CS-273-1 Final Project (Summer 2021)

Name: ______________________________________

Goal 1: Correctly implement moderately complex software simulation from specification.
Goal 2: Practice applying the concepts and techniques you have learned this semester.

Final Project Implementation

Project Implementation	**Points**	
FinalSpec.doc (see rubric)		30
FinalSummary.doc (see rubric)		20
Implementation:		
Use of class **inheritance** and **polymorphism**		10
Use of **set**, **map**, and/or **hash table** (1 or more)		10
Use of **queues** (a **priority queue** counts)		10
Final Code (see rubric)		20
TOTAL		100

Final Project Evaluation Rubric

	Emerging		**Developing**		**Mastering**	
	0	4	5	8	9	10
Final Specification Document	More than 2 missing or poorly specified components: requirements specification, use cases, UML diagram, and pseudo-code.		Adequate inheritance structure but does not use polymorphism. Alternatively, inadequate **use cases** or UML diagram in specification.		Excellent **inheritance** structure, using **polymorphism**, with detailed **use cases** and **UML** diagrams. **Design should match implementation.**	
Final Project Summary Report	Poor answers to the questions. No summary of problems or lessons learned in the implementation.		Good summary of problems encountered, final design, and lessons learned in the implementation.		Excellent project summary, with good discussion of results, including the required graph.	
Final code	Code is sloppy, unstructured, and/or uncommented No unit tests No useful commit comments in Github		Code is decent, but has some sloppiness or lack of structure Few unit tests Few useful commit comments in Github		Code is solid and well thought-out Many unit tests All useful commit comments in Github	

Project Overview

- **Pick one of the problems described in this document, and design/implement a solution for it.**

Solution Strategy

- **Decide on a real world problem to simulate (Traffic simulation, fire in a building).**
- **Understand the rubric, and know what your design and implementation will need to have.**
- Meet with your partner and work through the **use case** for this problem. Note that since this is a simulation, you will want to understand what your simulator needs to do at each **clock tick**. **Treat a clock tick like a user input and write down its use case.** I have always found this a helpful activity.
- Start the design specification by drawing a high-level UML diagram. You will not need to know all the properties or methods for your classes at the beginning. Just understand the high-level structure first, **including what will be your base and sub classes. Model your specification on design specifications given in the book (airport / phone directory)**
- You can start implementing a **prototype** to flush out the details of your design. Fill in the details of your UML diagram.
- When you are confident about your design, **throw your prototype away**, and **begin implementing** your solution.

What You Should Not Do

- **Do not just copy the airport simulator homework solution and modify it.**
- Do not forget to test everything you implement. Remember to unit-test as much as possible.

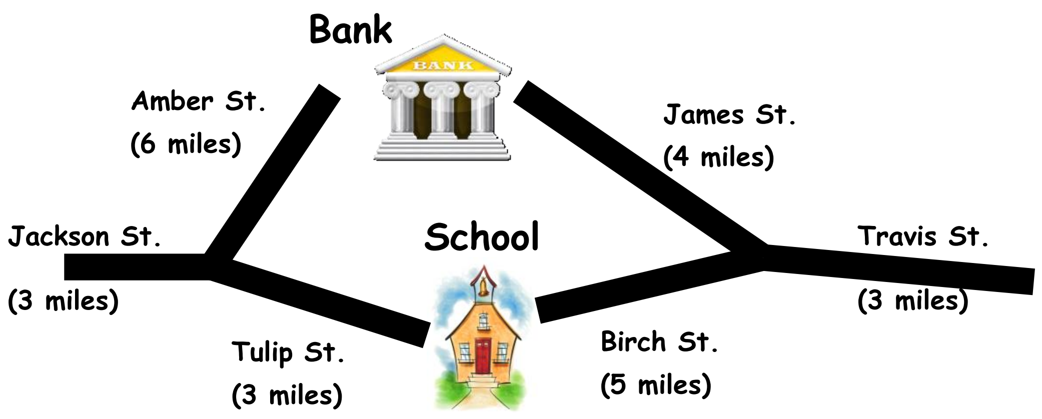
Project Deliverables

- **Two or more pages design/specifications document called "FinalSpec.doc"**: It needs to include a requirements specification, use cases, UML diagrams, and any pseudo-code for your simulation. The more detailed the design, the higher your score will be. **Keep a copy in GitHub.**
    - Your design will inevitably change when you begin your implementation. Make sure you update your design documents constantly to reflect the current state of your design.
- **Event-based simulation in folder PR_Final_Simulation**:
    - An implementation of the simulation of one of the problems described on the next page.
    - Simulation must include **inheritance and polymorphism**
    - Simulation must use **set**, **map**, and/or **hash table**. At least ONE of these data structures must be used.
    - Simulation must use one or more **queues (a priority queue counts as well)**.
- **One page project summary report "FinalSummary.doc"**: That includes answers to the questions given below in the project descriptions, and includes the required graph. Also indicate what was changed from your initial design in your final implementation. If implementation did not work, describe your design, implementation challenges, and lessons learned.

Problem 1: An Emergency Room Simulator

- You need to simulate a hospital emergency room located in the town of *273ville*, population 2000.
- The names of all the residents of *273ville* are stored in our class folder.
- Every person in 273ville is equally healthy, but they do occasionally need to go to the emergency room.
    - I.e. there is equal probability that anyone in town will be admitted into the emergency room as a patient.
- Your emergency room is small, but it has a big heart. It tries to run its operation as efficiently as possible.
- When a patient arrives at the emergency room, he/she is triaged – that is, the patient is assigned a priority number from 1 to 20, depending on the severity of the illness. **Higher priority values indicate more serious illnesses and are always treated ahead of lower priority illnesses.**
    - Illnesses with priority 1 to 10 occur approximately 70% of the time with equal probability
    - Illnesses with priority 11 to 15 occur approximately 20% of the time with equal probability.
    - Illnesses with priority 16 to 20 occur approximately 10% of the time with equal probability.
- The emergency room has 2 categories of caregivers: **Doctors** and **Nurses**.
    - **Nurses** can treat patients with **priority 1 to 10**, and **doctors** can treat patients with **priority 1 to 20**.
    - Empirically, we also know that **nurses** take **1 to 10 minutes** to treat a patient, and **doctors** take **1 to 20 minutes** to treat a patient, on average.
- The hospital **keeps a record of all patients** that were treated in the emergency room. Each record stores:
    - the number of visits to the emergency room, and
    - the severity of illness on each visit
- Your simulation needs to examine a week in the life of the emergency room on a minute-by-minute basis (i.e. it needs to simulate at least 7x24x60 minutes).
- Allow the user to input the following data values:
    - The **average hourly patient arrival rate** (patients/ hour) at the emergency room –assume that there will not be more than 60 patients per hour.
    - The **number of doctors** working in the emergency room.
    - The **number of nurses** working in the emergency room.
- You simulation will need to calculate the **average visit time** (arrival to discharge time) for emergency room patients.
- At the end of the simulation, you will need to **display a menu** with options to list the names of all residents that were treated, and retrieve the record of a resident by "name".
- For your **final report** and **presentation**:
    - **Compare and comment** on the average patient visit time (for some fixed patient arrival rate) when the emergency room has
        - 1 doctor and 1 nurse
        - 1 doctor and 2 nurse
        - 2 doctors and 1 nurse
    - **Display a plot** of the visit time for increasing patient arrival rates, for a combination of doctors and nurses of your choice

Problem 2: Downtown Traffic Simulator



- You need to simulate the **inflow** traffic pattern for the downtown core of 273ville, population 2000.
- The names of all the residents of *273ville* are stored in our class folder
- After studying the downtown visitation habits, we know that at any particular day, there is a 30% chance for a resident to visit the Bank, and a 70% chance for a resident to visit the School.
- When a resident visits the Bank, he/she will spend anyway from 10 to 20 minutes, and when a resident visits the School, he/she will spend anyway from 5 to 10 minutes.
- Residents visiting the downtown core can arrive and depart using only two streets: Jackson and Travis. *Residents are equally likely to arrive and depart from any of these streets.*
- The speed limit through downtown is strictly enforced, so citizen will typical drive within the range of 25 to 35 MPH (miles per hour). *You can randomly select a driving speed within this range for all visiting resident.*
    - *Note: The driving speed determines how long a resident takes to travel along each street. E.g. a car traveling at 25 MPH on Birch St. will take 12 minutes (5/25 * 60) to travel down that street.*
- Additionally, all streets have a **road capacity**. E.g. if a street has a road capacity of 2, it will only allow 2 cars to be traveling on the street at any time. If more than 2 cars arrive at the street, the additional cars will have to wait until at least one car completes travelling down the road before a new car can begin travelling down the street.
- Your simulation needs to examine a week in the life of downtown 273ville on a minute-by-minute basis (i.e. it needs to simulate at least 7x24x60 minutes).
- Allow the user to input the following data values:
    - The **average hourly visitor arrival rate** (visitors/ hour) –assume that there will not be more than 60 visitors per hour.
    - The **road capacity** for all the streets (if you like, you can have your simulator read this from a file).
- Your simulation needs to keep a record of everyone who visited downtown. Each record stores:
    - the number of visits downtown
    - the destination on each visit
- Furthermore, you can assume the following decisions when a resident reaches a 3-way intersection:

Intersection	Destination	Street to take
Jackson-Amber-Tulip	School	Tulip St.
Jackson-Amber-Tulip	Bank	Amber St.
Travis-Birch-James	School	Birch St.
Travis-Birch-James	Bank	James St.

- You simulation will need to calculate the **average travel time** for the downtown visitors (i.e. time from arriving until departing the downtown core after visiting the School or Bank).
- At the end of the simulation, you will need to **display a menu** with options to list the names of all residents that visited downtown, and retrieve the record of a resident by "name".
- For your **final report** and **presentation**:
    - **Compare and comment** on the average travel time (for some fixed visitor arrival rate) for a combination of road capacities (your choice).
    - **Display a plot** of the travel time for increasing visitor arrival rates, for a combination of road capacities

Problem 3: An event-based simulation of your choice

- **Your simulation must be of the same complexity as the other 2 problems.**
    - Please read the other 2 problems first and understand what is expected if you pick this option.
- Your simulation must include **inheritance and polymorphism**
- Your simulation must use **set**, **map**, and/or **hash table**. At least ONE of these data structures must be used.
- Your simulation must use one or more **queues (a priority queue counts as well)**.
- Note that if you choose this option, there is one **additional deliverable** to the default set of deliverables!
- **Deliverables**:

    1) Requirements specification, and

    2) a detailed description of a property you are simulating you can compare and comment on in your final presentation, as well as the type of graph you will plot based on the results of your simulation.

    - **Design/specifications document called "FinalSpec.doc" as described above**
    - **Presentation, final event-based simulation code, and final report as described above**
    - I will **need** to approve this before you can continue. The expectation is that the property you will discuss, and the graph you will plot, should be answerable only with a simulation.