REPORT FOR GPA CALCULATOR ASSIGNMENT – Christopher Collins - [G00387822@gmit.ie](mailto:G00387822@gmit.ie)

Contents

[An Overview Of How The Python OOP & Python Procedural Program Worked 2](#_Toc80164624)

[Converting From OOP to Different Python Programming Style 2](#_Toc80164625)

[The Object Oriented Approach 3](#_Toc80164626)

[Difference 1 - Encapsulation 3](#_Toc80164627)

[Difference 2 – Inheritance 3](#_Toc80164628)

[Difference 3 – Polymorphism 4](#_Toc80164629)

[Difference 4 – Abstraction 4](#_Toc80164630)

[Functions Within Class 4](#_Toc80164631)

[5](#_Toc80164632)

[The Procedural Approach 5](#_Toc80164633)

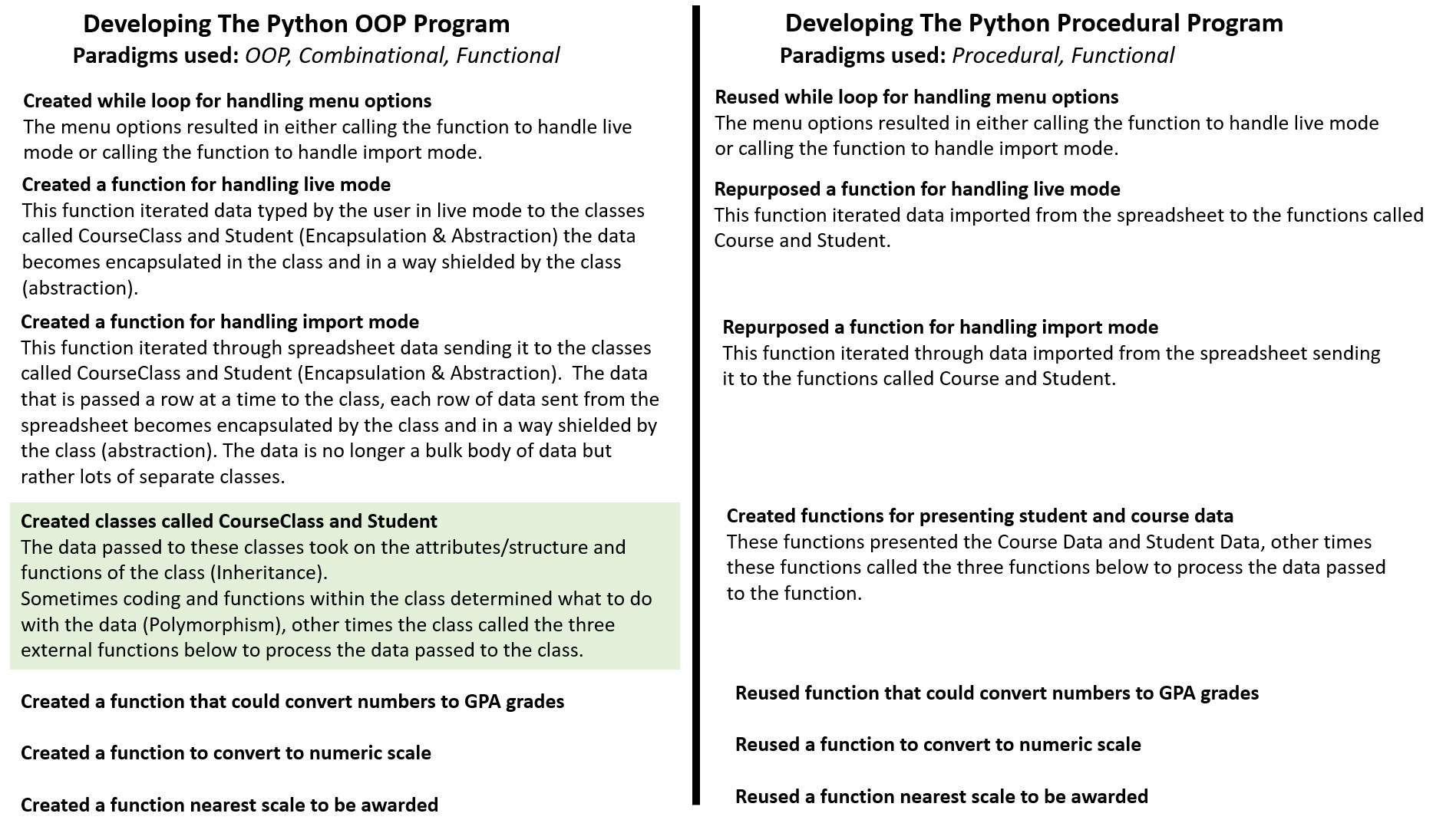
[Is It Procedural Or Is It Functional? 5](#_Toc80164634)

[Why I failed to create a working C program. 6](#_Toc80164635)

[My attempt at C 6](#_Toc80164636)

[Similarities in forming a Class in Python and a Struct in C 8](#_Toc80164637)

# An Overview Of How The Python OOP & Python Procedural Program Worked



*\*Both Python programs can do GPA conversions, convert to numeric scale and determine nearest scale to be awarded. The work has been submitted as Jupyter Notebook so you can see the run results.*

# Converting From OOP to Different Python Programming Style

In the Object Oriented Programming which I first created the Classes would have attributes / variables which could store lists and strings, the Classes could contain functions or call external functions to manipulate the data passed to the class.

By making small adjustment to the syntax I was able to convert the class to a function, and it basically did the same thing returning and printing data to the screen. The main difference being is that the data manipulated by the function didn’t gain any characteristics of being class data, it wouldn’t be possible at a later stage to use a variable name and extend it by a class attribute or function.

Such as Course.NameOfCourse or Course. Subjects

# The Object Oriented Approach

**See file:** GPA\_Calculator-CLASS-fixed.ipynb

In trying to achieve the Object Oriented Approach, I reviewed the principles of Object Oriented programming.

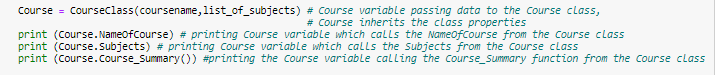
**Encapsulation, Abstraction, Inheritance, and Polymorphism**.

## Difference 1 - Encapsulation

In short ‘Encapsulation can be viewed as a shield that protects data from getting accessed by outside code.’ [[1]](#footnote-1)

In trying to protect the ‘Course’ and ‘Student’ data in my program being accessible by outside code I tried to iterate through the course and student data in the CSV and pass it directly into the Course Class and Student Class.

## Difference 2 – Inheritance



The Course variable passes data to the CourseClass

Course variable inherits the attributes of CourseClass

CourseClass has attributes NameOfCourse, Subjects and a function called Course\_Summary

When the variable Course is used it can use any of the inherited attributes and functions of the CourseClass

So Course.NameOfCourse is able to pull the name of course from the data contained within the Course class.

I could have created many courses in this program which could have inherited the same attributes and functions of the class, by doing coding similar to

Course1 = CourseClass(coursename1 , list\_of\_subjects1)

Course2 = CourseClass(coursename2 , list\_of\_subjects2)

Course3 = CourseClass(coursename3 , list\_of\_subjects3)

Each time the CourseClass Class is used the variable would inherit attributes and functions of the class.

So it would be possible to do **Course1.NameOfCourse** to get name of Course 1,

or **Course2.Subjects** to get a list of subjects for Course 2, also call a function of the CourseClass by doing Course3.**CourseSummary()**.

Any course inheriting every attribute and function of the class CourseClass.

## Difference 3 – Polymorphism

polymorphism means having many forms. In programming, polymorphism means the same function name (but different signatures) being used for different types.[[2]](#footnote-2)

I am not sure that I achieved Polymorphism in my attempt at OOP programming.

Polymorphism I believe would have required me to handle the GPA Conversions using class functions, instead I went for a combinational programming style where I called normal functions to handle GPA Conversions of data within the class. This meant I was able to reuse the normal functions again in my procedural Python program.

Some examples of Python Polymorphism class programming can be found on this blog <https://www.geeksforgeeks.org/polymorphism-in-python/>

## Difference 4 – Abstraction

By using Classes in Python I achieved Abstraction.

It is a lot harder to read Python code that uses Class methods.

Abstraction in Java and Python is a programming methodology in which details of the programming codes are hidden away from the user, and only the essential things are displayed to the user. Abstraction is concerned with ideas rather than events. It’s like a user running a program (Web Browser) without seeing the background codes. Abstraction is achieved in either Abstract classes or interface in Java and Python. NetBeans and Eclipse IDE implements abstraction for Java while Django implements abstraction for Python.[[3]](#footnote-3)

### Functions Within Class

I believe that a typical OOP approach would encourage functions as much as possible to be done within a class, whereas sometimes due to my lack of experience with working with class syntax I would call external functions from the class.

I did create a function within the CourseClass called Course\_Summary

## 

# The Procedural Approach

Even though the assignment required procedural programming in both C and Python, the syntax in C is much more difficult than Python. I was only able to achieve a working Procedural Program in Python **(see attached file GPA\_Calculator-PROCEDURAL-PYTHON-fixed.ipynb)** and a working OOP Program in Python.

## Is It Procedural Or Is It Functional?

Because Python supports multiple paradigms of programming styles, not just OOP and Procedural, it appears that I may have made an error and created a program that some programmers could describe more as a functional program rather than as a procedural program.

‘A functional programming is identical to procedural programming in which global variables are *not* being used.’[[4]](#footnote-4)

I felt the second version of my Python Program satisfied the Procedural definition below.

Procedural: Tasks are **treated as step-by-step iterations where common tasks are placed in functions that are called as needed**. This coding style favors iteration, sequencing, selection, and modularization. Python excels in implementing this particular paradigm.[[5]](#footnote-5)

There may have been coding nuances which were missed that might identify it more as a functional program than as a procedural program.

# Why I failed to create a working C program.

When trying to program in C I hit the stumbling blocks that programmers commonly encounter.

1. ‘It is compulsory to declare the variable type in C’[[6]](#footnote-6)
2. ‘Pointers available in C’, in Python ‘No pointers functionality available’[[7]](#footnote-7)
3. In C‘Implementing data structures required its functions to be explicitly implemented’ whereas Python ‘Gives ease of implementing data structures with built-in insert, append functions’[[8]](#footnote-8)

# My attempt at C

Here is an example of my work in C in preparing the GPA calculator:

// C program

#include <stdio.h>

int convert\_to\_alpha(double num) {

double result;

char grade;

printf("value of num : %f \n", num);

if (num >= 99.6)

grade = 'A+' ;

if (num >= 94.08 && num < 99.6)

grade = 'A' ;

if (num >= 88.2 && num < 94.08)

grade = 'A-';

if (num >= 82.32 && num < 88.2)

grade = 'B+';

if (num >= 76.44 && num < 82.32)

grade = 'B';

// and so on

else

grade = 'Z';

return grade;

}

int main() {

double num;

char convert;

num = 95.00;

printf("value of num : %f \n", num);

convert = convert\_to\_alpha(num);

printf("value of convert : %c \n", convert);

return 0;

}

I tried to understand what was wrong with my code. I couldn’t set the grade to be a two character variable such as 'A+' , I could only put 'A'

Below is advice that I got back from a more experienced C programmer who reviewed my code.

*The issue was due to the use of "char" variable. The "char" variable can only store one character at a time, that is why you can only print "A" instead of "A+".  
You have to use a character array (also known as strings) to store multiple characters. I have used a character array (constant character pointer) to fix this issue.*

*const char\**

*char \*grade="XX";*

*const char\* convert*

#include <stdio.h>

const char\* convert\_to\_alpha(double num)

{

double result;

char \*grade="XX";

printf("value of num : %f \n", num);

if (num >= 99.6)

grade = "A+";

else if (num >= 94.08 && num < 99.6)

grade = "A";

else if (num >= 88.2 && num < 94.08)

grade = "A-";

else if (num >= 82.32 && num < 88.2)

grade = "B+";

else if (num >= 76.44 && num < 82.32)

grade = "B";

else if (num >= 70.56 && num < 76.44)

grade = "B-";

else if (num >= 64.68 && num < 70.56)

grade = "C+";

else if (num >= 58.8 && num < 64.68)

grade = "C";

else if (num >= 52.92 && num < 58.8)

grade = "C-";

// and so on

return grade;

}

int main()

{

double num;

num = 65.0;

printf("value of num in main function: %f \n", num);

const char\* convert = convert\_to\_alpha(num);

printf("value of convert : %s \n", convert);

return 0;

}

Taking the advice of experienced programmer on board I realised a problematical issue that I would expect to encounter in creating further work for the C program is that course names and student names would be different lengths. This is not a problem if coding in a language such as Python which handles the memory allocation for you.

To be able to create the C program to be able to read different sizes of the data supplied by the user or imported from the spreadsheet, I would have not only needed to be able to declare different variable types and been very familiar with how to use pointers with structs, but I also would have had to learn how to use syntax for dynamic memory allocation and would have needed to implement a malloc function which I came across in the lecturer’s sample code.

*“malloc” or “memory allocation” method in C is used to dynamically allocate a single large block of memory with the specified size. It returns a pointer of type void which can be cast into a pointer of any form. It initializes each block with default garbage value.[[9]](#footnote-9)*

There are commonalities between C and Python which I found in the lecture videos, where with more practice I could have substituted programming skills learned for Python to work with programming for C.

*The #include in C is like an import in Python*

*The printf function in C is like Python print*

*C compiler relies on curly braces rather than indentation*

*Struct in C is like an Object in Object Oriented Programming (e.g. Class in Python)*

## Similarities in forming a Class in Python and a Struct in C

In OOP Python I began class for Course like this:

**class** **Course**:

**def** \_\_init\_\_(self,NameOfCourse,Subjects):

**global** count\_of\_subjects

self.NameOfCourse = NameOfCourse

self.Subjects = Subjects

count\_of\_subjects = len(Subjects)

To create a C Struct I for Course would have needed to do something like this

**Struct Course**

**{**

**char NameOfCourse;**

**char Subjects;**

**};**

**// however with char in C you have to indicate the memory size you are allocating to the characters**

1. https://www.nerd.vision/post/polymorphism-encapsulation-data-abstraction-and-inheritance-in-object-oriented-programming [↑](#footnote-ref-1)
2. <https://www.geeksforgeeks.org/polymorphism-in-python/> [↑](#footnote-ref-2)
3. https://www.nerd.vision/post/polymorphism-encapsulation-data-abstraction-and-inheritance-in-object-oriented-programming [↑](#footnote-ref-3)
4. <https://stackoverflow.com/questions/23277/what-is-the-difference-between-procedural-programming-and-functional-programming> [↑](#footnote-ref-4)
5. <https://newrelic.com/blog/nerd-life/python-programming-styles> [↑](#footnote-ref-5)
6. https://www.educba.com/c-vs-python/ [↑](#footnote-ref-6)
7. https://www.educba.com/c-vs-python/ [↑](#footnote-ref-7)
8. https://www.educba.com/c-vs-python/ [↑](#footnote-ref-8)
9. <https://www.geeksforgeeks.org/dynamic-memory-allocation-in-c-using-malloc-calloc-free-and-realloc/> [↑](#footnote-ref-9)