

# CS F213: OOP Lab 9 - Exceptions

Ishan Garg, Gunjan Arora

5th November 2019

## General instructions:

1. Read the question carefully
2. Indent your code so as to make your code more readable and amenable to debugging

Let us continue our exploration of BITS-Pilani, Chandigarh campus. This lab will do that using the concept of exceptions.

In Lab 3 we created a few classes that simulated the basic structure of the BITS-Pilani, Chandigarh campus. While the code works fine for the ideal situation but when the situation is non-ideal, we need to take care of some troublesome situations using exceptions.

For a basic understanding of campus structure use the description of lab 3 or the JavaDoc. In this document, description of only new methods and exceptions is given.

## New classes to be created

### 1. InvalidDimensionException

It should be thrown when the dimensions of any room are not valid. For example, it is not possible for a room to have length, breadth or height less than or equal to zero.

The class will override the getMessage() method of the Exception class and will provide the message "Invalid dimensions given."

### 2. InvalidStateException

This class handles the situation when an invalid state of the appliance(states other than boolean states) is passed.

For example:- flipApplianceState() method should not accept any other state other than true or false.

The class will override the getMessage() method of the Exception class and will provide the message "Invalid state given. Possible states are true and false only."

### 3. MaximumCapacityException

This class handles the situation when more objects of one type cannot be added to the objects of another type. For example, there is a limit on the number of classrooms in the academic block.

The class will override the getMessage() method of the Exception class or you can directly call the super class constructor for setting the message. For details of the message to be provided refer to JavaDoc

#### **4. invalidDeviceException**

This class handles the exception when objects are tried to be used in invalid places. For example, no other device other than AC and Lights is excepted in the classroom.

The class will override the getMessage() method of the Exception class and will provide the message "(A given room) cannot have appliance (appliance which is passed as argument)". For further details of the message to be passed see JavaDoc.

#### **5. InvalidBrokenAppliances**

It is assumed that a generic room will have tables, lights, and fans. This class handles the exception when an invalid number of broken tables, lights or fans are given for any type of room. For example, in any room, their number cannot be less than zero.

The class will override the getMessage() method of the Exception class and will provide the message "Number of broken appliances invalid."

#### **6. DivideZeroException**

This class handles the exception when division by zero takes place. This exception may arise, for example, while calculating the average.

The class will override the getMessage() method of the Exception class and will provide the message "Cannot calculate average cost because number of <Object> are insufficient". For further details of the message to be passed see JavaDoc.

Test Cases:

- |                     |         |
|---------------------|---------|
| 1. AcadBlock class  | 2 marks |
| 2. Campus class     | 2 marks |
| 3. Hostel class     | 2 marks |
| 4. Room class       | 2 marks |
| 5. HostelRoom class | 1 marks |
| 6. Classroom class  | 1 marks |