|  |  |
| --- | --- |
| Lecturer: | John O’Raw |
| Report Title: | Python Walkthroughs 1-6 |
| File Name: | L00177560.pdf |
| Submit to: | Blackboard |
| Date Submitted: | 10/09/2022 |

|  |  |
| --- | --- |
| Student Name: | Nathan Hand |
| Student Number: | L00177560 |
| Programme of Study: | PGDip in Cloud Computing |
| Module: | Infrastructure as Code |

Please refer to the Institute’s Quality Assurance Handbook, Version 3.0, September 2018

1. Practical work, forming part of the CA of a module, will only be assessed if the student has attended the relevant practical classes.
2. CA work must be completed within the schedules and specifications (specified in the CA brief). Students who submit CA late may forfeit some or all the marks for that work.
   1. The total marks available for an assessment be reduced by 15% for work up to one week late.
   2. The total marks available be reduced by 30% for work up to two weeks late.
   3. Assessment work received more than two weeks late will receive a mark of zero.

Work is deemed late when an unauthorised missing of a deadline has occurred.

1. CA must be the student’s own work, refer to Plagiarism Policy, in section 5.7 of the QA manual.

Contents

[Description 3](#_Toc116235231)

[Aims 3](#_Toc116235232)

[Methodology 4](#_Toc116235233)

[Results and Testing 5](#_Toc116235234)

[Conclusions 6](#_Toc116235235)

[References 8](#_Toc116235236)

# Description

This report covers the work carried out through Python Walkthroughs 1-6. The main topics covered over the past two weeks in these sections were, Assignments and Variables, Documentation, Data Structures, Loops and statements, Functions and lastly Modules and Packages.

Reading through the documentation for each walkthrough following highlighted key words and topics to further research to help the writing of this report and knowledge gained by the student.

# Aims

To complete the assignments listed in each topic through walkthroughs 1-6 the following aims were identified

1. To identify the scope and task of each exercise.
2. To determine the key words and researched needed to be carried out for each walkthrough.
3. Based on the research and exercise of each walkthrough to deliver a functioning coding example for submission.

# Methodology

In this section the procedures and methods used to carry out the work which we identified are documented. The walkthroughs were opened in sequence 1-6 and read through. The aims were then followed to result in a complete and working exercise.

1. The required work was identified from the lectures notes [1].
2. The programs needed to carry out the work were prepared. [2]
3. The information in the walkthroughs was studied.[3]
4. Additional external research on key words and methods was conducted. [4]
5. The exercise are coded and tested.[5]

# Results and Testing

Combining the lecture notes with the reading of external research has allowed 6 separate folders with exercise and examples of working code to be created.

# Conclusions

In this block of work it was intended that 6 main sub folders with working examples of code would be delivered with the specification of work to be carried out taken from the lecturer’s notes [1]. It was necessary to conduct a vast amount of independent research as the material was worked through and key words and methods were identified[4]. I will break down the conclusion based on each walkthrough with key words for each walkthrough identified and independently researched.

Notepad++, VSC and folder structure was setup to layout the ground work for the walkthroughs[2].

Walkthrough 1 covered assignments and variables the main deliverable from this walkthrough was a working example of **Arithmetic Operators** – Is an operator that performs arithmetic operations on numbers and groups performed by operators + ,-,\*,/,%,// and \*\*[2]. The resulting program delivered from this walkthrough demonstrated all the use of the above operators[5].

In this first section some other key words that came up were,

**ASCII** – American standard code for information interchanged is a set of standards that were developed to communicate with a computer using 8-bit code these 8 bits make up a byte so you could communicate such as 0111 0100.

**Unicode** – Was created to address the issue of different code sets and the confusion associated with it.

Walkthrough 2 covers the documentation method one should follow while producing blocks of code although no direct exercise is requested in this section some external research on git hub or other sources of code are beneficial to look over.[1][3]

Walkthrough 3 covered data structure this section had 3 main exercises to cover and produce, they laid out to work with Tuples using an index and verify it is immutable[1]. Review the data structure document listed and work through some examples and to use different methods of dictionary’s[1].

A key method used which required further research was **Tuples** – Is one of the three basic sequence types, they are immutable sequences and are usually used to store different collections of heterogeneous data.[4]

3 separate small code blocks were produced that demonstrated the work listed out in the notes[5]

Walkthrough 4 covers Loops and statements and very much like the previous walkthrough has 3 main exercises laid out to produce working blocks of code[1].

The main information given in this section is to cover coding loops and conditional statements in python[3].

The main end exercise of this walkthrough was to deliver a program that could convert kelvin temperature units into Celsius and Fahrenheit[1].

**Boolean conditions –** A boolean is a logical statement that will either be true or false, they can compare a large variety of data such as greater to or equals to and so on this is a very good condition to use for certain programs but as with all comes with its own limitations[4].

Walkthrough 5 moves onto Functions within coding the main aim is to demonstrate correctly how to unitize functions in python.[1]

The exercise in this section is both writing some code but also breaking down and explaining what is going on in an already existing example of code block[2].

The main block of code is to search a list of numbers and return a true of false statement given even numbers in the list.[5]

While working through this section **lambda function** popped out as something that needed further research.

**lambda function –** is a small anonymous function that can take any number of arguments but may only have one expression.[4]

After conducting additional research the requested exercise was completed producing a correct output[5].

The final walkthrough for this report 6 covered modules and packages, as with some of the other walkthroughs this part included one more exercise that broke down into smaller segments of coding[1].

The main deliverable was to tidy up the examples of code given and understand why and how the other blocks of code were working[2].

As a result of walking through all these walkthroughs it has shown that the student has the knowledge to do create some basic scripts and a broad understanding of some of the key concepts when it comes to programming.

# References

W3schools.com. 2022. *Python Lambda*. [online] Available at: <https://www.w3schools.com/python/python\_lambda.asp> [Accessed 7 October 2022].

Intel.com. 2022. *arithmetic operator Definition*. [online] Available at: <https://www.intel.com/content/www/us/en/programmable/quartushelp/17.0/reference/glossary/def\_arith.htm#:~:text=An%20operator%20that%20performs%20arithmetic,and%20minus%20(%20%2D%20)%20symbols.> [Accessed 7 October 2022].

Intel.com. 2022. [online] Available at: <https://www.intel.com/content/dam/www/program/education/us/en/documents/the-journery-inside/digital/tji-digital-info-handout4.pdf> [Accessed 7 October 2022].

Ibm.com. 2022. *IBM Documentation*. [online] Available at: <https://www.ibm.com/docs/en/zos/2.4.0?topic=definition-arithmetic-operators> [Accessed 7 October 2022].

Ibm.com. 2022. *IBM Documentation*. [online] Available at: <https://www.ibm.com/docs/en/workload-automation/9.5.0?topic=support-what-is-unicode> [Accessed 7 October 2022].

Docs.python.org. 2022. *Built-in Types — Python 3.10.7 documentation*. [online] Available at: <https://docs.python.org/3/library/stdtypes.html?highlight=tuples#tuple> [Accessed 7 October 2022].

Docs.oracle.com. 2022. *3.6 Boolean Expressions*. [online] Available at: <https://docs.oracle.com/cd/B12037\_01/olap.101/b10339/expression006.htm> [Accessed 7 October 2022].