



---

## ASSIGNMENT 2

---

**ECE 7400 - (Advanced) Concurrent Programming**



**JUNE 20, 2022**

**To: Dr. Reza Shahidi**

**By:**

**Kamal Ali**

**Ayman Abu Awad**

**ID: 201827946**

**ID: 201808672**

## Table of Contents

<b>1.0</b>	<b><i>Project Description</i></b> .....	<b>4</b>
1.1	Requirements to execute the project .....	4
<b>2.0</b>	<b><i>Terminal results</i></b> .....	<b>5</b>
<b>3.0</b>	<b><i>Overall Conclusions</i></b> .....	<b>9</b>

<b>Figure 1: Showing Elevator Threads .....</b>	<b>5</b>
<b>Figure 2: Riders Queue .....</b>	<b>5</b>
<b>Figure 3: Rider Generations .....</b>	<b>6</b>
<b>Figure 4: Assigning Elevators.....</b>	<b>6</b>
<b>Figure 5: Moving Elevators .....</b>	<b>6</b>
<b>Figure 6: Elevator Moving.....</b>	<b>7</b>
<b>Figure 7: Elevator Opening Doors .....</b>	<b>7</b>
<b>Figure 8: Multiple Elevators Assigned .....</b>	<b>7</b>
<b>Figure 9: Riders Rejection .....</b>	<b>8</b>
<b>Figure 10: Elevators Results after 200 cycles.....</b>	<b>8</b>
<b>Figure 11: Elevators Results after 7200 cycles.....</b>	<b>8</b>

## 1.0 Project Description

1. The elevator first travels towards the closest floor with a button pressed (if there are two different floors which are closest to the current floor, then one is chosen at random). Also, only one elevator will go after a given request on a floor. If the request is being handled, another elevator will not go after the request.
2. Then the elevator travels in the direction of travel towards the destination floor chosen by the rider. If there are intermediate floors where riders have pressed buttons to go in the same direction, then each of those riders should also be picked up if the maximum capacity of the elevator has not yet been exceeded. Otherwise, the potential rider(s) will be rejected and take the stairs instead.
3. The elevator will only switch directions once all requests in the current direction (up or down) have been handled, and there is another direction in the opposite direction pending. If there is no other request pending in the opposite direction, once the elevator has handled all requests in the current direction, then the elevator waits at the current floor until another request is made.

Design A: Because of COVID restrictions, the Chief Health Officer at the university proposes that there should be more elevators built, but each elevator should only have a capacity of 1 person at a time. In this scenario, there will be a total of 4 elevators built. Design B: The assistant to the Chief Health Officer thinks that if the elevators are built to be larger, then there should not be a major issue with having 2 people in an elevator at the same time. But because the elevators are supposed to be larger, only two elevators can fit in the allotted space for the elevator shafts instead of 4.

### 1.1 Requirements to execute the project

1. First, need to install all the dependencies. This project uses a Java 16 SDK and can be done by following the following instructions on this website: [click here](#)
2. To run the Elevator application, user can set up their own command using their preferred idea or execute the following command: `"javac CSFElevatorSimulator.java"`

## 2.0 Terminal results

Once everything is set up, the elevator project will be able to display results in a terminal as can be shown in the following figures:

```
/Users/ayman/Library/Java/JavaVirtualMachines/openjdk-16.0.2/Contents/Home/bin/java
Starting: Thread-2
Starting: Thread-0
Starting: Thread-1
Starting: Thread-3
```

*Figure 1: Showing Elevator Threads*

The project will first initiate all threads necessary to run the elevators as can be shown:

```
/Users/ayman/Library/Java/JavaVirtualMachines/openjdk-16.0.2/Contents/Home/bin/java
Starting: Thread-3
Starting: Thread-0
Starting: Thread-2
Starting: Thread-1
Floor: 1. Next rider: 115
Floor: 2. Next rider: 34
Floor: 3. Next rider: 99
Floor: 4. Next rider: 76
Floor: 5. Next rider: 24
current tick: 1
current Thread:main
```

*Figure 2: Riders Queue*

Once all threads are generated, the riders will be randomly generated for each associated five floors as can be seen above, and this will all occur in the first cycle in the main thread. The main thread will only be involved to show the current status of all elevators. Each thread will be associated with an elevator and will carry the tasks of moving riders from their starting floor to the destination floor.

```
Floor: 1. Next rider: 115
Floor: 2. Next rider: 34
Floor: 3. Next rider: 99
Floor: 4. Next rider: 76
Rider generated
Starting floor: 5
Dest floor: 1
Floor: 5. Next rider: 77
current tick: 24
```

*Figure 3: Rider Generations*

Once a rider is generated, the starting floor and destination floor will be determined for that rider, as can be observed from the figure above. Along with this, the next rider gets generated for that same floor.

```
The elevator will pick up from floor: 5 And drop off at floor: 1
current thread: Thread-3
elevator current floor: 1 Destination floor: 5 Moving UP
Elevator 01 moving UP Current time: 24 current floor: 1 Destination floor: 5
Floor: 1. Next rider: 115
Floor: 2. Next rider: 34
Floor: 3. Next rider: 99
Floor: 4. Next rider: 76
Floor: 5. Next rider: 77
current tick: 25
```

*Figure 4: Assigning Elevators*

From here we can observe that an elevator gets assigned from the available threads to pick up the generated rider from the starting floor. The elevator that will proceed to move up each with five cycles per floor intervals. Each cycle the elevator will check if there are new riders being generated.

```
Elevator 01 moving UP Current time: 29 current floor: 2 Destination floor: 5
Floor: 1. Next rider: 115
Floor: 2. Next rider: 34
Floor: 3. Next rider: 99
Floor: 4. Next rider: 76
Floor: 5. Next rider: 77
```

*Figure 5: Moving Elevators*

This picture shows the five-cycle duration it takes for an elevator to move through each floor. As has been shown, the elevator started moving at cycle 24 and reached the next floor at cycle 29. It will keep incrementing the floor every five cycles.

```

The elevator will pick up from floor: 1 And drop off at floor: 4
current thread: Thread-0
ElevQ: [1, 4]
elevator_queue after pickup: [4]. Size: 1
Elevator 0: Doors open
Floor: 1. Next rider: 73
Floor: 2. Next rider: 65
Floor: 3. Next rider: 66
Floor: 4. Next rider: 119
Floor: 5. Next rider: 71
current tick39
current Thread:main

```

*Figure 6: Elevator Moving*

Afterwards, the elevator will pick up the rider from the starting floor, as shown by the Elevator 0: Doors open statement. The elevator will wait for 15 cycles before it begins to move the destination.

```

Elevator 0: 15 seconds has passed by for rider to get in or out
current thread: Thread-0
elevator current floor: 1 Destination floor: 4 Moving UP
Elevator 0 moving. Current time: 53 current floor: 1 Destination floor: 4
Floor: 1. Next rider: 73
Floor: 2. Next rider: 65
Floor: 3. Next rider: 66
Floor: 4. Next rider: 119
Floor: 5. Next rider: 71
current tick54
current Thread:main

```

*Figure 7: Elevator Opening Doors*

As shown above, after 15 cycles, the elevator will begin to move.

```

Elevator 1 moving. Current time: 68 current floor: 1 Destination floor: 2
Elevator 1 moving. Current time: 68 current floor: 1 Destination floor: 2
Elevator 1 moving. Current time: 68 current floor: 1 Destination floor: 2
Elevator 0 moving. Current time: 68 current floor: 3 Destination floor: 4
Elevator 1 moving. Current time: 68 current floor: 1 Destination floor: 2
Elevator 1 moving. Current time: 68 current floor: 1 Destination floor: 2
Elevator 0 Dropping off riders
Elevator 1 moving. Current time: 68 current floor: 1 Destination floor: 2
Elevator 1 moving. Current time: 68 current floor: 1 Destination floor: 2
Floor: 1. Next rider: 73
Floor: 2. Next rider: 185
Floor: 3. Next rider: 186
Floor: 4. Next rider: 119
Floor: 5. Next rider: 71
current tick69
current Thread:main

```

*Figure 8: Multiple Elevators Assigned*

Since the elevator will move up 3 levels, it will arrive after 15 cycles. “Elevator 0 Dropping off riders” can be seen above. Also note, multiple elevators will be assigned to pick up other riders if elevator is busy, this can be shown as multiple threads in the figure above.

```
Elevator 1 moving. Current time: 70 current floor: 1 Destination floor: 2
Elevator 1 moving. Current time: 70 current floor: 2 Destination floor: 2
ElevQ: [3, 4, 5]
elevator_queue after pickup: [4, 5]. Size: 2
Rejected
Elevator 1: Doors open
```

*Figure 9: Riders Rejection*

Elevators will reject riders if they reach maximum capacity if a rider attempts to enter an elevator, this can be shown as multiple threads in the figure above.

```
current tick200
current Thread:main

Total Requests: 18
Total Rejects: 1
Total Served: 17
Average Time Per Request: 20
```

*Figure 10: Elevators Results after 200 cycles*

```
current tick: 7200
current Thread:main

Total Requests: 502
Total Rejects: 1
Total Served: 501
Average Time Per Request: 11
```

*Figure 11: Elevators Results after 7200 cycles*

Finally, at the end of the simulation time, the stats will be printed. Note that the simulation time above is 200 in the first case, and 7200 in the second case.



### 3.0 Overall Conclusions

The riders are able to be picked up using individual elevators that are to be assigned at the time of requests and dropped off at their associated floors. Each additional rider that requests to be picked up heading in the same direction will need to first check if the capacity of the elevator has reached its limits, if the capacity is at its limit then the rider will be rejected. As a result, elevators are able to keep a capacity of 1 person at a time in the scenario of design A implementation where there are a total of 4 elevators built. Along with that, the elevators are also able to be configured to a larger capacity with having 2 people in an elevator at the same time for the scenario implementation of design B implementation.