

Your aim is to write a console program that can simulate a batch allocation system

Users have the following capabilities

- Students should be able to register to the system.
- Admin should be able to register to the system.
- Admin should be able to create the batches.
- Students should be able to enroll for a particular stream.
- A student can enroll only once in any stream(IIT/NEET).
- Admin should be able to allocate a batch to students on the basis of multiple criteria.
 - **Gender based**
 - Female students should be allocated in the below sequence .
 - Morning -> Noon -> Evening i.e. if morning is available it should be allocated first
 - **Highest Capacity**
 - Biggest initial capacity batches should be allocated first.
- Batch allocation will happen to all the batches present in the system(created by any admin) for a given stream.

Input and output

The inputs for various actions supported should be taken by invoking some method. The method signature should contain sufficient information to support all the requirements.

You may change the input output format without changing the functionality to suit your needs.

x

Sample Examples

- register("Akhilesh", Male, Student) - Student1
- register("Komal", Female, Student) - Student2
- register("Rajnish", Male, Student) - Student3
- register("Mayuri", Female, Student) - Student4
- enroll(Student1, Stream : IIT)
- enroll(Student2, Stream: IIT)
- enroll(Student3, Stream: NEET)
- enroll(Student4, Stream: IIT)
- register("Kamesh", Male , Admin) - Admin1
- register("M", Male, Admin) - Admin2
- createBatch(Admin1, Capacity=3, Stream : IIT, Timing : Morning) - B1
 - **Capacity** of the batch is the max number of students which can be allocated to this batch
- createBatch(Admin1, Capacity=2, Stream: NEET, Timing: Evening) - B2
- createBatch(Admin1, Capacity=3, Stream : IIT, Timing : Evening) - B3

- allocateBatch(Admin2, Student2, **"Gender Based"**)
 - Note : Student2 is a female
 - B1
- allocateBatch(Admin2, Student4, **"Higher capacity"**)
 - B1
- allocateBatch(Admin1, Student1, **"Higher capacity"**)
 - B1
- allocateBatch(Admin1, Student3, **"Higher capacity"**)
 - Note: Capacity of B1 is full.
 - B2

Guidelines:

- Write a driver class for demo purposes.
- Output can be written to STDOUT.
- Store all the data in-memory.
- You are free to use the language of your choice.
- Do not create any UI for the application.
- Save your code/project by your name and email it. Your program will be executed on another machine. So, explicitly specify dependencies, if any, in your email.

Expectations:

- Code should be demo-able (very important). Code should be functionally correct and complete.
 - At the end of this interview round, an interviewer will provide multiple inputs to your program for which it is expected to work
- Code should handle edge cases properly and fail gracefully. Add suitable exception handling, wherever applicable.
- Code should have good object-oriented design.
- Code should be readable, modular, testable and extensible. Use intuitive names for your variables, methods and classes.
 - It should be easy to add/remove functionality without rewriting a lot of code.
 - Do not write monolithic code.
- Don't use any databases.

Send your code to below email

- mani.diatm@gmail.com
- **Subject line : Allen - <NAME>**
- **In Body, steps to execute the code**