CMM028 IT Infrastructure and Administration

Coursework 2 Specification – Practical Assessment

*This Coursework is an **individual** piece of work. All Coursework will be screened for evidence of Academic Misconduct (e.g. Collusion and Plagiarism).*

Hand-in Dates	16:00 on Wednesday 24 November 2021
Learning Outcomes Assessed	This coursework assesses: LO4 Implement effective interconnections between system programs. LO5 Design and implement scripts for network connectivity and information management.
Deliverables and Submission Method	A single zip file containing 7 Python scripts as described below. Submitted by you via the Practical Assessment dropbox on the module CampusMoodle page. These are discussed further on page 2. **Submission deadline is 4:00PM** – a 30 minute grace period has been added for technical issues that prevent submission by the deadline.
Coursework Grade and Feedback Arrangement.	Provisional grades and written feedback for this coursework will be available within 20 working days from the submission from the dropbox to which you submitted your work. Your mark for each element of the coursework will be determined using the number of marks given in the question. Element marks will be combined using the profile on page 4 to give the grade for this coursework.
Overall Module Grade	This coursework contributes 50% of the overall module grade. Your grade from this coursework will be combined with other component grades using the grid on page 4.
Questions? Contact	Mark Bartlett – m.bartlett3@rgu.ac.uk or via Teams For Staff Contact Details and Office Hours see http://campusmoodle.rgu.ac.uk/course/view.php?id=96625
Moderation	This coursework has been moderated by the School of Computing Moderation Panel, which comprised a detailed technical review and panel overview to schedule staggered submissions.
Notes on Penalties, Extensions and Deferrals	Coursework submitted late will be recorded as a Non-Submission (NS). If you are unable to meet the submission date/time please note the procedure as per the university's Fit to Sit Policy, found at: https://www.rgu.ac.uk/about/governance/academic-governance/student-and-applicant-forms Please ensure to follow the instructions at the webpage above. Coursework Extension requests must be submitted 24 hours before the submission date and time. Deferral Requests can be submitted in advance of the deadline and up to 5 working days after the deadline date/time. Either request must be sent to the following email address: StudentRequestSoC@rgu.ac.uk

Please read the entire Coursework specification carefully before starting. If any aspect is not clear, seek advice and assistance from the Module Coordinator. If you have used, or have been inspired by any other people's work, make sure that you state and reference the source clearly in your submission. Failure to do so may result in an accusation of plagiarism.

THIS IS AN INDIVIDUAL ASSESSMENT

You are not authorised to work with other students or non-students, not to share solutions for this assessment. This contravenes University Academic Regulations and those doing so will have been deemed to have colluded and will be subject to Academic Misconduct procedures.

All sources must be appropriately referenced at the point of their use in the submission.

The overall aim of this coursework is to develop a number of Python scripts to obtain information about the machine on which they are running, including its connection to the internet. Your Python scripts should work in Python 3.

You should submit your final scripts in a single zip file to the dropbox on the module Moodle page.

You may freely use and adapt code I have provided you with on the module. If you use code from anywhere else, you must clearly note this in your scripts as a comment. Failure to do so may be considered academic misconduct. You must not work on code with others in the class or exchange solutions, again this will be considered as academic misconduct.

The coursework asks you to develop 7 scripts in Python. In each case, you should seek to produce good quality code that adheres to Python standards and uses the principles of programming that you have learned on this module and others. You do not need to produce large quantities of comments in your code – restrict this to where you have something to note or explain. You do not need to provide any evidence of testing, but I would advice you do test it; otherwise how do you know whether it works? Your output from the scripts should be neat and tidy, and make sense to someone with a knowledge of the subject but who has not seen this assessment.

The maximum number of marks for the correct functioning of each task is noted next to the task. In addition, there are

- 10 marks for the overall quality of your code across the tasks.
- 5 marks for the quality of your output – based on whether your scripts produce reports and warnings that would make sense to a reasonably intelligent user.

Part A – A CPU Monitor [55 marks]

In this part, you will build a series of scripts that check the CPU usage of the machine. Each script is of increasing complexity. I recommend using the solution to each task as a starting point for the next one, e.g. when beginning task 2, make a copy of your solution to task 1 and use this as your starting point.

Task 1 – Basic CPU Monitor [5 marks]

Create a script that checks the CPU usage of your computer every 5 seconds. If it ever finds that this is over 90%, it should print a message to warn the user that the CPU usage is high. Your script should run forever.

Task 2 – Intelligent CPU Monitor [15 marks]

Create a script which monitors the CPU usage as above. However, this time it should also print out a warning if the CPU usage increases a lot above normal levels.

To do this, it will be necessary to know what the normal CPU usage level is. When the program starts, you should check the CPU usage every 5 seconds for 2 minutes, storing the usage levels in a list each time you check them.

Following this, you should start checking the usage every 5 seconds and print out a warning if either

1. The usage level is over 90%, or
2. The usage level is more than double its average value in the last 2 minutes.

(Don’t forget that you will need to keep your list up to date – it needs to contain data about the last 2 minutes. Every time you do a new check, you should remove the oldest check from the list and add the new check to it.)

Task 3 – CPU Problem Diagnoser [15 marks]

Create another script. This one should work exactly the same as the previous one, except that when the CPU usage is high, it should try to work out why and tell the user.

There are three explanations you can tell the user for the usage being high:

1. If any process is using over half of the current CPU usage, useful information about this process should be reported to the user.
2. If there are many more processes than normal running, this should be reported as the reason (of course this means you script will need to work out how many processes are usually running…).
3. If neither of these is true, you should say it is high for an unknown reason.

Task 4 – Process-based CPU Problem Diagnoser [20 marks]

Your final script should track the CPU usage of each individual process and warn the user if any process increases it’s usage to over 10 times what it was 5 minutes earlier. Hint: Rather than a list to track the CPU usage, you might want a dict where the keys are the ids of the running processes.

Part B – A Network Connection Checker [30 marks]

In this part of the assessment, you will build a simple system that can be used to check some network properties of the machine on which it is run.

Task 5 – Port State Checker [5 marks]

Produce a script which checks the status on the ports on your local machine. Specifically, it should look at the following ports:

- Port 22 Secure Shell (SSH)
- Port 23 Telnet
- Port 53 Domain Name System (DNS)
- Port 80 HTTP
- Port 443 HTTPS

Your script should display a report on the screen stating whether each of them is currently open or closed.

Task 6 – Connection Status Checker [10 marks]

Produce a script which checks the quality of the internet connection of the machine. To do this, you should try connecting to two different well-known and widely used websites. For each website, you should check whether the website responds to your request, and if so, how long it takes to do so.

Based on this information, you should output a report to the screen informing the user of whether you believe they have a good internet connection, a poor internet connection or no internet connection. Again, your output should make sense to someone with an understanding of the topic.

Task 7 – Network Checker System [15 marks]

Produce a script that combines the answers to the previous two tasks, and provides a menu system so the user can select which report to see.

The menu should tell the user what checks can be run and also have an exit option. It should then wait for the user to enter their choice. If the user selects one of your two checkers, you should run the required checks, display your report and then ask them to make another choice. This should continue until the user selects the exit option, at which point the program should stop.

Your grade for the practical assessment component will be determined as a mark out of 100 for the 6 tasks plus the quality of the code and the outputs.

GRADE	REQUIREMENTS
A	80 marks – 100 marks
B	70 marks – 79 marks
C	60 marks – 69 marks
D	50 marks – 59 marks
E	40 marks – 49 marks
F	Under 40 marks

This assessment is worth 50% of your overall module grade. Your mark from this assessment will be combined with your mark from the other assessment for this module to give your final module mark.

Exam Style Assessment

Practical Assessment

	A	B	C	D	E	F	NS
A	A	A	B	B	C	E	NS
B	A	B	B	C	C	E	NS
C	B	B	C	C	D	E	NS
D	B	C	C	D	D	E	NS
E	C	C	D	D	E	E	NS
F	E	E	E	E	E	F	NS
NS	NS	NS	NS	NS	NS	NS	NS

You require an overall mark of D or more to pass this module.