**Info about this Java Project**

Prompt Start

I am an undergraduate student that studies information technology. I do not know much about programming and would like to make a Java project to help build my knowledge. Can you be an assistant to help me make a project. I will provide more information in later prompts. Also clarify throughout the process if what was outputted okay. Just make a small, short response.

Context

I am nearly going to graduate and need to do 1 more advanced IT subject. I was advised to take a subject 1 semester as I couldn’t take a leave of absence.

*“If you pass your course in semester 2, 2024, you will be left with 1 more advanced IT option course. You can choose either of*

*Cloud Security or Blockchain fundamentals for this last course. Here are the course guides:*

* *Cloud Security:* [*http://www1.rmit.edu.au/courses/038407*](http://www1.rmit.edu.au/courses/038407)
* *Blockchain fundamentals:* [*http://www1.rmit.edu.au/courses/INTE26272450*](http://www1.rmit.edu.au/courses/INTE26272450)

*You mentioned that you want to take LoA for semester 1, because you want to take System Deployment & Operations (COSC2759).”*

* *Shekhar Kalra*

Here is the course information for Cloud Security, one of the courses I need to do enclosed in brackets:

{

**Course Title:**Cloud Security

**Credit Points:**12.00

**Course Description**

Cloud security is an evolving sub-domain of computer security, network security, and, more broadly, information security. It refers to a broad set of technologies deployed to protect data, applications, and the associated infrastructure of cloud computing.

This course provides a practical survey of both the principles and practice of cloud security. The emphasis of this course is on the underlying principles and techniques of cloud security with examples of how they are applied in practice.

Please note that if you take this course for a bachelor honours program, your overall mark in this course will be one of the course marks that will be used to calculate the weighted average mark (WAM) that will determine your award level. This applies to students who commence enrolment in a bachelor honours program from 1 January 2016 onwards.

**Program Learning Outcomes**

This course is an option course and not a core course, so that it does not need to contribute to the Program Learning Outcomes.

**Course Learning Outcomes**

Upon successful completion of this course you should be able to:

1. Demonstrate knowledge of cloud security principles and mechanisms
2. Demonstrate computer programming and configuration skills required to develop a cloud security infrastructure
3. Identify cloud security weaknesses by recognising and discovering threats and vulnerabilities to cloud computing
4. Problem solve how to fix cloud security weaknesses and mitigate security threats to cloud computing
5. Demonstrate knowledge and skills to prepare for industry cloud security certificate exams e.g. CCSK, CCSP
6. Communicate clearly and effectively using the technical language of the field correctly

**Overview of Learning Activities**

The learning activities included in this course are:

* Lectorials: Key concepts will be explained in lectorials in which course material will be presented and the subject matter will be illustrated with demonstrations and examples;
* Tutorial sessions focus on analysing and problem solving of given scenarios, such as security analysis and secure solutions;
* Computer laboratory sessions provide practices in the application of security techniques.

}

Here is information from one of tutors about what I should to prepare for the subject enclosed in brackets.

{

The programming tasks for this course are expected to be completed in Java or JavaScript. Generally speaking, and without going into too much detail, we use a programming language to apply a cryptographic concept, where the objective is to transform data into a form that appears random, but the original data can be recovered.

As a simplified example of this idea, consider that we have a character string like "Hello World!" and the task is to transform each character, such that the transformation can be reversed. For this task, the programming constructs we would need to use are:

* **String handling**: obtaining individual characters.
* **Basic data types**: understanding the relationship between character and integer data types.
* **IF statement:** can be used to handle characters like '!' and the space character as special cases.
* **FOR loop:** need a way to iterate over all characters in a string.
* **Arithmetic:** need to use a mathematical operation that can be reversed.

These constructs are found in either Java or JavaScript. In the case of JavaScript, we also need an API that allows us to get input from a webpage to make the implementation more interactive (that is, the Document Object Model that represents the HTML document). We typically avoid the use of language features like polymorphism or exception handling, because this can distract from the task at hand. For more advanced algorithms, we make use of programming libraries, which are provided as needed.

}

***Does this make sense? Just summarise what I provided concisely.***

# List of projects

**Beginner-Level Projects**

1. **Character Transformation Program**  
   *Difficulty: Easy*
   * **Goal**: Write a program that takes a string as input, applies a reversible transformation to each character (e.g., shift each character by a fixed number using ASCII values), and then reverses the transformation.
   * **Skills Covered**:
     + String handling
     + For loops and conditionals
     + Basic data types (e.g., char and int)
     + Arithmetic operations
2. **Basic File Encryption/Decryption**  
   *Difficulty: Easy*
   * **Goal**: Create a program that reads a text file, applies a simple encryption method (e.g., Caesar cipher), and writes the encrypted output to another file. Include a decryption option.
   * **Skills Covered**:
     + File handling
     + String manipulation
     + Loops and conditionals

**Intermediate-Level Projects**

1. **Password Hashing Simulator**  
   *Difficulty: Moderate*
   * **Goal**: Build a program that simulates password hashing using a cryptographic hash function like SHA-256 (use Java’s built-in libraries for hashing).
   * **Skills Covered**:
     + Using external libraries (java.security package)
     + Reading user input
     + Understanding hash algorithms
2. **Symmetric Encryption with a Key**  
   *Difficulty: Moderate*
   * **Goal**: Implement a basic symmetric encryption algorithm (e.g., AES). The program should encrypt and decrypt text using a shared key provided by the user.
   * **Skills Covered**:
     + Cryptographic libraries (e.g., javax.crypto)
     + Working with byte arrays and strings
     + Exception handling (basic)

**Advanced-Level Projects**

1. **Secure Cloud File Storage Emulator**  
   *Difficulty: Hard*
   * **Goal**: Create a simple program to encrypt a file before uploading it to a simulated cloud directory (use local storage as the "cloud"). Include options to decrypt and download the file.
   * **Skills Covered**:
     + File I/O
     + Encryption and decryption (e.g., AES)
     + Basic command-line interface
2. **Authentication and Token Generation System**  
   *Difficulty: Hard*
   * **Goal**: Build a basic authentication system where users enter their credentials, and the program generates a token (e.g., JWT). This token can then be validated.
   * **Skills Covered**:
     + String encoding/decoding (e.g., Base64)
     + Token creation and validation
     + Libraries for cryptographic functions

**Progression Plan**

1. Start with **Character Transformation Program** (1-2 weeks).
2. Move to **Basic File Encryption/Decryption** (2 weeks).
3. Transition to intermediate projects like **Password Hashing Simulator** (2 weeks).
4. Build your skills with **Symmetric Encryption with a Key** (2 weeks).
5. End with one of the advanced projects like **Secure Cloud File Storage Emulator** or **Authentication and Token Generation System** (4 weeks).

A screenshot of a computer program

Description automatically generated

## What is a module

A screenshot of a computer program

Description automatically generated A screenshot of a computer

Description automatically generated

# Know how to use classic tokens

<https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens>

* This is to replace passwords and make it more secure to commit changes. This is better as it allows read and write.

# Links

How to get started and create a Java File: <https://youtu.be/Cf2TFrZW2l0>

* + - This is shorter and better: https://www.youtube.com/watch?v=VDLgUo120\_I

How to add Eclipse to Git: https://www.youtube.com/watch?si=1ClVl322ltK6XHwu&v=LPT7v69guVY&feature=youtu.be

How to delete a repository: https://www.youtube.com/watch?v=u4qVRJOqMH0