Laboratory Report

Ollscoil Teicneolaíochta an Atlantaigh

Atlantic Technological University

Lecturer:	John O'raw
Report Title:	Walkthroughs 7-12
File Name:	Report2.pdf
Submit to:	Blackboard in PDF format only
Date Submitted:	30/10/2022

Student Name:	Nathan Hand
Student Number:	L00177560
Programme of Study:	MSc PGDip in Cloud Computing
Module:	HCT CP90x

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.
   a. The total marks available for an assessment be reduced by 15% for work up to one week late.
   b. The total marks available be reduced by 30% for work up to two weeks late.
   c. 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.

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

Contents

Description

This Report documents the work carried out through walkthroughs 7-12, Handling Errors, OO Coding, writing tests, Logging and time, Network utilities and creating a project with python. A key set of programs was set to create and record the work created throughout. On the student's computer vscode, notepad ++, CMD was used to record and create the work. In the conclusion the main highlights covered were discussed.

Aims

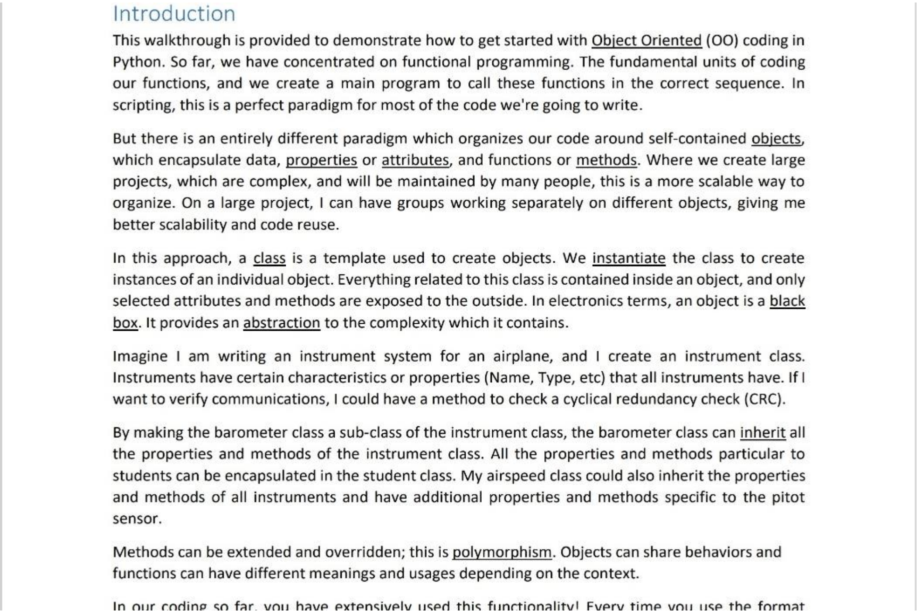
To further develop the student's ability's using python and the associated tools used during coding development, the following aims were identified.

1. Identify the key words and topics that need further learning/investigation.
2. Research key topics and words on external papers and websites
3. To create working code segments based on provided notes.
4. To create code segments requested by the exercises throughout the walkthroughs.
5. Demonstrate through work a clear understanding of learning carried out.

Methodology

In this section, the procedures used to carry out the work specified in the aims are documented. The student logged into blackboard and read through each walkthrough highlighting any topics they did not understand for further research, they then moved onto working on the code segments listed in the notes before proceeding onto the exercises listed in each walkthrough .

1. The required specifications were identified from the lecturer's notes [1]
2. To determine the key characteristics, the walkthroughs were read through, and key phrases/takeaways highlighted such as in Figure 1.



Introduction

This walkthrough is provided to demonstrate how to get started with Object Oriented (OO) coding in Python. So far, we have concentrated on functional programming. The fundamental units of coding our functions, and we create a main program to call these functions in the correct sequence. In scripting, this is a perfect paradigm for most of the code we're going to write.

But there is an entirely different paradigm which organizes our code around self-contained objects, which encapsulate data, properties or attributes, and functions or methods. Where we create large projects, which are complex, and will be maintained by many people, this is a more scalable way to organize. On a large project, I can have groups working separately on different objects, giving me better scalability and code reuse.

In this approach, a class is a template used to create objects. We instantiate the class to create instances of an individual object. Everything related to this class is contained inside an object, and only selected attributes and methods are exposed to the outside. In electronics terms, an object is a black box. It provides an abstraction to the complexity which it contains.

Imagine I am writing an instrument system for an airplane, and I create an instrument class. Instruments have certain characteristics or properties (Name, Type, etc) that all instruments have. If I want to verify communications, I could have a method to check a cyclical redundancy check (CRC).

By making the barometer class a sub-class of the instrument class, the barometer class can inherit all the properties and methods of the instrument class. All the properties and methods particular to students can be encapsulated in the student class. My airspeed class could also inherit the properties and methods of all instruments and have additional properties and methods specific to the pitot sensor.

Methods can be extended and overridden; this is polymorphism. Objects can share behaviors and functions can have different meanings and usages depending on the context.

In our coding so far, you have extensively used this functionality! Every time you use the format

(Figure1)

3. The tool notepad++ and vscode were ran from the student's computer.
4. The lecture guided code segments were created.
5. The student led self-exercise segment was then created
6. All completed code segments were broken down and saved in a file structure as listed in the lecture walkthroughs.

Conclusions

In this work it was identified the key learning topics and phrases needed to be gone over for this module and programme of study from the lecturer's notes [1]. It was then necessary to work through assisted lecture guided coding segments to give a broad understanding of what was trying to be thought and achieved through each walkthrough. Based on the research conducted and the read through of all these walkthroughs it is possible for the student to deliver either semi of fully working coding exercises along with the explanation of how each of them work.

Several key words and topics that the student was unsure of and needed to further investigate were identified before and now require external and independent learning on part of the student [2].

Run Time Error – Is a software or hardware issue that prevents a program from executing/functioning correctly. (*Runtime errors - Browsers*, n.d.)

Camel case – This is when two words are separated by underscores and all letters are lowercase. (*camelCase - CS50's Introduction to Programming with Python*, n.d.)

Test driven development – This is a method of building code and the tests for the code one use case at a time rather then in a larger block of code, it is often used to improve the quality of the code.

TimeDelta – This is a duration that expresses the difference between time, date or datetime in microseconds.

Multicast – One source reaching out to multiple destinations interested in receiving traffic. (*Multicast Overview | Junos OS | Juniper Networks*, n.d.)

After conduction this background research the student was then able to carry out the guided work described in the lecture's notes [3].

At the end of this segment of word the student should have 5 separate folders with corresponding file structure, each of these will have several completed blocks of code that lay the groundwork for the student to carry on the intendent exercise[3].

The student now works on the exercises as requested in the notes provided [1]. At the end of each walkthrough the student should have a semi working or working exercise program [4] with sufficient description and comments on each to show a clear understanding of the learning that has been carried out during the walkthroughs.[5]

References

Runtime errors in Internet Explorer - Browsers. (n.d.). Microsoft Learn: Build skills that open doors in your career. https://learn.microsoft.com/en-us/troubleshoot/developer/browsers/development-website/ie-runtime-errors[Accessed 26 10 2022]

camelCase - CS50's Introduction to Programming with Python. (n.d.). CS50: Computer Science Courses and Programs from Harvard | edX. https://cs50.harvard.edu/python/2022/psets/2/camel/[Accessed 26 10 2022]

(n.d.).IBM,Deutschland.| https://www.ibm.com/garage/method/practices/code/practice_test_driven_development/[Accessed 26 10 2022]

Python.org, v3 Datetime," [Online]. Available: https://docs.python.org/3/library/datetime.html. [Accessed 26 10 2022]

Multicast Overview | Junos OS | Juniper Networks. (n.d.). Juniper Networks – Leader in AI Networking, Cloud,&Connected,Security,Solutions. https://www.juniper.net/documentation/us/en/software/junos/multicast/topics/concept/multicast-ip-overview.html[Accessed 26 10 2022]