CSCE 3613 Operating Systems
Museum Programming Assignment, ver. 1.2

100 points

Instructions

- Name your Java program “Museum.java”.
- Make sure that your code can be run from the **command line** as “java Museum 200 50”, which means that it will run of 200 units of time and 50 visitors arrive to the museum (1 unit of time = 100ms).
- You may debug the program on other platforms, but you must compile and run the final program on the machine turing.uark.edu.
- Put your name and UA ID in a comment at the top of your code.
- Create one ZIP file of your assignment consisting of the following and upload it to Blackboard:
    - All source code
    - A readme.txt file that describes your program
    - An output.txt file that is the output of your program from “java Museum 200 50” and should look like the example below

Implement a simulated Museum using the programming language Java. You must use Java with Threads. In addition, you cannot use the synchronized command in this assignment but instead use semaphores and mutexes.

System description
There is a museum that has a waiting room, dinosaur room, zoology room and a gift shop. Visitors come to the museum and start outside in a line and then enter the waiting room that has room for a maximum of **40** people. A security guard is posted and only lets 40 visitors in the waiting room because of fire marshal rules. A security guard is posted at the entrance of each room and the gift shop to enforce the number of visitors in each room. Once in the waiting room, a visitor attempts to go to the dinosaur room that holds **20** people and spends a random amount of time that averages *10* units of time once they enter the dinosaur room. Then, the visitor goes from the dinosaur room to the zoology room that holds **25** people and spends a random amount of time that averages *15* units of time. After the zoology room, the visitor goes to the gift shop that can hold **30** people and spends an average of *30* units of time. If the next room is full, the visitor must wait. All random time has a uniform distribution. Finally, the visitor leaves the gift shop and museum.

Parameters

The amount of time that we permit visitors in the system and number of visitors are variables in this programming assignment.

Example
Your program should be compiled and run by the following commands:
javac Museum.java
java Museum 200 50

java Museum 200 50 means that the program will permit 50 visitors to visit the Museum for 200 units of time. At the end of 200 units of time, no more visitors enter the system, but the remaining visitors still can make a tour then leave until the system is empty.

Structure of Program
This program should work as follows: The user will enter on the command line the *sleep time* and the *number of visitors*. Each visitor should be a single thread and there should be a thread for the system that will run for *sleep time * unit of time*. One example of the Java application from the command line could be "java Museum 200 50". **The code must use threads, mutexes, and semaphores.** The `Museum` application will create a thread for the system consisting of the museum's rooms and 50 different visitor threads. Each Visitor thread will also choose a random amount of time to sleep using `Random()` (make sure to look up what kind of random number `Random()` provides) and `Thread.sleep()` that is *less than the given sleep time*. **Set the sleep time of each Visitor thread to be a random amount of time between 0 to *time_unit*corresponding sleep time* (depends on in which room the visitors are).** Each visitor thread will print to standard I/O when it enters the system, enters or leaves the waiting area, enters and leaves each room to the screen, counts number of visitors in room when a visitor enters the room (see sample output below). Note that **you must use mutex when counting to avoid a race condition among threads**!

All code is recommended to be in one file called *Museum.java.* Below is the skeleton of the program that you can follow:

```
Class Visitor implements Runnable {
        // Class attributes here, including:
        //      id
        //      Semaphores, mutexes for each room
        //      variables for counting visitors

        // Class constructor
        Visitor(….) {
                // Insert here
        }

        // Get waiting room
        Private void getWaitingRoom() {
                // Insert here
        }
```

```
          // Get dinosaur room
          Private void getDinosaurRoom() {
                    // Insert Here
          }

          // Zoology Room
          Private void getZoologyRoom() {
                    // Insert Here
          }

          // Gift Room
          Private void getGiftRoom() {
                    // Insert Here
          }

          // Exit Museum
          Private void exitMuseum() {
                    // Insert Here
          }

          // Implement run() method
          Public void run() {
                    // Time to call all implemented sub-methods
          }
}


// Museum Class
Public class Museum {
          // Arguments parsers
          // Shared semaphores, mutexes, variables

          // Create Visitor objects using for-loop
          // Call start() method
          For (…) {
                    // Insert here
          }

          // Sleep the program thread using Thread.sleep()
          // Insert here
}
```

Usage and Typical Output

```
java Museum <sleep time> <number of visitors>
```

Output should be printed with delimiters corresponding to each action (see below). The output below is just the **sample** and therefore is not numerically correct due to space limit, but your output **must** follow this style.

```
$ java Museum 200 50
Sleep time = 200  Number of users = 50

Visitor 0 enters the system
        Visitor 0 enters the waiting area and is waiting to enter the museum. There are 1 waiting
                Visitor 0 enters dinosaur room. There are 1 watching dinosaurs!
Visitor 2 enters the system
        Visitor 2 enters the waiting area and is waiting to enter the museum. There are 1 waiting
                Visitor 2 enters dinosaur room. There are 2 watching dinosaurs!
Visitor 1 enters the system
        Visitor 1 enters the waiting area and is waiting to enter the museum. There are 1 waiting
Visitor 29 enters the system
                Visitor 1 enters dinosaur room. There are 3 watching dinosaurs!
Visitor 27 enters the system
        Visitor 27 enters the waiting area and is waiting to enter the museum. There are 1 waiting
                Visitor 27 enters dinosaur room. There are 4 watching dinosaurs!
        Visitor 24 enters the waiting area and is waiting to enter the museum. There are 1 waiting
                Visitor 24 enters dinosaur room. There are 5 watching dinosaurs!
        Visitor 28 enters the waiting area and is waiting to enter the museum. There are 1 waiting
                Visitor 28 enters dinosaur room. There are 6 watching dinosaurs!
        Visitor 34 enters the waiting area and is waiting to enter the museum. There are 1 waiting
                        Visitor 28 enters Zoology room. There are 1 enjoying animals!
        Visitor 26 enters the waiting area and is waiting to enter the museum. There are 2 waiting
        Visitor 23 enters the waiting area and is waiting to enter the museum. There are 4 waiting
                        Visitor 30 enters Zoology room. There are 21 enjoying animals!
                Visitor 29 enters dinosaur room. There are 20 watching dinosaurs!
                                Visitor 25 enters Gift room. There are 20 looking to buy!
                        Visitor 9 enters Zoology room. There are 21 enjoying animals!
                        Visitor 8 enters Zoology room. There are 22 enjoying animals!
                Visitor 32 enters dinosaur room. There are 19 watching dinosaurs!
                                Visitor 10 enters Gift room. There are 21 looking to buy!
                Visitor 31 enters dinosaur room. There are 20 watching dinosaurs!
                                Visitor 14 enters Gift room. There are 20 looking to buy!
                        Visitor 0 enters Zoology room. There are 21 enjoying animals!
                        Visitor 27 enters Zoology room. There are 22 enjoying animals!
                        Visitor 20 enters Zoology room. There are 23 enjoying animals!
                        Visitor 18 enters Zoology room. There are 24 enjoying animals!
                                        Visitor 22 exits the system
                                Visitor 44 enters Gift room. There are 23 looking to buy!
                        Visitor 33 enters Zoology room. There are 24 enjoying animals!
                                Visitor 23 enters Gift room. There are 23 looking to buy!
                        Visitor 38 enters Zoology room. There are 24 enjoying animals!
                                Visitor 16 enters Gift room. There are 23 looking to buy!
                                        Visitor 24 exits the system
                                        Visitor 19 exits the system
                                        Visitor 44 exits the system
                                        Visitor 29 exits the system
```