

CSC202 Programming 2

Assignment 1- Spring 24-25

Student Name	
Student ID	
Submission Guideline: <ul style="list-style-type: none">1) Each student must submit a PDF file containing clear and well-organized Java code of all class implementations, main class execution, and screenshot of program output.2) Submit a zip file of the java code.3) Late submissions will be penalized by 5% per day after the deadline.4) It is not allowed to resubmit graded assignments.5) Plagiarized work will be reported to OAI.	
Due date: Monday 7 April 2025 – 11:59 PM	

Objective:

Demonstrate a strong understanding of object-oriented programming concepts in Java, including **classes, objects, encapsulation, association, and the use of ArrayList**.

Assignment Requirements:

1. **Assignment Scope:**
 - Develop a Java program that models a **Car rental System** with a suitable level of complexity.
2. **Program Structure:**
 - Implement at least three interconnected classes and a **main class** to drive the application.
 - Ensure proper **encapsulation** and **association** between classes.
 - Use **ArrayLists** to manage collections of objects.
3. **Functionality & Features:**
 - Implement a **menu-driven interface** that allows users to:
 - **Add** new objects
 - **Search** for specific objects
 - **Remove** objects
 - **Sort** objects based on relevant criteria
 - **Display** objects and their related information
4. **Creativity & Originality:**
 - Your solution must be **entirely original**—avoid using AI-generated or pre-existing code.

- Focus on structuring your code logically and applying best programming practices.

5. Code Explanation Video:

- Record a **short video** explaining your code and how it works.
- Showing your face in the video is **optional**.

Grading Rubric

Criteria	Excellent (5 pts)	Good (3.5 - 4 pts)	Satisfactory (3 – 2.5 pts)	Needs Improvement (2 - 1 pts)	Not Attempted (0 pts)
Program Structure	Implements at least 3 well-structured, interconnected classes with proper encapsulation and association.	Implements 3 classes with mostly correct structure, minor issues in encapsulation or association.	Implements required classes but lacks proper structure or encapsulation.	Less than 3 classes, poor organization, or missing key concepts.	No attempt at class design.
Use of ArrayList	Effectively utilizes ArrayList for object storage and manipulation.	Uses ArrayList correctly but with minor inefficiencies.	Uses ArrayList but with noticeable errors or improper usage.	Minimal or incorrect use of ArrayList.	No ArrayList usage.
Functionality & Features	Implements all required menu functions (add, search, remove, sort, display) with no errors.	Implements most menu functions, minor errors present.	Implements some functions, but others are incomplete or non-functional.	Only one or two functions work correctly.	No functional menu implementation.
Creativity & Originality	Solution is unique, well thought-out, and demonstrates creative	Mostly original with some unique elements, but some features may be common.	Solution is somewhat generic, minor originality but lacks depth.	Lacks originality, closely resembles existing solutions.	No evidence of originality, copied or AI-generated content.

	problem-solving.				
Code Readability & Documentation	Code is well-structured, formatted, and includes meaningful comments.	Code is mostly well-structured but lacks some documentation.	Code is somewhat readable but lacks comments or proper formatting.	Code is difficult to read with little to no documentation.	Code is unorganized and unreadable.
Code Explanation Video	Clear and well-explained video demonstrating code logic, structure, and features.	Video provides a reasonable explanation with minor gaps.	Video is unclear or lacks detailed explanation.	Video is incomplete or poorly structured.	No video submission.

Total: __/5 points