INFO6001:
BLOCKCHAIN TECHNOLOGY AND APPLICATIONS

Assessment Details

Unit Code Title	INFO6001: BLOCKCHAIN TECHNOLOGY AND APPLICATIONS
Assessment #	Assessment 1
Assessment Type	Practical
Due Date	Start of Week 3
Weighting	20%
Length	N/A
Individual / Group	Individual
Academic Integrity	Contract cheating and any use of GenAI, such as ChatGPT, in this assignment are strictly prohibited. Any breach may have severe consequences. Please read carefully the "Academic Integrity" section below.
Submission	You will need to submit the following to the submission link provided for this assignment on the MySCU site: • The project with all your source code files • The link to your GitHub repository
Unit Learning Outcomes	This assessment task maps to the following ULOs: ULO1: discuss the concepts of distributed ledger and blockchain technology.
GenAI Use Level	Level 2. Purpose-Specific GenAI Use Permitted

Rationale

The purpose of this assessment is to test your ability to:

- Translating theoretical blockchain concepts into a functioning system.
- Writing and debugging code to implement blockchain features.
- Demonstrating understanding of the fundamental principles of distributed systems and their security implications.

By completing this assessment, you will gain critical skills needed in industries such as finance, healthcare, and supply chain, where blockchain is being increasingly adopted.

Task Description

Students will create a simple blockchain-based application that demonstrates their understanding of blockchain principles. This application should implement core features of a blockchain, such as creating blocks, chaining them together, and validating the chain.

Task Instructions

Your task in this assessment is to create a simple blockchain-based application using Python. The application must demonstrate fundamental concepts of distributed ledger and blockchain technology.

Create a Blockchain

- Each block in the chain must include:
  - A unique identifier (e.g., block number or timestamp).
  - Transaction or arbitrary data input from the user.
  - A cryptographic hash of the block's contents.
  - The hash of the previous block to maintain chain linkage.
- Provide functionality to:
  - Add new blocks to the blockchain.
  - Automatically calculate and store the hash for each block.
  - Ensure that blocks are linked via the hash of the previous block.

Validate the Blockchain

- Implement a method to verify the blockchain's integrity:
  - Check that all blocks are correctly linked.
  - Ensure no tampering has occurred by verifying the stored hashes.

User Interaction

- Implement a Python script that allows users to:
  - Input transaction data to add a new block.
  - Display the entire blockchain.
  - Run a validation check to ensure the blockchain's integrity.

You will also **create a repository on GitHub** to store your project work with all files and documents. You **must show your work progress** in this assignment by regularly committing your project to the GitHub repository. **Failing to show the correct work progress will fail the assignment.**

Task Submission

Submit the work to the provided submission link on the Blackboard learning site. You will submit:

- Your source code - A well-structured and commented Python script implementing the blockchain.
- The link to your GitHub repository

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.

At SCU the use of GenAI tools is acceptable, unless it is beyond the acceptable limit as defined in the Assessment Item by the Unit Assessor.

Purpose-Specific GenAI Use:

- GenAI tools used for specific assessment tasks or purposes as identified and scaffolded by the Unit Assessor.
- Grammarly, spell checking, and other referencing software may be used.
- Examples: Research Proposal, Programming, Map, Report, Thesis, Reflective writing, Curriculum Vitae, Infographic, Calculations.

For further information regarding conditions of use, speak to the Unit Assessor and refer to Generative AI for Students.

GenAI Use Declaration

You **must include one of the following statements** in your final submission, depending on whether or not you used Generative AI (GenAI) tools for this assessment.

A. If you **DID** use GenAI tools, include this statement:

*I acknowledge that I have used GenAI tools to complete this assessment. I used <**GenAI tool(s)**> to <**specific purpose(s) of using GenAI**> within the parameters outlined in the Assessment Brief and by the Unit Assessor.*

B. If you **DID NOT** use GenAI tools, include this statement:

I acknowledge that I have not knowingly used GenAI to complete this assessment.

IMPORTANT: Misuse of GenAI or failure to acknowledge its use may breach academic integrity rules. The Unit Assessor may also ask you to describe or demonstrate which GenAI tools you used, how you used them, and how your use complied with the assessment guidelines. Be ready to discuss this if asked.

Rules relating to Assessment and Examination

For further information regarding; extensions, special consideration, late submissions, resubmissions, grades, appeals and academic integrity, please refer to: Rules Relating to Awards - Rule 3 - Coursework Awards - Student Assessment and Examinations and How to apply for Special Consideration.

Academic Integrity Declaration

By submitting this assessment, I declare that:

I have read and understood SCU's Academic Integrity policies and referencing guidelines. I am aware of the consequences of academic misconduct and confirm that this submission is my own original work, referenced appropriately, and has not been previously submitted. I authorise its reproduction for authentication purposes and understand the implications of a false declaration. I have adhered to guidelines regarding Generative AI.

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%
Blockchain Implementation (ULO1) 35%	Fully functional blockchain implementation with advanced features (e.g., cryptographic hashing, timestamping, efficient block structure). Code is robust and modular.	Fully functional blockchain with basic features correctly implemented. Code is clear, modular, and error-free.	Blockchain implementation is functional but lacks optimization. Some minor issues in code structure or robustness.	Basic blockchain implementation with limited features. Some functionality missing or contains significant issues.	Blockchain implementation is non-functional or lacks essential components like hashing or chaining.
Blockchain Validation (ULO1) 30%	Validation process is efficient, handles edge cases, and ensures complete integrity of the chain. Includes tamper detection.	Validation process works correctly and ensures chain integrity. Handles common errors but misses some edge cases.	Validation process works but may fail for certain edge cases or lacks detailed implementation of tamper checks.	Validation process is basic and prone to errors. Integrity checks are incomplete or unreliable.	Validation is non-functional or fails to check chain integrity.
User Interaction (ULO1) 10%	Highly intuitive interface, allowing seamless input/output functionality. Detailed error messages and user guidance provided.	Interface is clear and functional. Handles user input/output correctly with minor usability improvements needed.	Interface is functional but lacks clarity. Error handling for user input is basic or incomplete.	Interface is functional but clunky. Minimal error handling or unclear instructions for users.	Interface is non-functional or confusing, with no error handling or unclear instructions.

Uses comments in the source code to communicate with stakeholders (ULO1) 5%	Effectively integrates detailed and precise comments within the source code, enhancing readability and providing insights into the logic and functionality of the code.	Integrates sufficient and explicit comments within the source code, improving readability and offering insights into the code's logic and functionality.	Integrates basic comments within the source code, contributing to readability but lacking in-depth insights into the code's logic and functionality.	Integrates minimal comments within the source code, resulting in some readability improvement but lacking meaningful insights into the code's logic and functionality.	Does not provide sufficient comments in the source code to improve readability.
Code Quality 20%	Code is clean, well-structured, and adheres to professional standards. Demonstrates efficient use of Python features and good modularity. Repository on GitHub was created, thoroughly documented and professionally maintained.	Code is clear, well-organized, and uses appropriate Python features effectively. Some minor improvements needed. Repository on GitHub was created, documented in detailed manner and properly maintained.	Code is functional but shows minor inefficiencies or lacks modularity in parts. Generally clear structure. Repository on GitHub was created, documented and maintained with acceptable standard.	Code is functional but poorly structured. Inefficient implementation or inconsistent adherence to Python practices. Repository on GitHub was created, documented and maintained with some limitations.	Code is poorly written, unorganized, and difficult to follow. Shows little adherence to Python best practices. Repository on GitHub was not created or was not documented and maintained as an acceptable standard.

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.