**ICS4U1 Final Project**

For the final project, you will be creating a database system. Examples of such systems are:

* booking system (e.g., room reservation, doctor appointments, computer lab)
* lookup system (e.g., real estate property lookup, product search)
* accounting system (e.g., family budget)
* tracking system (e.g., sport team database for the school)
* simulation (e.g. solar system)
* game (e.g. board game)

The system must perform a minimum of these functions:

* Add and delete records
* Modify records
* Search records using at least two different criteria (not necessarily simultaneously)
* Sort records using at least one different criteria
* Read and write records from and to plain text files

In addition, the solution to your system must correctly use the following programming concepts:

* Object-oriented design
  + Encapsulation
  + Polymorphism
  + Inheritance
* Array of objects (ArrayList)
* File input/output
* Sorting
* Searching

The system should have a friendly user interface. **Your system should also handle all errors without crashing.**

You must go through every stage of the [Software Development Life Cycle](https://docs.google.com/document/d/1WXlLGm8uMU6zPyVqxTj1b1BYQFRB51pDucj8rPL5vSA/edit?usp=sharing) to plan and create the software. The following Stages guide you through this cycle.

**Stage 1: Analysis**

The Analysis stage focuses on the problem, and the proposed solution to solve the problem. The purpose of the analysis stage is to perform tasks which formulate a precise understanding of the problem. The conclusion of this section defines the solution decisions made including defining the program's input and output, and its exact functionalities and behaviours.

1. ***Analysing the problem & Criteria for success***

This section should briefly describe the problem and clearly state the objectives/goals of the solution to the problem. The objectives/goals should not be too vague to measure (e.g. “user friendliness” should not be a goal because it is hard to measure). The goals should be “testable” and/or “provable”.

For example, the goals for a bank’s client transaction software may be:

* store personal client info
* store history of account transactions
* store current balance
* search for client information by names
* sort client information by account numbers
* calculate number of NSF checks
* calculate average balance
* display client info
* report clients with balances over x amount of dollars

For increased clarity, the goals can be described with concrete examples of solution behaviour with a wide variety of data, e.g., given data x, the solution will produce the result y.

These criteria for success will be referred to in the subsequent testing stage.

The first part should be a one-paragraph "product pitch" about what your program is and does, as if you were selling it to a customer.

1. ***Project Delegation (pairs only)***

A detailed plan for the equitable delegation of duties, where each partner is *coding* no less than one-third of the total program.

**Evaluation Rubrics**

1A **Analysing the problem & Criteria for success**

| **0** | The student has not reached a standard described by any of the descriptors given above. |
| --- | --- |
| **1** | The student states **some** objectives of the solution. |
| **2** | The student describes **most** of the objectives of the solution. |
| **3** | The student describes **all** of the objectives of the solution. |
| **4** | The student **relates** all of the objectives of the solution to the **analysis** of all aspects of the problem. |

1B **Project Delegation (pairs only)**

| **0** | The group has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The group includes a **limited** delegation plan of the program |
| **2** | The group includes a **basic** delegation plan of the program |
| **3** | The group includes a **detailed** delegation plan of the program |
| **4** | The group includes a **detailed and well-presented** delegation plan of the entire program |

## **Due Date for Stage 1: Dec 18, 2023**

**Stage 2: Design the Program**

Note that the following order of the criteria does not imply you have to develop or document your design in this sequence

1. ***Data Structures***

At the design stage, data structures should be chosen to fully support the data-storage requirements of the problem, and allows clear and efficient algorithms to implement the objectives specified in criterion 1A. The classes chosen should be logical in that the data is sensible for the objects in question and the methods are appropriate for the data given.

This section of your document should include the following

* *format of input / output files*
* *file structure (how your files will be organised)*
* *definition of each class in English: its purpose, all of its fields (~~their data type and structure~~) and methods*

1. ***Algorithms (pairs only)***

At the design stage, algorithms should be chosen to fully support the process needed to achieve the objectives specified in criterion 1A and provide sufficient support for the required data structures.

You must include the purpose of all methods in each class, including their parameters, return values, and descriptions.

This section can be a list or outline of all the algorithms, presented as text, *possibly* in outline format (pseudo code).

It is not necessary to describe every algorithm in detail. The following algorithms require only a brief description:

* *Standard searching and sorting algorithm: you may want to discuss how your choices are adapted for use with the properties of the data set or your selected data structure*
* *Accessor and mutator methods, except where important processes may be carried out (e.g. validation)*

1. ***Modular Organisation***

At the design stage, modules should be chosen that incorporate the data structures and methods required for the solution (criteria 2A and 2B) in a logical way. This organisation must be presented in a structured way to clearly show connections between modules, algorithms and data structures.

This section should include the UML diagram (see UML handout and past lesson) of each module (class) to describe its data members (with specified data structures) and methods, and showing their connections.

**Evaluation Rubrics**

1. **Data Structures**

| **0** | The student has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The student has outlined some of the data structures/types to be used in the solution. |
| **2** | The student has described some of the data structures/types to be used, and provided sample data. |
| **3** | The student has discussed most of the data structures/types to be used, and provided sample data. |
| **4** | The student has discussed and clearly illustrated all of the data structures/types to be used to solve the problem, and provided sample data for all of them. |

1. **Algorithms(pairs only)**

| **0** | The student has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The student has **outlined some** of the algorithms to be used in the solution. |
| **2** | The student has **described most** of the algorithms to be used, with details of parameters and return values. |
| **3** | The student has **discussed all** of the algorithms to be used, with details of parameters, return values. |
| **4** | The algorithms discussed are sufficiently **logical, detailed** and **well documented** to be used to create the solution in Java. |

1. **Modular Organization (UML)**

| **0** | The student has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The student has **outlined some** of the modules to be used in the solution. |
| **2** | The student has **described most** of the modules to be used, **showing connections** between them. |
| **3** | The student has **described all** of the modules to be used, and has shown the connections to data structures and methods. |
| **4** | The modules discussed are sufficiently **logical, detailed** and **well documented** to be used to create the solution in Java. |

## **Due Date for Stage 2: Dec 21, 2023**

## **Stage 3: Code the Program**

1. ***Using good programming style***

Good programming style can be demonstrated by program listings that are easily readable. These would include the following:

* clearly separated objects
* small and clearly-structured Java methods
* full Javadoc comments for classes and methods
* sufficient and appropriate comments describing
  + code that is difficult to understand
  + constant and variable declarations
* meaningful identifier names
* consistent indentation scheme and white space

1. ***Usability***

Your program should have user-friendly features such as helpful menus, help instructions, and useful guidance to the user during the execution of the program.

1. ***Handling errors***

Your program is expected to detect and reject erroneous data input from the user, and prevent common run-time errors caused by calculations and data-file errors.

1. ***Success of the program***

The program is considered successful if all objectives and goals listed in 1A are achieved.

The implementation of your program must correctly use the following programming concepts:

* Object-oriented design
  + Encapsulation
  + Polymorphism
  + Inheritance
* Array of objects (ArrayList)
* File input/output
* Sorting
* Searching

**Evaluation Rubrics**

1. **Using good programming style**

| **0** | The program has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The program demonstrates **little** attention to good programming style. |
| **2** | The program demonstrates **some** attention to good programming style. |
| **3** | The program **mostly** demonstrates attention to good programming style. |
| **4** | All parts of the program demonstrate **considerable** attention to good programming style. |

1. **Usability**

| **0** | The program has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The program has very **few** user-friendly features. |
| **2** | The program has **some** user-friendly features. |
| **3** | The program is **reasonably** user-friendly |
| **4** | The program is **extremely** user-friendly and it **meets** the usability objectives in section 1B. |

1. **Handling errors**

| **0** | The program has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The program facilitates **little** error-handling |
| **2** | The program facilitates **some** error-handling |
| **3** | The program facilitates **sufficient** error-handling |
| **4** | The program facilitates error-handling for **each** input and output method |

1. **Success of the program, part 1**

| **0** | The student has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The program **functions minimally**. It successfully achieved **few** of the objectives from section 1A. |
| **2** | The program **functions partially**. It successfully achieved **some** of the objectives from section 1A. |
| **3** | The program **functions mostly**. It successfully achieved **most** of the objectives from section 1A. |
| **4** | The program **functions fully**. It successfully achieved **all** of the objectives from section 1A. |

3D **Success of program, part 2**

| **0** | The student has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The student correctly uses **few** required concepts in their program. |
| **2** | The student correctly uses **some** required concepts in their program. |
| **3** | The student correctly uses **most** required concepts in their program. |
| **4** | The student correctly uses **all** required concepts in their program. |

## **Due Date for Stage 3: Jan 17, 2024**

## 

## **Stage 4: Test and Document the System, Project Review**

1. ***System testing***

Thorough testing must be done to ensure that the software achieves all goals and objectives defined in 1A. A test plan must be carefully designed to test all branches of the program with different sets of valid and invalid data. Testing with a well designed test plan can guarantee that the software will function as expected under all foreseeable circumstances.

You may wish to break the test plan down into testing phases and organize them in a table. Here is an example test plan for a database:

| **#** | **Functionality** | **Purpose** | **Input** | **Expected Output** | **Output** |
| --- | --- | --- | --- | --- | --- |
| 1 | General | Grammar correct on all screens |  |  |  |
| 2 | General | Spelling on each screen correct |  |  |  |
| 3 | Main Menu | All options lead to the correct screen |  |  |  |
| 4 | Invalid entry issues an error message |  |  |  |
| : |  |  |  |  |
| : |  |  |  |  |
| 30 | Adding a Client | When all data not entered, error message issued |  |  |  |
|  |  | Detects invalid phone number |  |  |  |
|  |  | Adds client to the beginning of the list |  |  |  |
|  |  | Add client to the end of the list |  |  |  |
|  |  | Add client to the middle of the list |  |  |  |
|  |  | Add client when list is already full |  |  |  |

You will fill in the table as each feature / function is tested. Please note that software can not be considered fully functional until all test cases pass when executed consecutively without any adjustment.

1. ***User documentation (pairs only)***

The purpose of user documentation is to help anyone unfamiliar with the program to use it. Good user documentation should include sample output as well as written instructions about running the program illustrated with screenshots.

1. ***Project review***

This project review should be a critical analysis of the resulting software. Effectiveness should be discussed in relation to the original goals and objectives that were defined in 1A by answering the following questions:

* Did it work?
* Did it address the goals and objectives?
* Did it work for some data sets, but not others?
* Does the program in its current form have any limitations?
* What additional features could the program have?
* Was the initial design appropriate?

1. ***Project Log (pairs only)***

The group includes a detailed log of all work done by each partner, including a description of task, time taken, and date.

**Evaluation Rubrics**

1. **System testing**

| **0** | The student has not reached a standard described by any of the descriptors given above. |
| --- | --- |
| **1** | The test plan represents a strategy that tests **a single** branch of the program with a single set of data. |
| **2** | The test plan represents a strategy that tests **some** branches of the program with **limited** range of **valid** data. |
| **3** | The test plan represents a strategy that tests **the most common** branches of the program with a **wide** range of **valid** data. |
| **4** | The test plan represents a strategy that tests **all** branches of the program with a wide range of **valid** and **invalid** data. |

1. **User documentation (pairs only)**

| **0** | The group has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The group includes **little** user documentation. |
| **2** | The group includes **some** user documentation. |
| **3** | The group includes user documentation that contains **clear** instructions about **running** the program. |
| **4** | The group includes user documentation that contains **clear**, **illustrated** instructions about and **running** the program. |

1. **Project review**

| **0** | The student has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | Project review demonstrates **minimal** reflection and assessment on the final solution. |
| **2** | Project review demonstrates **some** reflection and assessment on the final solution. |
| **3** | Project review demonstrates **mostly complete** reflection and assessment on the final solution. |
| **4** | Project review demonstrates **complete** reflection and assessment on the final solution |

1. **Project Log (pairs only)**

| **0** | The group has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The log contains **very few** details of the activities of the group. |
| **2** | The log outlines **some** of the activities of the group. |
| **3** | The log outlines **most** of the activities of the group in detail, as planned. |
| **4** | The log outlines **all** of the activities of the group in detail, as planned. |

## **Due Date for Stage 4: Jan 17, 2024**

**Evaluation Weighting**

| **Section** | **Weight** |
| --- | --- |
| Stage 1: Analysis | 10% |
| Stage 2: Design | 20% |
| Stage 3: Implementation | 50% |
| Stage 4: Testing, Documentation, Project Review | 20% |
| **Total** | 100% |

**Total weight of final mark: 20%**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1A **Analysing the problem & Criteria for success**

| **0** | The student has not reached a standard described by any of the descriptors given above. |
| --- | --- |
| **1** | The student states some objectives of the solution. |
| **2** | The student describes most of the objectives of the solution. |
| **3** | The student describes all of the objectives of the solution. |
| **4** | The student relates all of the objectives of the solution to the analysis of all aspects of the problem. |

1. **Data Structures**

| **0** | The student has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The student has outlined some of the data structures/types to be used in the solution. |
| **2** | The student has described some of the data structures/types to be used, and provided sample data. |
| **3** | The student has discussed most of the data structures/types to be used, and provided sample data. |
| **4** | The student has discussed and clearly illustrated all of the data structures/types to be used to solve the problem, and provided sample data for all of them. |

1. **Modular Organization (UML)**

| **0** | The student has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The student has **outlined some** of the modules to be used in the solution. |
| **2** | The student has **described most** of the modules to be used, **showing connections** between them. |
| **3** | The student has **described all** of the modules to be used, and has shown the connections to data structures and methods. |
| **4** | The modules discussed are sufficiently **logical, detailed** and **well documented** to be used to create the solution in Java. |

1. **Using good programming style**

| **0** | The program has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The program demonstrates **little** attention to good programming style. |
| **2** | The program demonstrates **some** attention to good programming style. |
| **3** | The program **mostly** demonstrates attention to good programming style. |
| **4** | All parts of the program demonstrate **considerable** attention to good programming style. |

1. **Usability**

| **0** | The program has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The program has very **little** user-friendly features. |
| **2** | The program has **some** user-friendly features. |
| **3** | The program is **reasonably** user-friendly |
| **4** | The program is **extremely** user-friendly and it **meets** the usability objectives in section 1B. |

1. **Handling errors**

| **0** | The program has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The program facilitates **little** error-handling |
| **2** | The program facilitates **some** error-handling |
| **3** | The program facilitates **many** error-handling |
| **4** | The program facilitates error-handling for **each** input and output method |

1. **Success of the program, part 1**

| **0** | The student has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The program **functions minimally**. It successfully achieved **few** of the objectives from section 1A. |
| **2** | The program **functions partially**. It successfully achieved **some** of the objectives from section 1A. |
| **3** | The program **functions mostly**. It successfully achieved **most** of the objectives from criterion 1A. |
| **4** | The program **functions fully**. It successfully achieved **all** of the objectives from criterion 1A. |

3D **Success of program, part 2**

| **0** | The student has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | The student correctly uses **few** required concepts in their program. |
| **2** | The student correctly uses **some** required concepts in their program. |
| **3** | The student correctly uses **most** required concepts in their program. |
| **4** | The student correctly uses **all** required concepts in their program. |

1. **System testing**

| **0** | The student has not reached a standard described by any of the descriptors given above. |
| --- | --- |
| **1** | The test plan represents a strategy that tests **a single** branch of the program with a single set of data. |
| **2** | The test plan represents a strategy that tests **some** branches of the program with **limited** range of **valid** data. |
| **3** | The test plan represents a strategy that tests **the most common** branches of the program with **wide** range of **valid** data. |
| **4** | The test plan represents a strategy that tests **all** branches of the program with wide range of valid and **invalid** data. |

1. **Project review**

| **0** | The student has not reached a standard described by any of the descriptors given below. |
| --- | --- |
| **1** | Project review demonstrates **minimal** reflection and assessment on the final solution. |
| **2** | Project review demonstrates **some** reflection and assessment on the final solution. |
| **3** | Project review demonstrates **mostly complete** reflection and assessment on the final solution. |
| **4** | Project review demonstrates **complete** reflection and assessment on the final solution |

**Evaluation Weighting**

| **Section** | **Mark** | **Weight** |
| --- | --- | --- |
| Stage 1: Analysis | /4 | 10% |
| Stage 2: Design | /8 | 20% |
| Stage 3: Implementation | /20 | 50% |
| Stage 4: Testing, Docs, Review | /8 | 20% |
| **Total** | /40 | 100% |

**Total weight of final mark: 20%**