Project

Consider a student activity that contains a group of contributors who are divided into instructors and participants.

1. Create class **Contributor** with data members: Name, id and no_of_tasks. Id is a unique number for each Contributor. The no_of_tasks is the number of tasks should be done by the contributor. Add the following member functions:
    - **Non-default constructor** that initializes Name, id and no_of_tasks.
    - **Increment_tasks()** that increases the number of tasks by 1.
    - Setters and getters for the data members.

2. Create class **Student_Activity** with data members: an array of 40 contributor pointers **ArrayOfContributors**. The array of contributors is the contributors who are in the student activity. (Think what will be the relation between the contributors & the student activity) Add the following Members functions:
    - **default constructor** that makes any needed initializations
    - a member function **AddContributor** (Memper * m) that adds a Contributor to the list
    - a member function **DropContributor** (int index) that takes an array index and drops the contributor (that pointed to by the pointer of this index) from the array by: making its pointer points to the last array element and making the pointer of the last element points to NULL then decrementing the elements count of the array.
    - a member function **PrintInfo()** that prints this information of the Student_Activity:
        - the number of its instructors
        - the number of its participants

3. Create two classes for the two types of contributors **Instructor** and **Participant. (**Think what will be the relation between the Class **Contributor** & its types). While adding the following member functions, determine if it is better to define a corresponding function in class **Contributor** and if so, determine whether the function will be virtual, pure virtual or non-virtual:
    - **Non-default constructor** that initializes Name, id and no_of_tasks.
    - **Print()** that prints the type of the contributor, his name an/d number of tasks **Hint:** you may need to use Dynamic_cast to figure out the type of the class.

- **Assign_task()** this function means that there is an instructor will assign a task to another contributor, here are some scenarios:
    - If an instructor assigns a task to a participant then the function **Increment_tasks()** will be called.
    - If an instructor assigns a task to an instructor, the function should output that such an assignment is invalid.
    - If a Participant assigns a task to an instructor, the function should output that such an assignment is invalid.

4. Write the main program to test your classes. You first need to
    - create two object of class Instructor
        - Contributor* Instructor1= new Instructor (“Yassin”,220,5).
        - Contributor* Instructor2= new Instructor (“Omar”,232,3).
    - create three objects of class Participant
        - Contributor* Participant1= new Participant (“Zeina”,238,3).
        - Contributor* Participant2= new Participant (“Yasmin”,230,4).
        - Contributor* Participant3= new Participant (“Ahmed”,249,6).
    - creates a Student_activity object and adds the 5 created contributors to it using AddContributor().
    - calls PrintInfo() function of the Student_activity object
    - Print all the contributors’ details.
    - Let Instructor1 assign a task to Participant1 and then print the details of all contributors
    - Let Participant3 assign a task to Participant1 and then print the details of all contributors
    - Let Participant2 call DropContributor () and then call PrintInfo() function of the Student_activity object.