To Build the Solution Using Object-Oriented Programming in Java

Prerequisites: OOPS. Java

Description: This assignment is designed to help understand working with OOPS in Java and how the polymorphic behavior can be implemented.

Problem Statement: Yamin is a Gym Instructor. He takes multiple sessions of Zumba during the day, for which he has divided his participants into batches running in the morning and evening hours. To manage his participants, he needs a software solution to record the details of his participants and batches. Luckily his friend Mike runs a software company. Therefore, Yamin has requested the company to develop a software solution to manage his participants and batches.

The company aims to develop a core java based solution using OOPS.

Tasks:

Perform the below activities as a solution:

1. Create a Java Project in Eclipse
2. Create a Participant Class
3. Create a Batch Class

   Frank Note: You may also create any other classes you find you may need
4. Create 1 to many relationship mapping for the Batch and Participant.

   Frank Note: One batch may contain many Participant registrations for either morning or evening sessions.
5. Take the inputs from the User using Scanner class to create Participant and Batch Objects
6. Implement Method callbacks using Run time Polymorphism to implement. *(Frank Note: You will need Polymorphism).* When a batch starts, the Participant should receive a message.

   Frank Note:

   a. *When a Batch is processed, each participant should receive a notification message as to their status for registration of the session (accepted / rejected / pending).*
   b. *Processing of a Batch will register the Participant for the session they requested (or not).*
   c. *A reminder message containing pertinent information should be sent to a Participant for a session they are registered for when requested by the system.*
   d. *Don't worry about how big a batch will be nor the timing of when a batch will be processed or when messages request will occur. The system will control this. We don't know how to do that.*
7. Run the Code in the main class.

   Frank Note: Create an application program to test your code:

   a. *Instantiate a few Sessions for mornings and evenings*
   b. *Instantiate a few Participants*
   c. *Associate the Participants you instantiated with one or more Sessions you instantiated*
   d. *Create a few Batches containing Participant registrations*
   e. *Code a way to display all registrations in a batch.*
   f. *Code a way to test sending messages to Participants for a session they are registered for.*
   g. *Include any other testing of your code you think is necessary.*

Frank Note: This is not a complete list of all you may do. It is meant to get you started. Feel free to ignore or modify any and all of the "Frank Note" comments.

Don't forget to write User stories from the perspective of a Participant and owner (or others that might interact with the app). I.e. What would a Participant like to do and what would an owner like to do.