Softwarica COLLEGE OF IT & E-COMMERCE | in collaboration with Coventry University

Student Assignment Brief

CONFIDENTIAL DOCUMENT

This document is intended solely for Softwarica College of IT & E-Commerce students for their own use in completing their assessed work for this module. It must not be passed to third parties or posted on any website.

Contents

Assignment Information

Module Name:	Programming and Algorithm 2
Module Code:	ST5062CEM
Assignment Title:	Coursework 2- Individual
Assignment Due:	8th March, 2026

Assignment Credit:	20 credits (CW2- 10 credit)
Word Count:	4000 words
Assignment Type:	Coursework
Grading:	Percentage Grade (Applied Core Assessment)

Assessment Overview

You will be provided with an overall grade between 0% and 100%. You have one opportunity to pass the assignment at or above 40%.

Important Notice

The work you submit for this assignment must be your **own independent work**. More information is available in the 'Assignment Task' section of this assignment brief.

Assessed Module Learning Outcomes

1. Reason about algorithm efficiency in order to select and implement the most appropriate for a given task
2. Evaluate patterns and paradigms appropriate for specific tasks
3. Develop secure software through the application of standards and secure programming principles
4. Create software that requires multi-threading, inter-process communication, memory management and close interaction with the host operating system
5. Develop software with a variety of user interfaces

Assignment Task

Individual Project:
The project is an individual based project that is handled by everyone. The project is specific to the students of Ethical Hacking and Cyber Security degree course to assess the module learning outcomes only.

Project description:

1. Student can choose their own project topic that requires the development of a software system that has multiple, distributed components, in which security is required for both data at-rest and in-transit, user authentication, prevention of MITM/replay/DoS attacks and so on.
2. The project will contain suitable interface (Command Line Interface / Graphical User Interface / Menu Driven Interface).
3. Should have appropriate testing of software (unit testing, integration testing).
4. Should work with Git and GitHub version control system.
5. Students are required to submit a recorded video and a written project report at the submission deadline.
6. The video submission of your application should be approximately 10 mins. Your video should be uploaded to YouTube as unlisted. A link for the video should be provided in the submission portal.
7. The report on the work should be up to 4000 words.

Example project ideas include:

- Prevention of MITM/replay/DoS attacks
- Antivirus
- Firewall protection
- Secure messaging application with end-to-end encryption
- Intrusion detection system
- Secure cryptocurrency wallets
- Malware Detection and Removal Tools
- Secure cryptocurrency exchanges
- Prevention of phishing attacks in voting system using visual cryptography

Notes:

1. Students are encouraged to use their own user-defined data structure rather than built-in data structure of given language.
2. You are expected to use the **APA 7th style** for referencing. For support and advice on this, students can contact Centre for Academic Writing (CAW).
3. Please notify your registry course support team and module leader for disability support.
4. The University cannot take responsibility for any coursework lost or corrupted on disks, laptops, or personal computers. Students should therefore regularly back up any work and are advised to save it on the University system.
5. If there are technical or performance issues that prevent students submitting coursework through the online coursework submission system on the day of a coursework deadline, an appropriate extension to the coursework submission deadline will be agreed. This extension will normally be 24 hours or the next working day if the deadline falls on a Friday or over the weekend period. This will be communicated via your Module Leader.
6. Collusion between students (where sections of your work are similar to the work submitted by other students in this or previous module cohorts) is taken extremely

seriously and will be reported to the academic conduct panel. This applies to both coursework and exam answers.

7. A marked difference between your writing style, knowledge and skill level demonstrated in class discussion, any test conditions and that demonstrated in a coursework assignment may result in you having to undertake a in person examination in order to prove the coursework assignment is entirely your own work.
8. If you make use of the services of a proof-reader in your work, you must keep your original version and make it available as a demonstration of your written efforts.
9. You must not submit work for assessment that you have already submitted (partially or in full), either for your current course or for another qualification of this college, except for resits, where for the coursework, you may be asked to rework and improve a previous attempt. This requirement will be specifically detailed in your assignment brief or specific course or module information. Where earlier work by you is citable, i.e., it has already been published/submitted, you must reference it clearly. Identical pieces of work submitted concurrently may also be self-plagiarism

Submission Instructions

Requirement	Details
File Naming	NAME_studentID
File Format	.pdf format
Submission Method	Campus 4.0 platform (submission link provided 2 weeks before deadline)

Marking and Feedback

How will my assignment be marked?

Your assignment will be marked by the Module Team using standardized criteria.

How will I receive grades and feedback?

Provisional marks will be released once internally moderated. Feedback will be provided alongside grades release within 2 weeks (10 working days).

What will I be marked against?

Details of the marking criteria for this task can be found in the Assessment Marking Criteria section at the end of this brief.

Grade Requirements

You must achieve 40% or above to pass this assessment. Ensure you understand the marking criteria for successful completion.

Assignment Support and Academic Integrity

Getting Help

If you have any questions about this assignment, please meet your respective module leader or teacher for more information.

Language Standards

You are expected to use effective, accurate, and appropriate language within this assessment task.

Academic Integrity

The work you submit must be your own. All sources of information need to be acknowledged and attributed; therefore, you must provide references for all sources of information and acknowledge any tools used in the production of your work, **excluding Artificial Intelligence (AI)**.

We use detection software and make routine checks for evidence of academic misconduct. Definitions of academic misconduct, including plagiarism, self-plagiarism, and collusion can be found in Student handbook in Campus 4.0.

All cases of suspected academic misconduct are referred to for investigation, the outcomes of which can have profound consequences to your studies.

Support for Students with Disabilities

If you have a disability, long-term health condition, specific learning difference, mental health diagnosis or symptoms, contact the Student Support Office for assistance.

Unable to Submit on Time?

If events prevent you from submitting on time, guidance on extenuating circumstances is available in the Student Handbook or from the Student Support Office.

Administration of Assessment

Module Leader Name:	Er. Suman Shrestha
Module Leader Email:	stw0023@softwarica.edu.np
Assignment Category:	Written
Attempt Type:	Standard
Component Code:	CW

Assessment Criteria

	40% >=above	50% >= above	60% >= above	70% >= above
Report and Video (30%)	The application is fully functional, and the explanation of the code is clear. No persistence	Application coding follows the specific programming language coding convention. No persistence.	Application uses the different Structure and use of library given by the programming language. Uses suitable persistence storage mechanism (either file or database)	Application uses a third-party library and the techniques. Uses suitable persistence storage mechanism (both file and database)
Functionality (Use of algorithms) (30%)	Application that uses different functions for different tasks.	Each function meets the specifications given.	Each function follows the convention of programming language. i.e., docstring, comments	Uses of functions that use their own logic rather than using in-built function and having efficient time complexity.
Layout and Design (Interface) (20%)	Use of CLI and properly functioning layout and design.	Use of GUI and properly functioning layout and design.	Use of GUI with appropriate user interface. Form based interface.	Use of Form based interface having clear, consistent and user-controlled interface.
Version Control and Testing (20%)	Use of GitHub repo and unsuccessful attempt has been made to write simple tests.	Use of GitHub repo with multiple commits. Having limited number of tests written and run.	Use of GitHub repo with regular commits over an extended period of time with a range of tests showing how they contribute to code quality.	Use of branching and merging. A full suite of unit testing ensures full code coverage.

Mark allocation guidelines to students: marking rubric