Object Oriented Programming

Lab 1 – 09/02/2021

Closely monitoring the trend of decline in cases of COVID-19 in India and keeping in mind the inability of certain students to attend the classes, and the online labs, the administration has decided to open up the campus to certain students across different batches.

The opening of campus is planned out in a phased out manner and is based on the student's preference of Laboratory and the maximum number of students they can manage with social distancing norms. Although the order in which students requests are acknowledged is unclear, the maximum number of students per batch (2017,2018,2019,2020) is fixed.

Additionally, acknowledging the requests from many students, the admin decides to increase the number of seats in certain labs but they require your help with an Object Oriented Java Program which would facilitate an orderly return.

Instructions

1. Ensure that name of file is the same as the class name with appropriate extension
2. Ensure that the name of methods is same as those given in Javadoc. Failure to follow this instruction may cause deduction in credit even if the implementation is correct.
3. You are provided with a template file with helper code already written. **Please do not edit the helper code.**
4. Please do not copy directly the names of methods from Javadoc, instead write them out yourselves.
5. You are required to upload all the classes at once, individual submissions would not be accounted.
6. A brief description of the classes and methods to be used is given further in the problem statement but you are advised to refer to the Javadoc for the complete description
7. The advisable order of writing the code:
    a. Student
    b. Lab
    c. Admin

Prepared by – Kshitij Chhabra, Sagar Badiyani, Vishal Bharambhe

List of Classes and Methods

Class- Student

Fields

1. **ID** – Stores the BITS ID number of person
   Example- 2017A8PS0691G
   Here, 2017 is the year of enrollment and A8 is the branch code

2. **preferredLab** - Contains the code of the desired lab for student. BITS has decided to open five labs and the student has to chose the lab he intends to be considered for.

Methods

1. **getID()-** returns the ID number of the person
2. **getYear()-** returns the year of enrollment in integer format
3. **getBranch()-** Returns the branch the student is enrolled in
4. **getPreferredLab()-** Returns the code for the lab selected by the student

Class- Lab

Fields

1. **labCode-** Stores the code for this lab
2. **currentNumberOfStudents-** Stores the number of students who will be currently enrolled in this lab. Initialised to zero inside the constructor.
3. **maxCapacity-**Stores the maximum number of students who are eligible to enroll for this lab

Methods

1. **getLabCode()-**Returns the lab code
2. **getCurrentNumberOfStudents()-** Stores the number of students who are currently enrolled in the lab
3. **getMaxCapacity()-** Returns the max number of students which are eligible to enroll inside the lab
4. **increaseSeats(int numSeats)-** Increases the maximum capacity of lab by numSeats.
5. **updateCurrentNumberOfStudents()-** Updates the currently enrolled count and increases it by one whenever a student is added in the lab

Point to note- Majority functions described above (except increaseSeats and updateCurrentNumberOfStudents()) are called getter functions, their primary use is to return private elements of a class outside the class.

Class – Admin

Fields

1. **student-** An array object of student class storing the list of students coming to campus. Initialise the student to store a maximum of 100 students.
2. **labs-** An array object of lab class
3. **maxPerYearCapacity-** An integer containing maximum number of students per batch that can come to campus
4. **currentNumberOfStudents-** A variable containing the total number of students per lab combined across all batches who are coming to campus
5. **currentPerYearStudent-** An array of length 4 containing the current number of students per batch coming back to campus. The array index is matched to year of enrollment as:

Year of Enrollment	Array Index
2017	0
2018	1
2019	2
2020	3

6. **numberOfLabs-** An integer storing the number of labs that BITS has planned on reopening . Initialised to 5.
7. **numberOfStudentsApplied:** Stores the total number of students whose requests are considered throughout the program.
8. **numberOfPeopleAppliedForALab:** An array of size 5 containing the number of students who applied for the respective lab.

Methods

1. **getCurrentNumberOfStudents():** Returns information about the total number of students who are coming to the campus
2. **increaseLabCapacity(int labCode, int num):** Increases the lab capacity of the lab with code of labCode by a number num
3. **addStudent:** Adds the student to the list 'student' if the total number of students from the student's batch is less than the number of students the admin has planned. Additionally, check if the number of seats from the students preferred lab aren't full yet. Update the remaining variables accordingly. Returns true if the student is added, else return false.
4. **successRatio():** Returns the ratio of number of students whose requests to come back got accepted to the number of people who applied to return to campus.
5. **labSuccess():** Returns an array of ratios of the number of people who got the preferred lab to the total number of students who applied to that particular lab.

Please note: In case of discrepancies in variable names or method names between Javadoc and problem statement, please refer to Javadoc.

Marking Scheme

Test Function	Marks
getPreferredLab()	0.5
updateCurrentCapacity()	0.5
increseSeats()	1
increaseLabCapacity()	1
addStudent()	0.5
addStudent()	0.5
addStudent()	0.5
successRatio()&labSuccess()	1
students []	1.5
Total	**7**

Best of Luck !

Prepared by – Kshitij Chhabra, Sagar Badiyani, Vishal Bharambhe