Higher Diploma in Science in Data Analytics

Multi Paradigm Programming - Shop Assignment

Karolina Szafran-Belzowska, G00376368@gmit.ie

The goal of this report is to compare the shop solution for three different programming languages: C, Python Object-Oriented Language and Python Procedural Language.

The program simulates the Shop and process customer's orders.

The Shop has a stock of the items which are available or not available and also has current float (cash).

The customer can purchase items by reading customer data from CSV file or by live mode. In this the order is typed by the customer. The program checks if the product is available or if the customer has enough money. Every successful purchase should be updated, namely: stock and cash.

A **multi-paradigm programming** language is a programming language that supports more than one programming paradigm. The design goal of such languages is to allow programmers to use the most suitable programming style and associated language constructs for a given job. Languages such as C, Java and Python are multi-paradigm programming languages that support object-oriented programming to a greater or lesser degree, typically in combination with imperative, procedural programming.[1]

Languages can be classified into multiple paradigms.[2]

Common programming paradigms include:

1. ***imperative*** in which the programmer instructs the machine how to change its state:
   - procedural which groups instructions into procedures,
   - object-oriented which groups instructions with the part of the state they operate on,
2. ***declarative*** in which the programmer merely declares properties of the desired result, but not how to compute it:
   - functional in which the desired result is declared as the value of a series of function applications,
   - logic in which the desired result is declared as the answer to a question about a system of facts and rules,
   - mathematical in which the desired result is declared as the solution of an optimization problem.

Languages that fall into the imperative paradigm have two main features: they state the order in which operations occur, with constructs that explicitly control that order, and they allow side effects, and then later read at a different point in time inside a different unit of code. The communication between the units of code is not explicit.

Meanwhile, in object-oriented programming, code is organized into objects that contain a state that is only modified by the code that is part of the object.

To make programs simpler for a human to read and write, imperative statements can be grouped into sections known as code blocks.[3]

The blocks of code are known as procedures. Once a procedure is defined, it can be used as a single imperative statement, abstracting the control flow of a program.

The process allows the developer to express programming ideas more naturally. This type of imperative programming is called procedural programming, and it is a step towards higher-level abstractions such as declarative programming.

In procedural programming, the code is organized by using procedures and/or functions, while object orientated programming is based on objects which use methods and functions.

Examples of imperative programming languages:

- C
- C++
- Java
- Julia
- Python
- Ruby
- MATLAB
- Pascal
- and much more

C Procedural Programming Language

The C programming language is a programming language developed by Dennis Ritchie[4] at Bell Labs in 1972 from an almost unknown language named B.

The first major software written in C was the Unix operating system, and for many years, C was considered to be inextricably linked with Unix.[5]

It was designed to be compiled to provide low-level access to memory and language constructs that map efficiently to machine instructions, all with minimal runtime support.

Despite its low-level capabilities, the language was designed to encourage cross-platform programming.[6]

Characteristics of the C programming language:

- It has a small, fixed number of keywords, including a full set of control flow primitives: if/else, for, do/while, while, and switch.
- More than one assignment may be performed in a single statement,
- Data typing is static, but weakly enforced,
- C has no "define" keyword; instead, a statement beginning with the name of a type is taken as a declaration. There is no "function" keyword; instead, a function is indicated by the presence of a parenthesized argument list.
- Complex functionality such as I/O, string manipulation, and mathematical functions are consistently delegated to library routines.
- Heterogeneous aggregate data types (struct) allow related data elements to be accessed and assigned as a unit.
- Low-level access to computer memory is possible by converting machine addresses to typed pointers.
- Procedures are a special case of function, with an untyped return type void.
- There is a basic form of modularity: files can be compiled separately and linked together, with control over which functions and data objects are visible to other files via static and extern attributes.

Many later languages have borrowed directly or indirectly from C, including C++, C#, Unix's C shell, D, Go, Java, JavaScript (including transpilers), Julia, Limbo, LPC, Objective-C, Perl, PHP, Python, Ruby, Rust, Swift, Verilog and SystemVerilog.

The first line of the program contains a preprocessing directive, indicated by #include. This causes the compiler to replace that line with the entire text of the stdio.h standard header, which contains declarations for standard input and output functions such as printf and scanf. The angle brackets surrounding stdio.h indicate that stdio.h is located using a search strategy that prefers headers provided with the compiler to other headers having the same name, as opposed to double quotes which typically include local or project-specific header files.

int main(void) - it indicates that a function named main is being defined. The main function serves a special purpose in C programs; the run-time environment calls the main function to begin program execution. The type specifier int indicates that the value that is returned to the invoker as a result of evaluating the main function, is an integer.

The keyword void as a parameter list indicates that this function takes no arguments.

The opening curly brace indicates the beginning of the definition of the main function.

The printf format string refers to a control parameter used by a class of functions in the input/output libraries of C and many other programming languages. It is the name of one of the main C output functions, and stands for "print formatted".

The keyword struct is to define variables of structure type. To define a structure, the user/programmer must use the struct statement.

The struct statement defines a new data type, with more than one member.

Python Programming Language

Python has been acclaimed as one of the easiest programming languages to learn for many years and it was one of the most trending & hottest programming languages in 2019.[7]

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.[8]

Python was created in the late 1980s, and first released in 1991, by Guido van Rossum[9] as a successor to the ABC programming language. Python 2.0, released in 2000,

introduced new features, such as list comprehensions, and a garbage collection system with reference counting, and was discontinued with version 2.7 in 2020.

As of December 2020 Python ranked third in TIOBE's index of most popular programming languages, behind C and Java.[10]

Features of Python:[11]

1. Easy to code. Python is a high-level programming language and very easy to learn. It is also a developer-friendly language.
2. Free and Open Source. Python language is freely available at the official website. It is recommended to download Anaconda which constitutes the distribution of the Python and R programming languages for scientific computing with the goal of simplifying package management and deployment. The distribution includes data science packages suitable for Windows, Linux and macOS.
3. Object-Oriented Language. One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation, etc.
4. GUI Programming Support. Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python.
5. High-Level Language. Python is a high-level language. When we write programs in python, we do not need to remember the system architecture.
6. Extensible feature. Python is a Extensible language. We can write some Python code into C or C++ language and also we can compile that code in C/C++ language.
7. Python is Portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.
8. Python is Integrated language. You can easily integrated python with other languages like C, C++, etc.
9. Interpreted Language. Code is executed line by line at a time. There is no need to compile python code this makes it easier to debug our code.
10. Large Standard Library. It provides a rich set of module and functions so you do not have to write your own code for every single thing.
11. Dynamically Typed Language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature you do not need to specify the type of variable.

How to run Python script in Command Prompt:

1. Assuming the interpreter is installed as an executable program on your system, the most platform-neutral way to start an interactive interpreter sessiion is usually type python at your operating system's prompt, without any arguments.
2. Type in the word python, or python3 if you have both versions, followed by the path to your script.

The difference between procedural and object oriented programming language

Both Procedural Oriented Programming (POP) and Object Oriented Programming (OOP) are the high level languages in programming world and are widely used in development of applications.

A procedural program is typically a list of instructions that execute one after the other starting from the top of the line. On the other hand, object-oriented programs are built around well objects. Object-oriented programming has several advantages over procedural programming, which is the programming style you most likely first studied. Object-oriented programming enables you to develop large, modular programs that can instantly expand over time. Object-oriented programs hide the implementation from the end-user.

An object has characteristics and actions. Characteristics have a specific name. They are called **attributes**, and the actions are called **methods**. The color, size, style, and the price would be called attributes in the shoe example.

There are fundamental terms in object-oriented programming:

- Class – methods and attributes
- Object – an instance of a class. It can help to think of objects as something in the real world like a yellow pencil or small dog. Objects can be more abstract.
- Attribute – a descriptor or characteristic. Example could be a color, length or size. These attributes can take on specific values like blue, 3 cm or small.
- Method – an action that a class or object could take.

Python use **__init__** to create a specific object. The **self** variable saves attributes like color, size, making those attributes available throughout in the certain class. The word self is just a convention and it should be always the word self rather than some other word.

Python is considered as an object-oriented programming language rather than a procedural programming language. There are Python packages built with object-oriented programming such as **Scikit-learn**, **pandas**, and **NumPy**.[12]

Difference between Procedural Programming and Object Oriented Programming:

PROCEDURAL ORIENTED PROGRAMMING	OBJECT ORIENTED PROGRAMMING
In procedural programming, program is divided into small parts called ***functions***.	In object oriented programming, program is divided into small parts called ***objects***.
Procedural programming follows ***top down approach***.	Object oriented programming follows ***bottom up approach***.
There is no access specifier in procedural programming.	Object oriented programming have access specifiers like private, public, protected etc.
Adding new data and function is not easy.	Adding new data and function is easy.
Procedural programming does not have any proper way for hiding data so it is ***less secure***.	Object oriented programming provides data hiding so it is ***more secure***.
In procedural programming, overloading is not possible.	Overloading is possible in object oriented programming.
In procedural programming, function is more important than data.	In object oriented programming, data is more important than function.
Procedural programming is based on ***unreal world***.	Object oriented programming is based on ***real world***.
Examples: C, FORTRAN, Pascal, Basic etc.	Examples: C++, Java, Python, C# etc.

Image 1: Difference between Procedural Programming and Object Oriented Programming, taken from: https://www.geeksforgeeks.org/differences-between-procedural-and-object-oriented-programming/, 27/12/2020

Difference between C and Python Programming Language

Both C and Python are the majorly used programming languages. It is various characteristics and features that makes them popular in programming world for application development. The main difference between C and Python is that, C is a structure oriented programming language while Python is an object oriented programming language. In general, C is used for developing hardware

operable applications, and python is used as a general purpose programming language. C language is run under a compiler, python on the other hand is run under an interpreter. Python has fully formed built-in and pre-defined library functions, but C has only few built-in functions. Python is easy to learn and implement, whereas C needs deeper understanding to program and implement.

C	PYTHON
An Imperative programming model is basically followed by C.	An object-oriented programming model is basically followed by Python.
Variables are declared in C.	Python has no declaration.
C doesn't have native OOP.	Python has OOP which is a part of language.
Pointers are available in C language.	No pointers functionality is available in Python.
C is a compiled language.	Python is an interpreted language.
There is a limited number of built-in functions available in C.	There is a large library of built-in functions in Python.
Implementation of data structures requires its functions to be explicitly implemented.	It is easy to implement data structures in Python with built-in insert, append functions.
C is compiled direct to machine code which is executed directly by the CPU	Python is firstly compiled to a byte-code and then it is interpreted by a large C program.
Declaring of variable type in C is necessary condition.	There is no need to declare a type of variable in Python.
C does not have complex data structures.	Python has some complex data structures.
C is statically typed.	Python is dynamically typed.
Syntax of C is harder than python because of which programmers prefer to use python instead of C	It is easy to learn, write and read Python programs than C.
C programs are saved with .c extension.	Python programs are saved by .py extension.

Image 2: Difference between C and Python Programming Languages, taken from: https://www.geeksforgeeks.org/difference-between-c-and-python/, 27/12/2020

C and Python languages are similar yet have many key differences. These languages are useful languages to develop various applications. The difference both is that python is a multi-paradigm language and C is a structured programming language. Python is a general-purpose language that is used for machine learning, natural language processing, web development and many more. C is

mainly used for hardware-related application development such as operating systems and network drivers.

Shop in C and Python

The shop program is presented in a single file for C (shop.c). Code could easily become untidy and difficult to maintain if the shop program was much bigger which is a disadvantage of C. Struct in C can provide a structure to store related pieces of information e.g. Product struct – name and price. C struct is like what a class is in Python and contains different information such as the shops products stock or customer.

The shop in Python (OOP) uses classes and objects. It is also presented in a single files called python_oop.py. This model is based on the idea that programs will be organized around data, or objects. When creating the shop in Python OOP each class has its own file which makes it easier to pull information and update method or information relating to that class. For example class Product contains a name and a price of the product. This class object allows me to access the different attributes as well as to instantiate new objects of that class. I used the self parameter inside the class in Python Object-Oriented Language. This is because, whenever an object calls its method, the object itself is passed as the first argument. The __init__ function gets called whenever a new object of any class is instantiated. I have at least two functions in each class: __init__ and __repr__. First is to initialize the variables and second to display the number properly.

In Python Procedural Language the Shop (shop.py) was broken into functions, for example: def read_customer. The program is calling functions but the last one which is called def main() calls just live_mode. Each function carries out one and only one job.

I used CSV files to get more informations from arrays about customer and stock in the Shop.

Overall, I found Python Procedural much easier to use and get my head around. Python OOP seemed to be more difficult for me to implement.

References:

[1] https://slideplayer.com/slide/2811690/, 27/12/2020

[2] https://en.wikipedia.org/wiki/Programming_paradigm, 22/12/2020

[3] https://www.computerhope.com/jargon/i/imp-programming.htm, 22/12/2020

[4] https://en.wikipedia.org/wiki/Dennis_Ritchie

[5] https://www.computerhope.com/jargon/c/c.htm, 27/12/2020

[6] https://en.wikipedia.org/wiki/C_(programming_language), 27/12/2020

[7] https://blog.eduonix.com/featured/2019-survey-stackoverflow-know/, 27/12/2020

[8] https://en.wikipedia.org/wiki/Python_(programming_language), 27/12/2020

[9] https://en.wikipedia.org/wiki/Guido_van_Rossum, 27/12/2020

[10] https://www.tiobe.com/tiobe-index/, 27/12/2020

[11] https://www.geeksforgeeks.org/python-features/, 27/12/2020

[12] https://towardsdatascience.com/python-procedural-or-object-oriented-programming-42c66a008676, 27/12/2020

[13] https://www.educba.com/c-vs-python/, 27/12/2020

Also reviewed:

Dr Dominic Carr – Lecture materials : https://learnonline.gmit.ie/course/view.php?id=1902

https://docs.python.org

https://github.com/realpython

https://www.stavros.io/tutorials/python/

https://codecondo.com/10-ways-to-learn-python/

https://www.linuxjournal.com/article/3946

http://www.trytoprogram.com/python-programming/

https://python-3-patterns-idioms-test.readthedocs.io/en/latest/

https://www.programiz.com/c-programming

https://www.tutorialspoint.com/cprogramming/index.htm

https://www.learn-c.org/

https://www.youtube.com/watch?v=iT_553vTyzI

https://www.geeksforgeeks.org/c-programming-language/

https://www.tutorialspoint.com/differences-between-procedural-and-object-oriented-programming

https://www.tutorialspoint.com/difference-between-c-and-python