|  |  |  |
| --- | --- | --- |
| American University of SharjahSchool of Engineering Department of Computer Engineering  P. O. Box 26666  Sharjah, UAE  **Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **ID : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |  | **Instructor:** Omar Arif **Office**: ESB-2178  **Phone**: 971-6-515 4821  **e-mail**: oarif@aus.edu  **Semester**: |

COE312 –

**The quiz open book and notes and you can use the Internet. You are not allowed to consult other individuals or your classmates.**

PLEASE UPLOAD YOUR ANSWER USING THIS **WORD** FILE. DO NOT UPLOAD ANY ZIPPED OR PDF FILE. ONE WORD FILE ONLY.

The question(s) have been written in a manner such that it is not possible for two students to have the same solution. Therefore, please refrain from the temptation of copying code and changing order of and names of variables, etc. AUS code of conduct will be strictly enforced, and no violation will be tolerated as per AUS policy. Please note that the AUS code does not discriminate between who copied from whom so it is not advisable to share your solution.

**Q1. (10 points)** Each door in the ESB building is equipped with people counting sensor. The sensor counts the number of people entering and leaving the building every **two seconds**. Assume between 1 to 5 people enter and leave the building in two seconds. You can use the following to simulate this:

Random rand = **new** Random();

**int** peopleEntering = rand.nextInt(5);

**int** peopleLeaving = rand.nextInt(5);

The building has **5** **front doors** and **5** **back doors**. A central **observer** is responsible for monitoring the number of people inside. **Every 5 second**, the observer reports the total number of people in the building, as well as the number of people who entered through the front and back doors.

Implement the above scenario using Java classes and **message-based** design pattern. Synchronize the shared resources where required.

You must run and provide output on the following program (without any changes to the program). You are not allowed to make any changes to this program

**import** java.util.ArrayList;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Counter c = Counter.*getInstance*(); // Counter is a singleton class

ArrayList<ConcreteSubject> subjects = **new** ArrayList<ConcreteSubject>();

**// Front doors**

**for** (**int** i=0; i<5; i++) {

subjects.add( **new** FrontDoor() );

}

**// back doors**

**for** (**int** i=0; i<5; i++) {

subjects.add( **new** BackDoor() );

}

CentralObserver observer = **new** CentralObserver(subjects, c);

}

}

Expected Output:

Total people in Building: 0

Front Door: 0

Back Door: 0

Total people in Building: 4

Front Door: 0

Back Door: 4

Total people in Building: 10

Front Door: 0

Back Door: 10

Total people in Building: 14

Front Door: 8

Back Door: 6

Total people in Building: 7

Front Door: 0

Back Door: 7

Total people in Building: 20

Front Door: 6

Back Door: 14

Total people in Building: 29

Front Door: 12

Back Door: 17

Please provide 1) formatted code and 2) screenshots of your running program. Not providing a screenshot (with or without errors) will limit your score to below 3/10.

|  |
| --- |
| **Solution: (Paste formatted code in this box).** |

|  |
| --- |
| **Solution: (Paste the screenshot in this box).** |

**Grading Rubric:**

|  |  |  |  |
| --- | --- | --- | --- |
| **0-3** | **4-6** | **7-8** | **9-10** |
| * The program does not compile or run but there is some notion of a solution OR * No screenshot is provided. | * Some of the program works but not fully. * Bad design practices have been used. | * The program partially matches the output provided. * Clean design practices have been followed. | * The program exactly matches the output. * Clean design practices have been followed. |