Logo

Description automatically generated

**Understanding core concepts of programming: algorithms, coding, debugging and paradigms.**

**Unit 4: Programming**



**2021 – 2022**

**Student: -------------------**

**Tutor: Marzieh Farahani**

**Jan 2022 Cohort – Group A1 B1**

|  |  |
| --- | --- |
| STUDENT NAME AND ID NUMBER |  |
| Qualification | Pearson BTEC Level 4 Higher National Certificate in Digital Technologies (General) |
| Academic Year | 2021-2022 |
| Unit Number & Name | Unit 4: Programming |
| Unit Leader | Marzieh Farahani |
| Unit Lecturer | Marzieh Farahani |
| Assignment Title | Understanding core concepts of programming: algorithms, coding, debugging and paradigms. |
| Type of Assignment | Report and e-Portfolio |
| Weighting | 100% |
| Issue Date | Week Commencing 14/02/2022 |
| Formative Submission Date | 25/03/2022  (25th March at 14:00 using Formative submission link on Moodle). |
| Summative Submission Date | 08/04/2022  (8th April at 14:00 using Summative submission link on Moodle). |
| Assessor | Marzieh Farahani |
| IV name | Sunita Kotta |

**Student Declaration**

|  |  |
| --- | --- |
| This is to confirm that this submission is my own work, produced without any external help except acceptable support from my lecturer. It has not been copied from any other person’s work (published or unpublished) and has not previously been submitted for assessment either at GBS or elsewhere. I confirm that I have read and understood the ‘[GBS Academic Good Practice and Academic Misconduct: Policy and Procedure](https://moodle.globalbanking.ac.uk/pluginfile.php/194290/mod_resource/content/1/GBS%20Academic%20Misconduct%20Policy%20V2.3.pdf)’ available on Moodle. | |
| **I confirm I have read and understood the above Student Declaration.** | |
| **Student Name (print)** | Mr. Olayinka Goodluck |
| **Signature** |  |
| **Date** | 22-11-2022 |

Table of Contents

[Introduction 4](#_Toc96964418)

[Task 1 - Design algorithms 5](#_Toc96964419)

[The process of turning an algorithm into working program code 5](#_Toc96964420)

[Benefits of using Python Programming Language 5](#_Toc96964421)

[Task 2- Programming Paradigms (Procedural, Object Orientated, and Event Driven) 6](#_Toc96964422)

[Example of Procedural 6](#_Toc96964423)

[Example of Event-Driven Programming 6](#_Toc96964424)

[Example of Object-Oriented Programming 6](#_Toc96964425)

[Task 3 - Implementation, Testing and Evaluation. 7](#_Toc96964426)

[Debugging Process 7](#_Toc96964427)

[Coding Standard 7](#_Toc96964428)

[Conclusion and Evaluation 8](#_Toc96964429)

[Reference list 9](#_Toc96964430)

[Appendix A: Code Listings 10](#_Toc96964431)

[Worksheet A – Procedural programs 10](#_Toc96964432)

[Worksheet B – Event driven Programs 10](#_Toc96964433)

[Worksheet C – Object-Oriented Program 11](#_Toc96964434)

[Worksheet D 11](#_Toc96964435)

# Introduction

Tips for writing an introduction:

* don’t write your introduction until you’ve completed your report. The introduction is a summary of what is contained in the report, and you cannot summarise what is in the report until you have finished it.
* The introduction should be only a few lines long. It is a brief paragraph designed to tell the reader what the report covers.
* The introduction should answer the following questions:
* Why has the report been written? most reports are commissioned to address a particular problem. Detail the problem and state why it’s significant to the business.
* Who commissioned the report? State who requested that the report be written in the first place – was it an individual, department, organisation or someone else.
* What is covered in the report? Detail the scope of the report and, if need be, say what is not covered too.
* How was the report carried out? Give details of what methods of assessment were used to investigate the problem.

With the vast growth of technology, people are concerned with how technology works, and how to get involved into technology, is it learnable, and can they learn it, or who exactly can learn it, is it made for special people.

One of the major worries about joining tech, is the fact that programming is involved, and thinking one needs to memorize a lot of syntax, do a lot of reading, and also glued to their system, as they mostly believe programmers live a boring lifestyle, unknown to them that programming is fun, and easy to learn, it all depends on the language they are speaking or coding with, understanding the algorithm, coding, debugging and its paradigms.

Here is why the Python Programming stands out, as it can be used to do a lot in the tech world, from web development/software development, data science, scripts and automation, Machine Learning, software testing, gaming and many more.

This is why Hillingdon hospital, radiology department has taken it upon themselves to create the awareness of becoming a python programmer, understanding the core concepts, as it will not just make you job ready, but you will be part of the solution tech driven society, writing codes that makes life easy for others to go about their daily activities.

It will help you to master the programming types, and deploy them in application development, either you are working as an individual, or in a team.

A proper knowledge of how to develop and publish an Application to what market place exactly, the cost of actualizing the app, and how long it will take to go live.

Duration of making payments, quarterly or annually, all this will be at your finger tips at the end of reading through this report.

Publishing an IOS App, and also an Android App, the review process and all, are also embedded in this report.

However, what was not covered in the report are names of such marketplaces you can patronize to publish your App, what IDE is best to use, Notebook editor to make use of for writing your codes for either Procedural programming, Object-Oriented Programming or Event Driven Programming Paradigms.

The report was carried out with solving problems with the three types of programming techniques,

Procedural programming, Object-Oriented Programming or Event Driven Programming Paradigms.

In comparison with the time taken to finish writing a solution to the problem, which started from the algorithm, to the codes, testing and debugging process.

Which of the techniques is best used for programming, and why the procedural programming technique, should not be considered if we want to create programming codes, that can be reusable, in cases where we have to create similar programs but for different customers/ clients.

And whatever one is doing, we need to be time conscious, as keeping to deadline of the project submission is very crucial. Which is one reason why making use of the python programming, where you have few lines of code and you have arrived at your solution using the OOP or EDV programming type.

If you are looking at becoming a programmer, I will recommend you go with python as it is easy to read, understand and debug.

And it only requires few lines of codes.

The syntax is easy to understand, and with the help of IDLE’s, with PEP8 installed you easily get guides on a better way of restructuring your codes, and refactoring your variables, making sure best naming convention is used for your variables.

As you make your decision to try, I wish you good luck with your programming endeavors filled with fun, as you are going to meet a lot of people in the python community, as the python programming is a very popular language.

I made new friends myself from our popular community while working on this project.

Cheers and I wish you all the best once again.

## **Task 1 - Design algorithms**

* Provide a definition of what an algorithm is and outline the process in building an application.

**Definition of Algorithm: -**

An Algorithm is a well-defined sequential computational step taken to solve a problem.

It takes in variables as inputs, and returns an output which solves a mathematical problem or real world problem.

**Process of building an App.**

1. Pre-Development Stage of the App

This has got to do with proper planning of the App development, in such a way that it addresses certain questions like,

a). who is your target market and audience?

b). what is your App going to do, and is there a similar App already in existence, what is the litigation of the already existing App, and how is your App going to address that?

c). what are user benefits of your App?

d). what problems is the App going to solve?

Once you have all the questions above answered by your App, then you can proceed with the scope of the project.

1. Scope of the App Project

At this stage, proper understanding of the App development is required, as it should be based on the following and answer the following questions like,

a). who are the development team?

b) is there a consultant to the development team?

c). what is the duration of the project?

d). what is the payment proposal for the App development project, per hour, per day, or upon completion of the project? After all the above questions have been answered, you are good to the wireframe of the App development project.

1. Wireframe of the App

The wireframe of the App determines the visual aspect of the App, what would be the homepage and what contents should be on it. Is it going to have a navigation bar, and what pages should be linked to each of the pages. Once that vision is clear, you are good to go the App Design stage.

1. App Design

The App design stage has got to do with beauty of the App, in regards to the color, font style for the headings, titles, font weights, logos inclusive. At this stage, ideas are open for suggestions before conclusion, as it addresses the user interface and how friendly it will be for the targeted users. and advancing to the next stage which is the App development proper.

1. App Development

Upon reaching a clear understanding and conclusion, the backend of the App can be developed, alongside with development team if any. Continuous communication is also required just to know if any adjustments have been done regards the App Design, Wireframe or Scope of the App Development project. Because any adjustments that comes up after this stage will attract additional charges which will affect the budget of the project.

1. App Beta Testing

This is the most crucial stage of the App development, as Beta or User testing is required to test the App functions, if the buttons and navigations are working, what’s the performance and response like, and how about the optimization? Beta testing should be amongst the contract of the App development Team as standard practice, and upon certainty of the App being bug free, before the App launch.

1. App Launch

At this stage the App is ready for launch, after being tested and working. And it could take couple of weeks for the App to be published pending on the market place you are looking at hosting your App on. As different review methods are used by IOS App and Android App.

IOS Apps are mainly reviewed before publishing, whereas Google play store might or might not review your App, and it will be published to go live right away on Google Play store. Meanwhile, Apple might review your App, and request for some changes to be made before it is being published. It is important to note that for your App to be published and sold, the developer will have to create an account with the particular market place they wish to publish their App. It could be an Individual account or Company account which will be charged annually.

1. App Post Launch

After the App launch, it is very vital to get user feedback as this could help with future App updates, ensure App is functioning according to its purpose of creation, and also, it’s user ability.

It is also very vital to make good use of the App internal Analytics to track downloads, App usage and retention, so as to monitor which of the pages is mostly visited and widely used, as it could enhance the App update for a better user experience.

* The relationship between algorithms and code.
* **RELATIONSHIP BETWEEN AND ALGORITHM AND A CODE**
* An Algorithm still remains the step-by-step formula to solving a mathematical problem or real-world problem.
* A Code uses an Algorithm as a guide in knowing what syntax to apply in solving Mathematical problems or real-world problems via a programming technique.

## The process of turning an algorithm into working program code

1. Defining the problem
2. Design algorithm
3. Write a program
4. Testing and debugging
5. Documenting the program

Sequence algorithm

* Variables to hold data, employee\_name, employee\_surname, em;ployee\_salary, rate,

first\_student\_name, second\_student\_name, third\_student\_name, exam\_grade\_of\_first\_student, course\_grade\_of\_first\_student, exam\_grade\_of\_second\_student, course\_grade\_of\_second\_student, exam\_grade\_of\_third\_student, course\_grade\_0f\_third\_student,

* Process to calculate net\_salary = confirmed\_employee\_salary – (confirmed\_employee\_salary \* confirmed\_employee\_rate |||| overallgrade\_of\_first\_student = exam\_grade\_first\_student + course\_grade\_first\_student, overallgrade\_of\_second\_student = exam\_grade\_second\_student + course\_grade\_second\_student, overallgrade\_of\_third\_student = exam\_grade\_third\_student + course\_grade\_third\_student, Average\_of\_three\_students = overallgrade\_of\_first\_student + overallgrade\_of\_second\_student + overallgrade\_of\_third\_student / 3
* Input and output employee\_name, employee\_surname, employee\_salary, rate, net\_salary,

exam\_grade\_of\_first\_student, course\_grade\_of\_first\_student, exam\_grade\_of\_second\_student, course\_grade\_of\_second-student, exam\_grade\_of\_third\_student, course\_grade\_0f\_third\_student, average\_of\_three\_students

Selection algorithm

* Variables to hold data, your\_number, x and y
* Process to calculate, 1. If a number inputted by a user is even: your\_number % 2 == 0,

2. Comparison of two numbers: if x != y and x > y, if x != y and x < y, else x == y.

* Input and output, exam\_grade\_of\_first\_student, course\_grade\_of\_first\_student, exam\_grade\_of\_second\_student, course\_grade\_of\_second-student, exam\_grade\_of\_third\_student, course\_grade\_0f\_third\_student, average\_of\_three\_students
* How to make decisions: validate all inputs and ensure the right arguments are given as values using the if, elif and else statements where necessary.

Iteration algorithm

* Variables to hold data, my\_number, I, number\_count, number\_sum, given\_number,
* Process to calculate, I = my\_number + 11, number\_count\_avg = number\_sum / number\_count
* Input and output, my\_number, I, number\_count, number\_sum, given\_number,
* How to make iteration, by using the For Loop especially when a range is known, or a while loop that will terminate when a certain condition is met.

## Benefits of using Python Programming Language

Python is a versatile language and has quite a lot of benefits.

1. Easy to read and understand.
2. Does not end with semi colon like other programming languages.
3. It is Object-Oriented.
4. It is a High-Level Language.
5. Does not need an interpreter.
6. It is popular and has a large community.
7. It has an extensive library, packages, frameworks and modules
8. It has a Graphical User Interface (GUI) support.
9. It uses few lines of codes to solve most problems in comparison to other programs.
10. It has a high employment opportunity.
11. It is a dynamically typed language.
12. It is an open-source language.
13. It is an independent platform.
14. It can be used for data science.
15. It can be used for machine learning.
16. It can be used for web development.
17. It can be used for Testing Frameworks.
18. It can be used for scripting and automation.

# Task 2- Programming Paradigms (Procedural, Object Orientated, and Event Driven)

Give explanations of what procedural, object orientated, and event driven paradigms are; their characteristics and the relationship between them.

Compare and contrast the procedural, object orientated, and event driven paradigms used in given source code of an application

## Example of Procedural

* Variables to hold data,
* Process to calculate
* Input and output
* How to make decisions
* Loops
* Functions

## Example of Event-Driven Programming

## What Events such as user actions (mouse clicks, key presses) are.

Event-driven programming is the paradigm used in applications with graphical user interfaces GUI

## Example of Object-Oriented Programming

* Class/Object
* Inheritance
* Abstraction
* Encapsulation
* Polymorphism
* They are all types of computer programming, which differs from each other by their functions and features.
* **Procedural Programming: -**
* This is derived from a structured and functional programming paradigm.
* It takes a step-by-step process to resolve computer programming problem.
* And the codes can be very long, that when working a project as a team, one can hardly remember the names of the variables being used.
* Outputs are being displayed on the terminal mostly, or printed on the screen, and it deals majorly with functions only.
* Procedural Programming divides many functions to solve a problem.
* **Object-Oriented Programming: -**
* As the name implies it is of Object type. It is made up of data’s, methods or procedures.
* These objects are being seen as Classes.
* One of its distinct features is the ability to modify objects, and data fields.
* It takes lesser codes to solve computer programming problem, and one can easily remember variables being used.
* Outputs are being displayed on the terminal mostly, or printed on the screen.
* It is the most preferred type of programming because of its abstract layers that has the power to add or deduct values. And once a class object is created, the abstract layer is activated.
* With the help of the encapsulation process, it helps developers to hide data.
* Object-Oriented Programming is the updated version of Procedural Programming.
* It has similar features with the Event Driven Programming, as they operate with Encapsulation, Polymorphism and Inheritance.
* Divides multiple programs into Objects.
* **Event Driven Programming: -**
* There are basically two types of Events Driven Program.
* The Keypress Event and the Mouse click events.
* It depends on the action executed. For instance, sensor outputs, keypress on your keyboards, and mouse clicks are displayed on a Graphical User Interface.
* EDV uses Events like notifications, messages, and outcomes, which are also similar in OOP and PP.
* EDV and OOP can be used together as they are both Orthogonal.
* The mainloop() is also present in the codes just to ensure all codes are run.

ADVANTAGES OF Object-Oriented Programming and Event Driven Programming Paradigms over the Procedural Programming.

* Faster development of software.
* Easier Debugging Process.
* It allows codes to be reusable, especially when writing similar programs.
* It isolates the user from the internal workings of the programs.

# Task 3 - Implementation, Testing and Evaluation.

* Write programs using IDE in worksheet A, B, C, D.
* Use IDE to manage the development process of the program.

## Debugging Process

* Explain the debugging process and debugging facilities available in the IDE.
* Evaluate how the debugging process can be used to help develop more secure, robust applications.
* Documentation of the debugging process in the IDE, with reference to watch lists, breakpoints, and tracing.
* How the debugging process can be used to help developers fix vulnerabilities, defects, and bugs in their code.

## Coding Standard

* What a coding standard is and its benefits when writing code.
* Outline the coding standard you have used in your code.

**DEBUGGING PROCESS**

The process of finding and correcting errors in a program is called Debugging.

It can be stressful sometimes especially when dealing with logical errors.

It is the beginning of the software testing part of programming.

The Debugging facilities available in IDE, taking Pycharm and VScode as an example, is

1. It gives the error a name.
2. States the line where the error occurred.
3. And even define why the error occurred with such a name.
4. You can click to highlight the errors, and as you traceback to the lines where the error occurred, and begin to debug.
5. Before you start debugging, the first step to take in debugging, is to copy all your codes, open up a new Python file, give it a name different from what you used to store your initial program file, paste it there and start debugging.
6. Upon fixing the bug, after testing, endeavor to copy the working codes, back to the program files after the deleting the old codes that gave rise to the error.

The debugging process is useful in developing more secured and robust Applications as it validates user inputs to the last, and prints a message to the terminal or on the screen informing the user of the invalid input, and also guides the user on the wright input the program is expecting from him/her.

**CODING STANDARDS:**

Is the habit of maintaining the rules and techniques to writing a clean code, which can be easily read, maintained, reused, and enhanced.

**Benefits of maintaining a good coding standard**

1. It helps your code to be readable, and well understood.
2. It facilitates easy debugging process.
3. It enhances faster App development.
4. It allows for better team integration.

**Coding Standards used in my code**

1. Standard way of naming variables, by assigning variables names that facilitates easy understanding of what the variable values should be.
2. I used relevant comments where necessary, so as to ensure a quick debugging process if a bug is encountered.
3. Labeled my files properly according to the worksheet and exercise, for easy understanding as well.
4. Indentations and line breaks were duly observed.
5. Used an underscore to join variable names together.
6. Simplified my codes to the best of my knowledge, and ensured the programs terminated properly leaving the user with a print statement of what the error encountered is and how to avoid such errors while achieving their results
7. Same technique for naming variables was used for naming functions as well.

**Conclusion and Evaluation**

* Evaluate the implementation of an algorithm in a suitable language and the relationship between the written algorithm and the code variant.

The algorithm is like the visions you get as you try to interprete the question.

This vision leads to step-by-step guide, of how the problem should be tackled and solved and this step by step guides follows the antiques of a good coding standard, from what your input will be, and what should happen to your input, in cases of a mathematical problem, what are trying to calculate with your input. In a real world problem, what should happen to your input, these are all processes under calculation, and a good coding standard will bring those vision from the algorithm to lime light. So, once you have successfully documented your algorithm, it’s time to code it out, after coding it out, the next is to test, if it was able to depict what the algorithm vision that was documented was. Debugging continues from there until the solution is achieved.

So, as good coding practice, one should always have their algorithms written before coding properly, as it will guide them on what step to take next upon completing previous steps, till the program ends.

* Critically evaluate the source code of an application which implements the procedural, object-orientated and event driven paradigms, in terms of the code structure and characteristics.

To be very candid, while writing a procedural programming code, you don’t want to leave your PC until you are done coding and have arrived the solution. Because, the process of debugging a procedural code can be so discouraging due to the number of lines of codes one has to write. The names of the variables used, are not remembered mostly and it takes the

help of a good comment to refresh your memory as to what the codes in a particular section is meant to do.

As for the Object\_Oriented Programming and Event Driven Prorgamming they have a systematic way of making long codes short, with their abstract nature encapsulation techniques, where data’s can be hidden and the desired result is still achieved, since they can easily group objects into classes.

With the help of functions, longer codes are broken down as well.

Codes written with OOP and EDV can be debugged easily, and it also facilitates faster software development without wasting much time.

* Evaluate the use of IDLE for development of applications contrasted with not using an IDLE.

What else can one use for developing applications without IDLE, a command prompt.

If you are working on algorithms that will definitely be debugged, working on command prompts simply means having to re-enter each codes line by line. But working with an IDLE, one can simply go straight to the line number he or she wants to edit, and test if the bug has been fixed in less than no time, hence, it will be not just faster to develop applications with IDLE, but also stress free. And anything that is stress free, becomes enjoyable.

* Critically evaluate why a coding standard is necessary in a team as well as for the individual.
* It helps your code to be readable, and well understood.
* It facilitates easy debugging process.
* It enhances faster App development.
* It allows for better team integration, such that a new member added to the team, would not be able to tell if the code was written by just an individual or a group of people.
* Changes made to the codes can easily be tracked as well with the help of comments.
* It can enhance easy migration for updating codes, and adding new designs to the program.

# Reference list

Books

Online resources

# 

# Appendix A: Code Listings

## Worksheet A – Procedural programs

Exercise 1:

* Algorithm (sequence)
* Code
* Testing

Exercise 2:

* Algorithm(sequence)
* Code
* Testing

Exercise 3:

* Algorithm(selection if statement)
* Code
* Testing

Exercise 4:

* Algorithm (selection if statement)
* Code
* Testing

Exercise 5:

* Algorithm (Iteration for loop)
* Code
* Testing

Exercise 6:

* Algorithm (Iteration for loop)
* Code
* Testing

Exercise 7:

* Algorithm (Mix of algorithm and function)
* Code
* Testing

Exercise 8:

* Algorithm (Mix of algorithm and function)
* Code
* Testing

## Worksheet B – Event driven Programs

Exercise 9: Keypress

* Algorithm
* Code
* Testing

Exercise 10: Mouse

* Algorithm
* Code
* Testing

## Worksheet C – Object-Oriented Program

Exercise Grade calculator or Employee class:

* Algorithm
* Code
* Testing

## Worksheet D

Whole application for Distinction