



"Investing in Africa's Future"

COLLEGE OF ENGINEERING AND APPLIED SCIENCES (CEAS)

NCSE 103: OBJECT-ORIENTED SOFTWARE DEVELOPMENT

END OF SECOND SEMESTER EXAMINATIONS

APRIL/MAY 2025

LECTURER: MR BRAITON U. MUKHALELA

TIME: 3 HOURS

INSTRUCTIONS

You are required to answer questions as instructed in each section

1. Complete all sections in this Examination.
2. Submit a zip file containing:
 - Source code
 - UML diagrams
 - Test reports/screenshots
3. **Code must follow OOP principles and best practices.**

Section A: UML Modelling

(25 Marks)

1. Design a **Library Management System** with the following:
 - **Classes:** Book, Member, Librarian, BorrowTransaction
 - **Relationships:** Inheritance (e.g., Librarian inherits from Member), Associations
2. Create the following UML diagrams using **ArgoUML**:
 - **Class diagram**
 - **Sequence diagram** for book borrowing
 - **Use case diagram**
3. Submit the UML diagrams in **.png or .pdf format**.

Section B: Object-Oriented Implementation

(25 Marks)

1. Implement the **Library Management System** in **Java/Python**.
2. Apply OOP principles:
 - **Encapsulation:** Private attributes with getter/setter methods
 - **Inheritance:** Librarian should inherit from Member
 - **Polymorphism:** Implement method overriding for different user roles
3. Ensure the code is well-structured and follows **best practices**.

Section C: Design Patterns

(15 Marks)

1. Implement the **Singleton pattern** to ensure only one instance of the **Library Database Manager** exists.
2. Use the **Factory pattern** to create different types of books (e.g., Fiction, Non-Fiction, Science).
3. Submit **code snippets** and an explanation of how the patterns improve code structure.

Section D: Testing

(20 Marks)

1. Write **unit tests** for the following using **JUnit/Pytest**:
 - Adding a new book
 - Borrowing a book
 - Returning a book
2. Run the tests and submit:
 - **Test script**
 - **Screenshot of test results**

Section E: AI Integration

(15 Marks)

1. **Integrate AI** to recommend books based on borrowing history.
 - Example: If a user borrows "Python for Beginners," suggest "Advanced Python."
2. Implement a **basic recommendation system** using a **simple algorithm** (e.g., frequency-based suggestions).
3. Submit the AI module code and test results.

END OF EXAMINATION