420-921-VA - Programming Concepts 1
Course Project: Event-Driven Simulation

Overview

This project aims to bring together the concepts and techniques learned in the course by designing and implementing an **event-driven simulation** in Java. You will develop a complete, functional application using object-oriented principles and Java programming features. The goal is to simulate a dynamic system where *events trigger changes in state or behavior*, reinforcing your problem-solving and programming skills.

Project Requirements

Core Requirements

- **Object-Oriented Programming:**
  - Design your program using **classes** and **objects**.
  - Implement interactions between objects to simulate the system's behavior.
- **Java Programming Features:** (use as many as as possible)
  - Use **variables, operators, and expressions** for data processing.
  - Apply **decision structures** (e.g., `if-else`, `switch`) for conditional logic.
  - Implement **loops** (`for`, `while`, `do-while`) to handle repetitive tasks.
  - Write **methods** for modular and reusable code.
  - Work with **arrays** or **collections** to manage data.
  - Incorporate **file operations** (e.g., reading from and writing to files) for data persistence.
- **Event-Driven Simulation:**
  - Design a program that models a dynamic system with actions or events driving changes in state.

  - Interaction can involve:
    * **Players** (e.g., human vs computer or two-player interactions).
    * **Entities** within the simulation (e.g., customers and tellers in a bank system).

Design and Development

- Clearly define the **problem statement** and solution scope.
- Develop a **description** that includes:
  - Classes and objects required.
  - Interaction flow and event handling.
- Write **clean, well-documented code** with meaningful comments and clear structure.

Project Objectives

- Strengthen your understanding of **object-oriented programming**.
- Apply Java programming concepts in a practical and integrated way.
- Develop problem-solving skills by designing and implementing an application from scratch.
- Improve your ability to write **modular, reusable, and maintainable code**.

Recommended Project Ideas

Sample Ideas

1. **Simple Poker Game**
   - Simulate a basic two-player poker game.
   - **Key Features:** Deal cards to players, evaluate hands, determine the winner.
   - **Programming Concepts:** Arrays for cards, loops for gameplay, file I/O for saving scores.
2. **Bank Teller Simulation**
   - Model a bank where customers queue for services, and tellers process transactions.
   - **Key Features:** Handle deposits and withdrawals, simulate queue management.
   - **Programming Concepts:** Classes for customers and tellers, loops for queue processing, file I/O for transaction history.

3. **Quiz Game**
    - Create a single-player or two-player quiz with dynamic scoring.
    - **Programming Concepts:** Arrays for questions, decision structures for scoring, methods for modularity.

Other Ideas

- **Traffic Light Simulation:** Control traffic lights at an intersection with timed changes or random events.
- **Virtual Library:** Manage book borrowing, returns, and user accounts in a simple library system.
- **Basic Inventory System:** Track inventory levels, add/remove items, and generate reports.

Deliverables

1. **Design Document:**
    - Problem statement.
    - Description of the classes, methods and object interactions.
    - Optional: Class diagrams as a bonus.
2. **Source Code:**
    - Submit clean, well-documented `.java` files.
    - Include comments explaining your logic.
3. **Demonstration of Output:**
    - Provide a sample output file with screenshots and a screen recording showing the program in action.

Steps

1. Select a project idea and write the design document.
2. Implement basic classes and methods.
3. Complete functionality, integrate features and test the program.
4. Submit the project and the demo.

Grading Rubric

Criteria	Points
Functional Implementation	40
Object-Oriented Design	20
Use of Java Programming Features	20
Code Quality (Comments, Structure)	10
Design Document	10
Total	**100**

Submission Instructions

1. Compress your project folder into a `.zip` file.
2. Upload the file to the course portal by the **Dec 6, 2024**.

Final Notes

This project is your opportunity to apply the skills you have learned in the course to a practical problem. Choose a project idea that excites you, and focus on creating clean, functional code. If you have any questions, feel free to ask for guidance.