Laboratory Report

Ollscoil Teicneolaíochta an Atlantaigh

Atlantic Technological University

Lecturer:	John O'Raw
Report Title:	Report – Walkthroughs 7-12
File Name:	L00177769.pdf
Submit to:	Blackboard in PDF format only
Date Submitted:	28/10/2022

Student Name:	Martina Atkinson
Student Number:	L00177769
Programme of Study:	MSc PGDip in Cloud Computing
Module:	HCT CP90x

Contents

Description

This report documents the practical work carried out in the final 3 weeks of the Python programming course. The lecturer provided six technical walkthroughs, covering the basic theory of the Python language with sample code and exercises. Topics covered were error handling, object-oriented programming, testing, logging, the transmission of files using network protocols and project folder creation. Students were required to type sample code, complete the exercises and store their work in a folder each week for later consolidation into a compressed archive for submission and grading.

Aims

The intention of the work was to build on the Python skills acquired over the previous weeks. Code reuse was emphasised and generally practised. The following individual aims were identified: -

1. To write maintainable and robust code using proper error-handling techniques.
2. To gain skills in object-oriented programming.
3. To develop testing skills to aid in the delivery of bug-free quality software.
4. To gain experience in system logging techniques.
5. To enhance network utility skills using standard network protocols in the exchange of files.
6. To understand best practices in creating directory structures for projects.

Methodology

All work was completed using Windows 11 Desktop, Python 3.10 and Visual Studio Code editor. The lecturer provided six technical walkthroughs containing Python theory, sample scripts and exercises to facilitate learning. Students were required to type the sample code to become familiar with the language's syntax and complete the required exercises to improve their problem-solving skills. Code reuse was emphasised and used to teach new coding principles. References were provided, and students were expected to do additional background reading. All completed code was stored in a folder structure for each walkthrough and for later submission and grading.

Results and Testing

Based on the briefing for this work, the code written and tested was sufficient proof of work completed. The attached zip-compressed file contains six separate folders directly addressing each walkthrough. Below is the list of the issues encountered in the various walkthroughs [Table 1].

Table 1 Walkthrough Results

Walkthrough No	Status	Comments/Issues
7. Error Handing	Completed	The exercise on zero division was completed successfully.
8. Object Oriented Coding	Completed	Class template and device exercises were successfully completed. OOP required considerable extra reading for understanding.
9. Writing Tests	Completed	The Pylint code analyser proved annoying as it misinterpreted variables as constants when defining outside functions and was excessively noisy and pedantic on trailing spaces.
10. Logging & Time	Completed	All tasks were completed but references were missing from the documentation.
11. Network Utilities	Completed	No issues were encountered, and all exercises were completed.
12. Creating a Project	Completed	The project folder structure was completed using the batch file.

Conclusions

During this three-week period, students progressed in learning the basics of the Python language via a series of technical walkthroughs provided by the lecturer. The course content became more focused on the infrastructure area with the writing of network utilities taught with real-world examples. This final part of the course also focused on improving programming skills and instilling best practices in coding through error-handling techniques, testing and proper documentation.

The subject of writing utilities and error handling was covered in walkthrough 7 [1]. Python comes well equipped with built-in modules for this purpose, and the Platform module was used to determine the operating system and the OS module to find the current working directory [2][3]. It has been said that the mark of a good computer program is one that runs successfully and is not easily broken. Programs may crash or fail for several reasons, such as invalid inputs (entered unintentionally or maliciously) or things outside the developer's control, such as disk space or broken network connection [4]. Some errors, such as syntax where the language's grammar has been ignored, cannot be handled, but other run-time errors, known as exceptions, should be anticipated and handled in scripts. While languages such as C++ and JavaScript use "try" and "catch" keywords to handle these errors, there is no such keyword as "catch" in the Python language, so it uses a "try" and "except" for these run-time errors or exceptions and optional else and final blocks [5]. In walkthrough 7, several practical examples were completed to demonstrate error handling [1]. The student was also challenged to write a script to handle zero division and input errors and was completed successfully.

In computing, the term programming paradigm refers to a way or style of programming and is not specific to one language [6]. Python is a multi-paradigm programming language as it can incorporate imperative, procedural, functional, and Object-Oriented Programming (OOP) styles [7). In walkthrough 8, students were introduced to the object-oriented paradigm, a programming method around objects rather than functions or logic [8][9]. The walkthrough helped navigate the minefield of OOP terminology, with explanations given of the key characteristics of OOP which include class, abstraction, encapsulation, inheritance and polymorphism. Students were guided step by step in creating a class with its associated constructor and methods and shown how to instantiate objects from this class for use as a template for other projects. An abstract class, which is not instantiated, was created to hold attributes and methods.

The current buzzword in information technology is DevOps which is an amalgam of the word development and operations. Traditionally development and operations staff have existed in their silos, with responsibility for problems in hardware and software being passed from one side to the other [10]. DevOps represents a culture shift and is centred around the principles of teamwork, reducing silo thinking by developing relationships between development and operation staff, fast iterations and automation where possible [11]. These all-present major challenges for infrastructure staff who are drawn more into the scripting world with the demand for high-performance deployment and more robust monitoring. Walkthrough 9 introduced Pylint, a static code analysis tool also known as a linter. It can check code for errors and coding standards and offers suggestions on refactoring, helping to develop a good coding style but was found to be excessively noisy at times [12][13]. The Unittest module was used for unit testing, where parts of the software are tested to validate performance against design [14]. A file was created with a filename starting with the keyword test, a class was created to extend ‘unittest.TestCase’, and methods, named with the starting word test, were written for test cases.

Dates and times challenge any programmer as they must deal with time zones, daylight savings and different date formats. [15]. In walkthrough 10, the DateTime module calculated the current and Unix

epoch time [16]. Students were also asked to code the current time and date, which was completed successfully. Logging was also part of this walkthrough, and it is a valuable tool in the development and debugging of software [17]. Python has a built-in logging module, but logging was done manually. Information on the operating system and CPU was written to a log and CSV (Comma Separated Value) files using various functions and modules, including PSutil, a cross-platform library used in system monitoring and retrieves information on running processes and system utilization [18]

Walkthrough 11 demonstrated working with network protocols, specifically in the transfer of files [19]. First, without authentication, the File Transfer Protocol (FTP) was used to transfer data between server and client using Windows Secure Copy (WinSCP) [20]. A Python dictionary item was created to store the FTP details in a separate file in the settings folder. In the second demonstration, User Datagram Protocol (UDP) was used to send packets and with this unreliable protocol, there was no error correction, acknowledgement or guaranteed delivery [21]. The socket module was imported to create a bidirectional communication channel for sending the files between the server and the client and tested using the console window [22]. In the third demonstration, the more reliable Transmit Control Protocol (TCP) was used with connections established prior to sending, packets acknowledged, and dropped packets resubmitted [23]. Finally, multicast was used to broadcast packets to every client on the network using a specific IP address for multicast transmission.

The topic of creating a project folder structure was covered in walkthrough 12 [24]. Starting a project with a solid folder structure benefits the author in staying organised and others using the code to discern the data flow and code execution [25]. It may also prove a timesaver in the future when code must be revisited. A folder structure for a project was successfully created using a batch file. A script was stored in the source folder, and the main file at the root of the project folder to allow for relative paths making the project more portable. A readme file to explain the project and a changelog showing revision history should also be placed at the root of the folder [26]. This walkthrough was beneficial in highlighting the best practices for project folder creation.

The last three weeks of the basic Python course were completed successfully, as all walkthroughs were comprehensible, and all aims were met. Students were left with an arsenal of code for a future project. Infrastructure as Code is a fearful subject for many seasoned infrastructure staff as some say that it is taking the pitfalls of networking and adding them to the pitfalls of programming [27]. The course has somewhat alleviated these fears and opened the possibilities that Infrastructure as Code may bring to the working lives of infrastructure staff.

References

[1] J. O'Raw, "Walkthrough – 7. Handling Errors" https://lyitbb.blackboard.com/ultra/courses/_58891_1/cl/outline (accessed Sep. 24, 2022).

[2] "Platform Module in Python," *GeeksforGeeks*, Jan. 16, 2020. https://www.geeksforgeeks.org/platform-module-in-python/ (accessed Oct. 22, 2022).

[3] "What is Python's OS Module." https://www.tutorialspoint.com/what-is-python-s-os-module (accessed Oct. 25, 2022).

[4] M. Bishop and D. Frincke, "Teaching robust programming," *IEEE Security & Privacy*, vol. 2, no. 2, pp. 54–57, Mar. 2004, doi: 10.1109/MSECP.2004.1281247.

[5] Carson, "What is Error Handling and Why is it Essential to Programs?," *Central Galaxy*, Apr. 09, 2022. https://www.centralgalaxy.com/error-handling/ (accessed Oct. 22, 2022).

[6] "What exactly is a programming paradigm?" *freeCodeCamp.org*, Nov. 12, 2019. https://www.freecodecamp.org/news/what-exactly-is-a-programming-paradigm/ (accessed Oct. 23, 2022).

[7] N. Thaker and A. Shukla, "Python as Multi-Paradigm Programming Language," *International Journal of Computer Applications*, vol. 177, pp. 38–42, Jan. 2020, doi: 10.5120/ijca2020919775.

[8] J. O'Raw, "Walkthroughs - 8. OO Coding" https://lyitbb.blackboard.com/ultra/courses/_58891_1/cl/outline (accessed Sep. 24, 2022).

[9] A. Hossain, "Object Oriented Programming in Python — What and Why?," *Medium*, Mar. 13, 2022. https://towardsdatascience.com/object-oriented-programming-in-python-what-and-why-d966e9e0fd03 (accessed Oct. 23, 2022).

[10] "DevOps For Dummies Cheat Sheet," *dummies*. https://www.dummies.com/article/technology/programming-web-design/general-programming-web-design/devops-for-dummies-cheat-sheet-263000/ (accessed Oct. 26, 2022).

[11] M. Artac, T. Borovssak, E. Di Nitto, M. Guerriero, and D. A. Tamburri, "DevOps: Introducing Infrastructure-as-Code," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, May 2017, pp. 497–498. doi: 10.1109/ICSE-C.2017.162.

[12] J. O'Raw, "Walkthroughs - 9. Writing Tests" https://lyitbb.blackboard.com/ultra/courses/_58891_1/cl/outline (accessed Oct. 7, 2022).

[13] "What is Pylint? Python Programming » Network Interview," Apr. 20, 2021. https://networkinterview.com/what-is-pylint-python-programming/ (accessed Oct. 23, 2022).

[14] D. Vikma, "Python Unittest Vs Pytest: Choose the Best," *Python Pool*, Jun. 27, 2021. https://www.pythonpool.com/python-unittest-vs-pytest/ (accessed Oct. 26, 2022).

[15] R. Python, "Using Python datetime to Work With Dates and Times – Real Python." https://realpython.com/python-datetime/ (accessed Oct. 23, 2022).

[16] J. O'Raw, "Walkthroughs - 11. Logging and Time" https://lyitbb.blackboard.com/ultra/courses/_58891_1/cl/outline (accessed Oct. 7, 2022).

[17] “An introduction to logging for programmers,” *freeCodeCamp.org*, Jul. 27, 2017. https://www.freecodecamp.org/news/you-should-have-better-logging-now-fbab2f667fac/ (accessed Oct. 28, 2022).

[18] G. Rodola, “psutil: Cross-platform lib for process and system monitoring in Python.” Accessed: Oct. 29, 2022.

[19] J. O’Raw, “Walkthroughs - 11. Network Utilities” https://lyitbb.blackboard.com/ultra/courses/_58891_1/cl/outline (accessed Oct. 7, 2022).

[20] “How to use FTP in Python,” *PythonForBeginners.com*, Jun. 13, 2013. https://www.pythonforbeginners.com/code-snippets-source-code/how-to-use-ftp-in-python (accessed Oct. 25, 2022).

[21] “Why UDP is an unreliable protocol? – Technical-QA.com.” https://technical-qa.com/why-udp-is-unreliable-protocol/ (accessed Oct. 27, 2022).

[22] “Sockets | Python,” *GeeksforGeeks*, Jul. 09, 2019. https://www.geeksforgeeks.org/sockets-python/ (accessed Oct. 27, 2022).

[23] “Types of Network Protocols and Their Uses.” https://www.w3schools.in/types-of-network-protocols-and-their-uses/ (accessed Oct. 27, 2022).

[24] J. O’Raw, “Walkthroughs - 12. Creating a project with Python” https://lyitbb.blackboard.com/ultra/courses/_58891_1/cl/outline (accessed Sep. 24, 2022).

[25] T. de Kok, “How to keep your research projects organised, part 1: folder structure,” *Medium*, Apr. 11, 2021. https://towardsdatascience.com/how-to-keep-your-research-projects-organized-part-1-folder-structure-10bd56034d3a (accessed Oct. 27, 2022).

[26] “How to Properly Organize Files in Your Codebase & Avoid Mayhem - SitePoint,” Feb. 20, 2020. https://www.sitepoint.com/organize-project-files/ (accessed Oct. 27, 2022).

[27] “Infrastructure as Code (IaC) Pitfalls and How to Avoid Them | env0 blog.” https://www.env0.com/blog/infrastructure-as-code-pitfalls-and-how-to-avoid-them (accessed Oct. 28, 2022).

Appendices