference List

C# Corner 2021, “Top 10 Most Important Features Of C#,” *C-sharpcorner.com*, viewed 14 October, 2021, <https://www.c-sharpcorner.com/article/top-10-most-important-features-of-C-Sharp-programming/>.

Friedman, J 2019, “C# vs Python: Two Object-Oriented Languages With Key Differences,” *C# Station*, viewed 14 October, 2021, <https://csharp-station.com/c-vs-python/>.

JavaTPoint 2011, “C# Features - javatpoint,” *www.javatpoint.com*, viewed 14 October, 2021, <https://www.javatpoint.com/csharp-features>.

Python Tutorial 2020, “Understanding Python Encapsulation Clearly By Practical Examples,” *Python Tutorial - Master Python Programming For Beginners from Scratch*, viewed 14 October, 2021, <https://www.pythontutorial.net/python-oop/python-private-attributes/>.

Python.org 2020, “Welcome to Python.org,” *Python.org*, Python.org, viewed 14 October, 2021, <https://www.python.org/search/?q=open&page=1>.

Schults, C 2018, “C# Features: An Exhaustive List of the Best Ones - NDepend,” *NDepend*, viewed 14 October, 2021, <https://blog.ndepend.com/c-features-best-ones/>.

Simon Bonello 2021, “C# vs Python. Which language is best for you? - Chubby Developer,” *Chubby Developer*, viewed 14 October, 2021, <https://www.chubbydeveloper.com/csharp-vs-python/>.

TechVidvan 2019, “Features of Python - Explore the essence of Python - TechVidvan,” *TechVidvan*, viewed 14 October, 2021, <https://techvidvan.com/tutorials/features-of-python/>.