# CP472: Assignment 1

## Overview

This assignment is designed to enhance your understanding of key programming concepts including algorithm implementation, data structures, and different programming paradigms. You will work with different languages (Python, Java, and C) to solve a variety of problems.

## Tasks

1. **Bubble Sort in Python**
   - Write a bubble sort algorithm in Python, but using the style and constraints of a language like FORTRAN or C (early versions), which did not support object-oriented programming and had more limited built-in functions. Handle errors or exceptions using basic conditional statements, avoiding Python's try-except constructs.
   - Constraints:
     - Do not use Python's built-in sorting functions or any other advance features.
     - Use basic fixed-size arrays similar to older languages instead of Python lists or other advanced data structures.
   - Deliverable: Source code.

2. **Fibonacci Numbers in Java**
   - Write a program in Java to calculate Fibonacci numbers, but in the style of early procedural languages, which focused on step-by-step instructions and had limited abstraction.
   - Constraints:
     - Do not use recursion, as early languages often did not optimize well for recursive calls. Focus on iterative solutions with explicit control structures like loops.
     - Minimize the use of Java's Object-Oriented features.
     - Avoid using Java's built-in classes or methods.
   - Deliverable: Source code.

3. **Matrix Multiplication in Python**
   - To understand the value of counting loops, implement matrix multiplication in Python using both counting loops (for loops) and control loops (while loops).
   - Comparison: Analyze both implementations in terms of readability, efficiency, and code length.
   - Deliverables:
     - Source code for both methods and a comparison report based on the given criteria.

4. **Student Database System in C**
    - To understand the value of records (structs), implement a simple student database system in C, first without using structs, and then using structs.
    - The database should include student ID, name, and age.
    - Functions required: Add new student data, retrieve a student's information by ID, and display all student data.
    - Deliverables: Source code for both versions, and a report comparing them in terms of data organization, code maintainability (e.g., how easy it is to add a new attribute such as grade), and readability.

## Marking Scheme

| Task | Marks |
|---|---|
| Task 1 | 8 |
| Task 2 | 6 |
| Task 3 (programming) | 10 |
| Task 3 (report) | 3 |
| Task 4 (programming) | 12 |
| Task 4 (report) | 3 |
| Total | 42 |

## Submission Guidelines

- Submit all source code files and report as a single compressed file.
- Ensure that each file is properly named.
- Report(s) should be concise, well-organized, and demonstrate your understanding of the concepts. The maximum allowable size is 3 pages.

This assignment will help you develop a deeper understanding of different programming constructs, algorithm design, and the use of data structures in various programming scenarios. Focus on clean, efficient coding practices and clear, insightful analysis in your report.