ASSESSMENT 1 BRIEF	
Subject Code and Title	MIS501 Principles of Programming
Assessment	Coding Essentials - Test Programming Efficiency
Individual/Group	Individual
Length	800 words (+/- 10%) or equivalent including source code and comments
Learning Outcomes	The Subject Learning Outcomes demonstrated by successfulcompletion of the task below include: (a) Analyse challenging programming issues and apply programming concepts to design and develop efficient and scalable software solutions. (b) Use an integrated development environment (IDE) and industry-best project management practices to create, test, and debug code that addresses requirements for different organisational settings.
Submission	Due by 11:55pm AEST Sunday end of Module 2.2 (Week 4)
Weighting	20%
Total Marks	100

Task Summary

Design and develop distinct programming tasks addressing the business needs raised in the requested programming tasks within the context of the provided case study, using the concepts that we have learned in Modules 1 and 2.

Context

Using programming skills to make informed business decisions is becoming increasing important in the current information era. The speed in which this can be achieved is also desirable in the fast- paced business environment. Businesses and organizations rely heavily on computer programs to better understand and analyze data. This assessment assesses your skills in designing and developing computer programs using procedural programming techniques to address business needs.

Task Instructions

You are now to develop a program using Python. The program must comprehensively include all the functional requirements described in the case study. The program must also satisfy the conditions discussed in the Case Study.

There is NO minimum length to your Python codes, but you need to make sure to have enough functionality in your programs to address the business needs in the case study. Please remember:

- Please read the attached MIS501_Assessment 1_Business Case Study and complete the assessment tasks.

- All implementations must be in Python 3 (that is NOT Python 2). Programs implemented in a different language will be marked 0. Programs implemented in Python 2 will be capped at 50% of the available marks.
- You may only use the language features and syntax taught in Module 1.1 through to Module 2.2. You MUST not use any other language features beyond what was taught by Module 2.2. (Penalties apply).
- You may use any Python IDE to develop and test your Python programs.
- The program should be provided with adequate and meaning comments.
- Your program should be robust. Check for input validity.
    - Hint: What would you do if user chose option 3 without entering any user information first?
- Please follow Python Style Guide: https://www.python.org/dev/peps/pep-0008/
- You are strongly advised to read the rubric which is an evaluation guide with criteria for grading the assignment. This will give them a clear picture of what a successful final solution looks like.
- Review the Case Study briefing document, located in the assessment area of your learning portal and submit **only one .py file** to complete the assessment.

General Assessment Requirement

Incomprehensible submissions. Assessments provide the opportunity for students to demonstrate their knowledge and skills to achieve the required standard. To do this, assessment responses need to be both clear and easy to understand. If not, the University cannot determine that students have demonstrated their knowledge and skills. Assessments will, therefore, be marked accordingly including the potential for **0 (zero)** marks where relevant.

Case Study. Assessment response must focus on the hypothetical Case Study given in the Task Instructions. Any assessment items that do not address the case study may be awarded **0 (zero)** marks.

Check marking criteria. Before submitting your assessment, you should check it against the assessment criteria and the marking rubric included in this specification to ensure that you have satisfactorily addressed all the criteria that will be used to **mark** your submission.

Academic language. All submissions should be thoroughly proof-read for spelling, typographical or grammatical errors before being submitted. Do not reply on the 'spell-check' function in your word processing program. If, for example, 'affect' is substituted for 'effect', your program may not detect the error.

Referencing

No referencing required for this assessment.

Submission Instructions

- Means of submission. ALL students must submit **ONE python file (.py extension)** via the Assessment link in the main navigation menu in MIS501 Coding Essentials - Test Programming Efficiency.
- **Physical copies/Email** submissions are not accepted.
- Complete and correct submission. **Assessment,** once submitted, are FINAL and therefore cannot be modified. You bear all the onus to ensure that your submissions are final, correct (correct files in correct format) and complete before submitting to Blackboard.
- You are expected to begin this assessment when you begin the trimester, especially as you relate the learning activities (formative assessment) in the modules to this and the other (summative) assessments. Be sure to keep several drafts of your work as well as your notes and any sources you used to draw on in preparing your report.

- Extensions will be considered only in extenuating circumstances where the student has applied before the due date. At that point, students are required to provide the latest draft, in case the extension is not granted and to demonstrate they have earnestly done everything to avoid lateness.
- Students are responsible for keeping appropriate back-ups and drafts of their assignments and to submit the correct version.
- Torrens University Australia policies apply to the preparation and submission of this assignment.

Academic Integrity Declaration

I declare that except where I have referenced, the work I am submitting for this assessment task is my own work. I have read and am aware of Torrens University Australia Academic Integrity Policy and Procedure viewable online at http://www.torrens.edu.au/policies-and-forms

I am aware that I need to keep a copy of all submitted material and their drafts, and I will do so accordingly.

All students are responsible for ensuring that all work submitted is their own and is appropriately referenced and academically written according to the Academic Writing Guide. Students also need to have read and be aware of Torrens University Australia Academic Integrity Policy and Procedure and subsequent penalties for academic misconduct. These are viewable online. (http://www.torrens.edu.au/policies-and-forms)

Students also must keep a copy of all submitted material and any assessment drafts.

Special Consideration

To apply for special consideration for a modification to an assessment or exam due to unexpected or extenuating circumstances, please consult the Assessment Policy for Higher Education Coursework and ELICOS and, if applicable to your circumstance, submit a completed Application for Assessment Special Consideration Form to your Learning Facilitator.

Assessment Rubric: Coding Essentials - Test Programming Efficiency

Assessment Attributes	Fail (Yet to achieve minimum standard) 0-49%	Pass (Functional) 50-64%	Credit (Proficient) 65-74%	Distinction (Advanced) 75-84%	High Distinction (Exceptional) 85-100%
Use of Concepts **Percentage for this criterion = 25%**	Little to no evidence of using essential Python concepts. Significant misunderstandings or incorrect implementation of core concepts.	Basic understanding and usage of essential Python concepts, but with some errors or oversights. Some relevant concepts are applied.	Good grasp and application of fundamental Python concepts, with minor errors. Demonstrates an understanding of core concepts and some advanced topics.	Strong command and effective utilization of Python concepts. Shows a deep understanding of core and advanced topics, applying them accurately and efficiently.	Exceptional understanding and mastery of Python concepts. Demonstrates creativity and expertise in applying a wide range of advanced concepts to achieve complex and sophisticated solutions.
Errors **Percentage for this criterion = 25%**	Code contains numerous critical errors, preventing it from running or producing correct results.	Code has some errors that impede correct execution or produce inconsistent results.	Code mostly functions correctly but may have occasional minor errors or inaccuracies.	Code is mostly error-free, with only minor and inconsequential issues that do not affect functionality or results.	Code is virtually error-free, with exceptional attention to detail, resulting in a flawless and robust implementation.

Problem Solving **Percentage for this criterion = 25%**	No evidence of problem-solving skills. Code lacks a clear strategy or approach to the given task.	Basic attempt at problem-solving with some logic, but the solution may be incomplete or inefficient.	Effective problem-solving skills demonstrated. The code provides a clear strategy and addresses the task with a well-organized solution.	Strong problem-solving skills shown. The code exhibits an efficient and effective strategy, resulting in a well-optimized solution.	Outstanding problem-solving skills displayed. The code showcases creativity, ingenuity, and a comprehensive approach, delivering a superior solution that surpasses expectations.
Code Readability and Style **Percentage for this criterion = 25%**	The code is disorganized, poorly formatted, and challenging to read. Variable names are not meaningful.	Code is somewhat readable, but the structure and style could be improved. Some formatting and variable naming conventions are followed.	Code is well-organized and readable. Proper formatting and consistent variable naming enhance the readability.	Code is highly readable and follows excellent style conventions. The structure is clear and easy to comprehend, with meaningful variable names and consistent formatting.	Code is exceptionally readable, adhering to outstanding style conventions. The organization, formatting, and variable naming contribute to an exemplary level of clarity and readability.
The following Subject Learning Outcomes are addressed in this assessment					
SLO a)	Analyse challenging programming issues and apply programming concepts to design and develop efficient and scalable software solutions.				
SLO c)	Use an integrated development environment (IDE) and industry-best project management practices to create, test, and debug code that addresses requirements for different organisational settings				