

**Deadline:02/02/2025 Sunday 11:59PM**

## Objective

In this assignment, you will design and implement a simple inventory management system using the Object-Oriented Paradigm (OOP). You will complete the implementation in both Python and C++ and compare the differences in design, syntax, and behavior between the two languages.

## Tasks

### Task 1: Object Oriented Analysis and Design

Read the story below carefully and analyze the e-commerce app's design. Based on the details provided, identify the key components of the system. Determine the **classes** that represent different entities in the app, as well as their properties and functionalities. Think about how the app is structured: What are the main entities (e.g., products, categories)? What characteristics do they have (e.g., name, price, size)? What actions can they perform or have performed on them (e.g., browsing, calculating discounts)? Clearly outline the classes, their relationships, and any behaviors they exhibit, ensuring that your design aligns with the object-oriented programming principles of encapsulation, inheritance, and abstraction.

### Sporting Goods Inventory System

Imagine you are designing an e-commerce app for a sporting goods store. This store has a variety of products divided into two main sections: **Indoor Sports** and **Outdoor Sports**. Each section caters to different types of activities and has unique characteristics and functionalities.

### The Indoor Sports Section

The indoor sports section includes equipment for activities like **Volleyball** and **Table Tennis**:

- **Volleyball:** This section features items like volleyballs and nets. Each volleyball has specific qualities like its material and size, and the nets vary by type (e.g., training or professional). Customers can browse through these items, and the store occasionally highlights special offers on this gear.
- **Table Tennis:** Here, customers can find paddles, balls, and tables. The paddles come in different styles, and tables are available in various materials. Customers can check the stock to ensure a full game setup is possible before making a purchase.

### The Outdoor Sports Section

The outdoor sports section offers gear for activities such as **Camping** and **Soccer**:

- **Camping**: The store provides camping essentials like tents, which vary in size and durability. Each tent also has a weather rating to indicate how well it handles rain or wind. You can add a few more products like sleeping bags or portable stoves.
- **Soccer**: This section includes soccer balls and goal nets. Balls are made from different materials, and goal nets come in various sizes. For soccer teams, the system can calculate discounts based on the number of players purchasing gear together.

### How It All Comes Together

The inventory is organized in a way that allows for seamless browsing and management. Every product is cataloged with essential information, such as its name, price, and availability. Customers can search for specific items, and the system ensures smooth updates to stock levels whenever products are sold or restocked.

Additionally, the app allows the store to group products under appropriate categories and subcategories, making it easy to explore and manage inventory for both staff and customers. Your task is to bring this concept to life using object-oriented programming principles!

It is not required but strongly suggested to create your UML diagrams before you start implementing (especially, class and use case diagrams)

\*\*\* We have provided you with a broad overview of how the e-commerce app is structured and how it functions. Use this as a foundation to identify the key components and relationships, but feel free to expand on the details. You can decide the specific properties and functionalities for each class based on the story, as long as they align with the overall purpose of the app. Additionally, you are welcome to introduce new categories, subcategories, or product types to enhance the system's functionality or showcase your creativity. Ensure your implementation reflects the core principles of object-oriented programming and results in a cohesive, extendable design.

### Task 2: Implementation in Python

- Implement the system in Python using its dynamic typing and built-in data structures.
- Ensure proper handling of attributes, encapsulation, and user interaction.

### Task 3: Implementation in C++

- Implement the system in C++ with a focus on memory management and type safety.
- Use vectors or lists to manage collections and demonstrate OOP principles in C++.

### Task 4: Comparison Report

After completing both implementations, write a report (minimum 2 pages) comparing the two languages based on the following criteria:

- Ease of implementation.
- Code readability and maintainability.
- Handling of object-oriented concepts (e.g., encapsulation, inheritance).
- Error handling (e.g., Python exceptions vs. C++ try-catch blocks).
- You can also include your UML diagrams (if you have)

**\*\*\* CS501 Students only:** Graduate students must include two additional paragraphs comparing Python and C++ in terms of garbage collection, memory management, and object-oriented programming (OOP) features. The comparison should clearly specify which OOP features are supported or not supported by each language, providing detailed examples or explanations to illustrate the differences.

### Deliverables

1. **Python Implementation: (Canvas & Github)**
  - Include the .py file with your Python code.
2. **C++ Implementation: (Canvas & Github)**
  - Include the .cpp file with your C++ code.
  - Provide a Makefile
3. **Comparison Report: (Canvas)**
  - Submit a PDF document with your observations and comparisons.
4. **Independent Completion Form (Canvas)**
  - Sign and submit the ICF (late submission -10 points, no exceptions)

### AI Usage

You are encouraged to use AI tools to assist with the coding portion of this assignment, but only for partial help, such as generating snippets, debugging, or exploring alternative implementations. Ensure that you use only UAB-approved AI tools and include screenshots of the prompts you used and the corresponding AI-generated results as part of your submission. However, remember that you cannot rely on AI to build the entire project; the majority of the implementation should reflect your understanding and effort. Additionally, the written report comparing the use of OOP principles in Python and C++ must be your personal work and should not involve any AI assistance. This report is a crucial component of the assignment, emphasizing your ability to analyze and articulate the differences between the two languages.

## Important

We aim to use this assignment as an opportunity to deepen your understanding of object-oriented programming (OOP) principles and their application across two different languages, Python and C++. As you design and implement the e-commerce app, incorporate core OOP principles such as inheritance to create hierarchies for categories and subcategories, polymorphism to define shared behaviors that can differ across product types, abstraction to encapsulate the system's complexity within clean interfaces, and encapsulation to protect the internal state of objects.

Your primary task is not just to code but to analyze and reflect on how OOP principles are implemented and utilized in Python versus C++. Consider aspects like syntax, features, and language-specific mechanisms (e.g., dynamic typing in Python vs. explicit typing in C++, garbage collection vs. manual memory management). Document your observations on the differences in designing, implementing, and testing OOP systems in these languages to gain insights into their strengths, limitations, and real-world applications.

## Grading Rubric

Criteria	Points
Python Implementation	20
C++ Implementation	20
Features and Functionality	20
Comparison Report	40
<b>Total</b>	<b>100</b>

## \*\*\* HW Submission Guide

Due date for each assignment will be written on the assignment document. Any assignment turned in after the deadline is considered late and will receive a score of zero. No late submission is allowed in this course. You need to include Independent Completion Form (ICF) for each assignment submission. Failing to submit ICF on time will result **in 10% penalty** in the homework grade.

Each homework submission must be uploaded to Canvas and also requires a corresponding submission on **GitHub** to demonstrate version control practices. You are expected to maintain a private repository for your homework assignments and commit your changes regularly. Once your work is complete, share your repository with the following GitHub accounts: **akshar2020**, **michaelgathara** and **uab-unan** Failure to submit on either platform may result in a deduction of points."