Assignment 2A

Background:

You are creating application that is going to track the swim clubs, their swimmers, swim meets and results.

References: Please refer to "General Assignment Requirements" document posted on eCentennial.

Part 2A:

Modify Assignment 1 to satisfy the following.

1. Add a method *AddSwimmer* to *Club* class so a registrant can be added to the club members. If the registrant is already assigned to a different club this method should throw an exception with a massage: "`Swimmer already assigned to {club name} club".` Also, make sure that registrant's club information is correct. Assume that it will be no more than 20 club members in any club.

2. Add *Club* property to the *Registrants* class to represent a registrant's club affiliation. User of this class can add club affiliation to a registrant by assigning the reference of a *Club* object to this property. Make sure that a registrant is also added to the club member list.

3. Add new field, *noOfLanes*, and corresponding property to a *SwimMeet* class to keep the information about number of lanes in the pool. Change your constructors as required.

4. Make necessary changes to *SwimMeet* class to ensure that course cannot be changed after it is initialized.

5. Add a method *AddEvent* to *SwimMeet* class that will add an event to the swim meet. Assume that there will be no more than 50 events at a meet.

6. Prior to the beginning of the swim event swimmers will enter the event. Each event consists of a number of heats. Each heat represents a number of swimmers racing at the same time. The process of assigning swimmers to heats and lanes where they are going to race is known as seeding. At the end of each heat each swimmer gets the time that they swam the race.
   You need to modify you *Event* class to keep information about the registrants that swim that event. There will be no more than 100 swimmers per any event.

Create a method *AddSwimer* that will add a registrant(swimmer) to the event. If swimmer is added twice to the same event throw an Exception with message "Swimmer {name of the swimmer, reg number} is already entered"
Each swimmer that is added to an event will have a corresponding swim. (Hint: You can use parallel arrays to accomplish this.)

7. When a meet is created and all the swimmers are registered for the events, the swim meet is seeded. Seeding a swim meet means seeding each event. The way the real meets are seeded is rather complicated to implement in this assignment, so we will assign heats and lanes in order. For example, in a pool with 8 lanes the first 8 swimmers are assigned lanes 1 to 8 in heat 1, the next 8 swimmers are assigned lanes 1-8 in heat 2, and so on. Please note that not all swim meets are held in the pool with 8 lanes.
   Add method *Seed* to *SwimMeet* class that will call the *Seed* method of each event. *Seed* method in *Event* class will assign heats and lanes in corresponding *Swim* object for a swimmer.

8. Add *EnterSwimmerTime* method to the *Event* class. This method will accept two parameters: a swimmer and the time that the swimmer made. The method will update appropriate information in the *Swim* object.

Additional requirements:

1. The test harness is provided in attached program.txt file. You must not modify the content of the class. You can change namespace names only.
2. Modify your code from assignment 1 to provide the similar output as in the text file output.rtf
3. You can add any additional methods that may be needed.

Note: This is the first part of the assignment 2 and must be submitted by the deadline in the drop box. The submission should be working as per requirements. This part will not have a separate grade as it will be marked together with part B. However, if not submitted or not mostly working will carry 10% penalty.
The minor changes and fixes to this part are acceptable before the submission of the second part of the assignment 2.

Submission: Submit the solution to Assignment2A drop box by drop box deadline.