

Roll Numb	Design [10]	Add Member [10]	Add Friend (F2) [10]	Get friendlist (F) [10]	Get Common Fr [10]	Get Friend Sug [20]	Test Program [10]	Total marks [80]	Comments
cs1401	8	10	10	9	8	14	10	69	Good design of the objects, but constructor should be public. F3. Return the friendlist as an object, not print F4. Return, not print. Also, intersection could be done faster. F5. Again return, not print. Very inefficient implementation of the friend suggestions. Use variable names that can be understood.
cs1402	10	10	10	10	8	16	10	74	Good design, good code. Could do faster implementation of friend suggestions and intersection.
cs1403	4	10	10	8	7	14	10	63	Need to improve the coding practice. Use comprehensible variable names, not m, d, etc. Define member variables and methods to be public or private or protected. You are using the objects as structures. Return objects by methods, do not simply print out. How will the methods be used in a large program otherwise? Faster implementation of intersection and friend suggestion could be done.
cs1404	1	10	10	10	8	16	10	65	Although it is subjective, but social network should be doing most of the work, while the member is treated like a simpler object. Your design is the opposite. As a result, you had to expose the map from social network using a function getMap() which is simply not the way to handle private variables. The map should be internal to social network, and the social network should provide the necessary functionalities to outside classes, not expose its skeleton and let others do whatever they want. This is critical. If you have confusion, please do visit one of your instructors and get it clarified. The variable names should not look like arrayList, map etc. What are those in the context of this particular problem? How will someone build on this code and do anything more?
cs1405	2	10	10	8	8	16	10	64	An interface should define what methods should be provided, not define the internal structure (e.g. the variables stored) of the class. Another class implementing the same interface should have full freedom to store things differently, but has to provide the basic functionalities. This is critical. If you have confusion in understanding, please visit one of your instructors. Getting the friendlist should be just one access. You are first determining the index of the emailid and then querying the hashmap. Why not use the email id as the key? Good test program.
cs1406	0	5	5	5	5	14	10	44	Completely unusable design. If it is not clear by now, please visit one of your instructors and get it clarified again. This is critical. Why use String[1000]? Why initialize arrays this way and not use ArrayList? Why each function takes input from standard input instead of taking objects as arguments and returning objects? You know a lot of the Java syntax, but need to understand the basic principles of OOP. As like others, could implement the intersection and friendsuggestion more efficiently, but that's far from the main concern.
cs1408	6	10	10	10	8	16	10	70	The design is as such fine, but -- (1) why is friendSuggestion not a public method? (2) why have listMutualFriend as a member variable in the class, when you need it only in one method (locally)? Other than these, intersection and friendsuggestion could be implemented more efficiently.
cs1409	10	10	10	10	10	12	10	72	Nice design and nice code. Only complain is the way you are computing friend suggestion. You are going through all the members in the network, and computing number of common friends with the target member. It's even more expensive than going through all friends of friends.
cs1410	6	10	10	10	8	10	10	64	The class Network does not do a justice to its name, it is only a dictionary of names. Instead of printing, return objects by methods. The implementation of friendsuggestion is not even done in a method, you have done it in the main program. It is not reusable.

cs1411	10	10	10	10	8	16	10	74	Good design. Implementation of intersection and friendsuggestion should have been faster, that's the only point.
cs1413	8	10	10	10	8	14	10	70	Design is fine, but instead of methods such as showFriends() use getFriends() and return the array list instead of printing. The printing should only be done in a test program (if one needs to print). Otherwise the code is not reusable. Also, could do faster implementations of intersection and friendsuggestion.
cs1414	4	10	10	10	6	6	10	56	The design is not object oriented, it cannot be reused by someone else. The friendlist is obtained via three for loops, very inefficient. Also, need to get the friendsuggestions, not friends of friends. If you are still confused about the object oriented design, please get it clarified.
cs1415	8	10	10	10	8	16	10	72	Design is fine, but instead of printing lists, return them as objects, otherwise they are not reusable. Mutual friends and friendsuggestions could be done more efficiently.
cs1416	6	10	10	10	8	16	10	70	Friends does not sound like an object. You could merge Friends and SocialNetwork. Why are all methods static? Why print, and not return objects? The intersection and friendsuggestions could be more efficient.
cs1417	9	10	10	10	8	15	10	72	Good design, but why is there no method to return the list of friendsuggestions? If you just print it cannot be reused by another method. Now coming to the friendsuggestion: when you are updating the friendsuggestion, you are not doing incrementally, it is taking the same time as if you did not know the friendsuggestion for this member. Hence, it is better to do it only when required. Consider the case when one friendSuggestion call is made for every 10 friendaddition calls. In your approach, you'd update suggestions everytime even if it is not necessary. Also, though it can return suggestions quickly, adding friends become equally expensive, so there is no gain. Explain, in case your solution is misunderstood.
cs1418	8	10	10	10	8	16	10	72	Do not print, return objects, just like you have done for getFriendList; otherwise how will someone use your method Suggestfriend? The intersection and suggestfriend could be done more efficiently.
cs1421	6	10	10	10	8	16	10	70	isAlreadyFriend etc are public, why not commonFriends? Which methods do you expect outside classes to use? Do not print, return objects. The design needs improvement in terms of reusability. The intersection and the friendsuggestions could be done more efficiently.
cs1422	8	10	10	10	8	16	10	72	Do not print, return objects, otherwise how will someone use your methods? The intersection and suggestfriend could be done more efficiently.
cs1423	8	10	10	10	8	16	10	72	Do not print, return objects, otherwise how will someone use your methods? The intersection and suggestfriend could be done more efficiently.
cs1424	3	10	10	10	8	16	10	67	Essentially you do everything in the Member class. You should use the Member class as an object and SocialNetwork as another object. Do not print, return objects in methods such as commonFriends(), otherwise how will the methods be used? Some of the methods need to be declared public. The implementation of commonfriends and friendsuggestions could be more efficient. You are using the same set of global variables i, j, k, find, num, index etc in both Main and Member class. Note that the goal of object oriented programming is not just getting the current problem to work, rather to write code which can be extended.
cs1425	8	10	10	10	8	16	10	72	Do not print, return objects, otherwise how will someone use your methods? The intersection and suggestfriend could be done more efficiently.
cs1429	4	5	10	10	8	12	10	59	You should not be exposing a HashMap of the Network class outside and let other classes use put to let new members join. Write a method for letting someone joining the Network. Do not print, return objects. The intersection and the friend suggestion could be done faster. The friendsuggestion method is very slow, and the method itself does not even give friendsuggestions, it just checks if two members should be in the suggestion list of one another.

cs1430	9	10	10	10	8	16	10	73	Nicely written and documented code. But, why use one HashMap and one array to store friendlist? Can you not do with one of them? The intersection of friendlists and the friendsuggestions could be done faster.
--------	---	----	----	----	---	----	----	----	--