PROG2007 PROGRAMMING II

Summary

Title	Assessment 2
Type	Portfolio
Due Date	Monday 8th April 11:59 pm AEST/AEDT (start of Week 6)
Length	NA
Weighting	60%
Academic Integrity (See below for limits of use where GenAI is permitted)	GenAI May be used for this assessment. Please see the Academic Integrity section below for acceptable use of GenAI in this assessment.
Submission	Please see the Submission section below on how to submit your assessment.
Unit Learning Outcomes	This assessment task maps to the following ULOs: ULO2: design, implement, test, and debug simple programs in an object-oriented programming language. ULO3: demonstrate how to use class mechanisms to support encapsulation and efficient code. ULO4: analyse and determine appropriate data structures and iteration methods to support a solution.

Rationale

This assessment, focused on creating a Python-based weather data extraction bot, aligns with current educational objectives by emphasizing real-world applicability and comprehensive problem-solving skills. It integrates core programming elements—object-oriented design, data handling, error management, and collections usage—preparing students for professional challenges in fields reliant on accurate data, such as environmental science and urban planning. Moreover, the task encourages students to develop robust, maintainable software, mirroring industry demands where software reliability is crucial. The added requirement of a video presentation hones their ability to communicate complex technical concepts, enhancing their employability by equipping them with a skill set that is highly valued across various industries, thereby making this assessment a holistic tool for learning and real-world application.

Task Description

In this assignment, you will write a bot in Python that extracts and processes data from a website which contains weather information, much like the BOM site. The site has been created for this unit, given that we don't have permission or want to overrun the BOM site.

The assignment contains two programming parts as follows:

- In Part 1, you will design and implement an object-oriented program that reads weather data from a csv file, stores it in an appropriate collection, and performs some simple data analysis.
- In Part 2, you will modify your program from part one and add the ability to handle dates and times, log messages to a log file, and implement web scraping and data analysis capabilities using third party libraries.

This assignment also requires you to create a video explaining why you completed the assignment the way you did.

Task Instructions

Part 1

Visit the website at https://prog2007.it.scu.edu.au/weather and view the following pages:

- https://prog2007.it.scu.edu.au/weather/sydney.html
- https://prog2007.it.scu.edu.au/weather/melbourne.html
- https://prog2007.it.scu.edu.au/weather/brisbane.html

You will notice that all three pages have an identical format as follows:

- A location for the forecast.
- The date and time the forecast was last updated.
- A daily forecast covering 7 days from Friday the 8th of March until Thursday the 14th of March.

Create a csv file based on the three pages (Sydney, Melbourne, and Brisbane) that contains all of the data from the pages. You can use the following example as a template if needed or you can create your own template. This example contains a headings row along with the data for the first record on the Sydney forecast page:

```
Location,Updated Date,Forecast Date,Min Temp,Max Temp,Condition,Possible
Rainfall,Chance of any Rain,Forecast,Warning
Sydney,Forecast updated at 9:39 am EDT on Thursday 7 March 2024,Friday 8
March,Min: 21,Max: 28,Morning shower or two.,Possible rainfall: 0 to 1 mm,Chance
of any rain: 50%,Partly cloudy. Medium chance of showers in the morning and early
afternoon. Light winds becoming east to northeasterly 15 to 20 km/h in the late
afternoon then becoming light in the evening. Sun protection recommended from 9:30
am to 4:30 pm. UV Index predicted to reach 9 [Very High]
```

Once you have created the csv file with all of the data from the Sydney, Melbourne, and Brisbane pages, your job is to write an object-oriented program in Python with a minimum of five classes that:

- Reads the data in the csv file and uses the data to create appropriate objects.
- Stores all of the objects in appropriate collection(s).
- Uses a Set to perform some basic data analysis.

Things to think about:

- How will you handle and manipulate the strings in the csv file?
- How will you structure your classes and what relationships will the classes contain?
- What attributes should each class have and what data type should they be?
- How will you secure and validate the data in each class?
- How will you handle any potential errors in your program?

Two possible ways to approach the problem may be to:

- Have a forecast class with attributes for individual items in each forecast.
- Have an observation class and view the individual items in each forecast as an observation (similar to the sample project used in the Live Coding sessions).

Your mark will be determined based on:

- Your proficiency in handling and manipulating the strings in the CSV file into suitable data types e.g. ints or doubles for values, units of measurements, start time and end time that sun protection should be worn etc.
- How well you demonstrate **ALL** the class design concepts you have learned in Modules 1 and 2.
- How well you demonstrate the error handling concepts you have learned in Module 3.
- How well you demonstrate your ability to choose and implement appropriate collections based on the concepts you have learned in Modules 3 and 4.

Part 2

Make a copy of your program from Part 1. Update your program so that it:

- Scrapes the data from the Sydney, Melbourne, and Brisbane pages using the request library and Beautiful Soup instead of reading it from a csv file.
- Uses the datetime module for any date or time related data.
- Uses the logging module to output appropriate messages where applicable e.g. info, warning, error messages.
- Performs some basic data analysis using pandas instead of a set.

You mark will be determined based on:

- The criteria listed in the previous section.
- How well you demonstrate the concepts you have learned in Module 5.

Video

You are required to create a video explaining why you completed the assignment the way that you did. Your video should address each part of the assignment separately i.e. cover part one first, then part two.

You are not required to explain your code line by line. Rather for each part of the assessment your video should focus on the following:

- Your class design.
- Any techniques you've employed to secure and validate your data.
- Your error handling.
- Your use of collections.
- The data analysis you chose to do.

Resources

Everything that you need to know to complete this assessment was covered in:

- Programming 1
- Modules 1 through 5 in this unit.

Task Submission

You can use PyCharm or VSCode to complete the assignment and your assignment must run using Python 3. This is covered in the Getting Started section of MySCU.

The submission requirements are identical to assessment one as follows:

- Your submission must be named FirstName_LastName_A2. For example, Alex_Hendry_A2
- You are only required to submit your Python files and not your project files.
- You are required to use the WakaTime plugin and submit a screenshot of your WakaTime dashboard when submitting your assignment clearly identifying how much time you spent on each part of the assessment.
- Your video must be uploaded to your student OneDrive account and shared using a link.

All of the above points are covered in the submission instruction video for assessment one. The video covers how to submit your Python files, your screenshot of the WakaTIme dashboard, and your video. The only difference is the A2 in the submission name.

Please DO NOT start this assessment without watching the video.

Please note that all submission instructions in this assignment and the submission video must be followed EXACTLY, including the folder names you are instructed to use. Failure to do so may result a requirement to resubmit. The reason for this is as a programmer, you will often work as part of a team and will be required to follow design documentation. If the design parameters are not followed precisely, bugs will be introduced into the software when all of the individual components of the program are assembled.

Academic Integrity

At Southern Cross University, academic integrity means behaving with the values of honesty, fairness, trustworthiness, courage, responsibility and respect in relation to academic work.

The Southern Cross University Academic Integrity Framework aims to develop a holistic, systematic and consistent approach to addressing academic integrity across the entire University. For more information, see: SCU Academic Integrity Framework

NOTE: **Academic Integrity breaches include** unacceptable use of generative artificial intelligence (GenAI) tools, the use of GenAI has not been appropriately acknowledged or is beyond the acceptable limit as defined in the Assessment, poor referencing, not identifying direct quotations correctly, close paraphrasing, plagiarism, recycling, misrepresentation, collusion, cheating, contract cheating, fabricating information.

Use of GenAI

There are no limitations on using GenAI in this assignment providing you follow the following guidelines:

- You understand the code you are submitting and can explain it.
- You are not using any concepts that are not taught in the unit.
- All code that was created using GenAI must be clearly identified using a comment. The following are possible examples of comments you may use:
    - This method was created using GenAI.
    - This block of code was partly created using GenAI.
    - This constructor was generated using ChatGPT.
- Your code runs without errors.

If your submission is deemed to be suspicious by your marker you may be asked to attend an interview in your tutorial class to explain your code. If you cannot explain your code, you may be submitted for academic integrity. Possible reasons your submission may be deemed suspicious could include:

- Using programming concepts not taught in the unit.
- Using programming concepts considered by your marker to be beyond your programming abilities as demonstrated in the class.
- Submitting code suspected of being generated using GenAI software but is not clearly identified.

To summarise:

- **You can use GenAI as much as you like as long as you clearly identify the code you submit and understand what it does.**
- **DO NOT SUBMIT ANY CODE THAT YOU DO NOT UNDERSTAND AND CANNOT EXPLAIN.**

Collusion

Please note that your source code for this assignment will be run through a plagiarism detection system designed for code that compares all assignments and highlights identical or very similar submissions. If you are found to have colluded with other students, you will be submitted for academic integrity.

Special Consideration

Please refer to the Special Consideration section of Policy.
https://policies.scu.edu.au/document/view-current.php?id=140

Late Submissions & Penalties

Please refer to the Late Submission & Penalties section of Policy.
https://policies.scu.edu.au/view.current.php?id=00255

Grades & Feedback

Assessments that have been submitted by the due date will receive an SCU grade. Grades and feedback will be posted to the 'Grades and Feedback' section on the Blackboard unit site. Please allow 7 days for marks to be posted.

Assessment Rubric

Marking Criteria and % allocation	High Distinction (85–100%)	Distinction (75–84%)	Credit (65–74%)	Pass (50–64%)	Fail (0–49%)
Analyse and convert data types (ULO2 & 4) 20%	String data from CSV files is analysed and converted flawlessly, demonstrating an advanced understanding of data type conversion. Potential issues are proactively identified and resolved, ensuring optimal program efficiency and data integrity.	String data from CSV files is accurately converted, showcasing a well-developed understanding of data type conversion. Potential issues are effectively addressed, maintaining high program efficiency and data integrity.	String data from CSV files is consistently converted accurately, reflecting a solid grasp of data type conversion principles. Common issues are identified and corrected, supporting smooth program operation.	String data from CSV files is adequately converted with minor errors, indicating a basic understanding of data type conversion. Fundamental issues are addressed to sustain program functionality.	Struggles with accurate conversion of string data from CSV files, revealing limited understanding and compromising data integrity and program performance. Significant conversion issues remain unresolved.
Design and implement class concepts (ULO3) 40%	Demonstrates exemplary skills in designing and implementing sophisticated class structures, demonstrating deep insights into object-oriented concepts, and creating highly efficient, well-encapsulated code.	Demonstrates strong proficiency in class design and implementation, showing a thorough understanding of object-oriented principles and producing efficient, well-structured code.	Demonstrates competent ability in designing and implementing classes, effectively applying object-oriented principles to produce functional and organised code.	Adequately designs and implements class structures, demonstrating basic understanding of object-oriented concepts with code that meets essential requirements.	Struggles with designing and implementing class structures, showing limited understanding of object-oriented concepts, resulting in ineffective and disorganised code.
Apply error handling and validation strategies (ULO2) 20%	Applies comprehensive error handling and validation strategies, ensuring robust program resilience and flawless handling of all potential exceptions.	Applies effective error handling and validation strategies, significantly enhancing program stability and effectively managing most exceptions.	Adequately applies error handling and validation strategies, maintaining program stability and addressing common exceptions appropriately.	Applies basic error handling and validation strategies, covering essential exceptions and maintaining minimal program continuity.	Fails to apply effective error handling and validation strategies, resulting in program instability and inadequate management of exceptions.
Evaluate and utilise collections (ULO4) 20%	Evaluates and utilises advanced collection types, demonstrating strategic choices that optimise data management and program efficiency.	Evaluates and uses appropriate collection types, significantly enhancing data handling and program functionality.	Evaluates and employs suitable collection types, showing a good understanding of their impact on data management and program performance.	Adequately selects and utilises basic collection types, meeting fundamental requirements for data management and program operation.	Struggles to choose or use collections effectively, resulting in suboptimal data management and program performance.

Description of SCU Grades

High Distinction:

The student's performance, in addition to satisfying all of the basic learning requirements, demonstrates distinctive insight and ability in researching, analysing and applying relevant skills and concepts, and shows exceptional ability to synthesise, integrate and evaluate knowledge. The student's performance could be described as outstanding in relation to the learning requirements specified.

Distinction:

The student's performance, in addition to satisfying all of the basic learning requirements, demonstrates distinctive insight and ability in researching, analysing and applying relevant skills and concepts, and shows a well-developed ability to synthesise, integrate and evaluate knowledge. The student's performance could be described as distinguished in relation to the learning requirements specified.

Credit:

The student's performance, in addition to satisfying all of the basic learning requirements specified, demonstrates insight and ability in researching, analysing and applying relevant skills and concepts. The student's performance could be described as competent in relation to the learning requirements specified.

Pass:

The student's performance satisfies all of the basic learning requirements specified and provides a sound basis for proceeding to higher-level studies in the subject area. The student's performance could be described as satisfactory in relation to the learning requirements specified.

Fail:

The student's performance fails to satisfy the learning requirements specified.