Course:	**Object-oriented Analysis & Design**	**Course Code:**	**CS-309**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2018**
Duration:	**180 Minutes**	**Total Marks:**	**60**
Paper Date:	**18-Dec-18**	**Weight**	
Section:	**All**	**Page(s):**	**8**
Exam:	**Final**	**Reg. No.**	

Instruction/Notes: Solve the exam on this paper. Do not submit answer sheets. You may use rough sheets but those shouldn't be attached.

Question 1 **15 points**

1. Relate the following design patterns to their respective intent / purpose by matching each entry in the first column to each entry in the second column

Pattern	**Intent/Purpose**
Composite	Allow creation of families of related objects independent of implementation
Singleton	Let objects observe the behavior of other objects so they can stay in sync
Factory method	Compose objects into tree structures. Let clients treat primitives & compositions uniformly.
Abstract factory	Abstract creational method that lets subclasses decide which class to instantiate
Observer	Ensuring a class has only one instance

2. Relate the following design characteristics to their respective meaning by matching each entry in the first column to each entry in the second column

Characteristic	**Description**
Cohesion	The degree of interaction between different classes
Coupling	Hides implementation details and provides easy-to-use interface
Abstraction	The characteristic of being easy-to-change
Modularity	The degree of similarity of the constituent parts
Maintainability	Multiple small independent units

3. Select the best option available:

(a) In a good design, coupling should be:
(i) Low (ii) High (iii) Medium (iv) Readable (v) Reusable

(b) In a good design, cohesion should be:
(i) Low (ii) High (iii) Medium (iv) Readable (v) Reusable

(c) In object-oriented methodology, modular unit of implementation is:

(i) object (ii) method (iii) function (iv) class (v) package

(d) Object can be defined as:

(i) an entity with a name and state
(ii) an entity with a name and behavior
(iii) an entity with a state and behavior
(iv) an entity with a name, state and behavior
(v) an entity with identity, state and behavior

(e) Polymorphism requires the following conditions for implementation:

(i) overriding, subtyping and static binding
(ii) overriding, subtyping and dynamic binding
(iii) overloading, subtyping and static binding
(iv) overloading, subtyping and dynamic binding
(v) overriding, subtyping and method binding

Question 2 **10 points**

Softec is a prestigious event at FAST-NU. A number of different events are held under the Softec umbrella. These include Software Competition, Programming Competition, IdeasXtreme, etc. Numerous participants from various universities and institutes across the country participate and register for these events.

Softec society plans to develop an online registration system for all these events.

Participants can register for multiple events either individually or in teams according to event requirements.

Each event has an associated registration fee that is to be paid manually using a Bank Draft. Registration of participant(s) is confirmed only once the registration fee is received by Softec society.

Some events may have some special requirements for registration. For example, Software Competition requires from the participating team to submit an Abstract of their software at the time of registration. This Abstract needs to be evaluated and approved by a panel of evaluators before the registration is accepted.

Similarly, other events may have their own special requirements

Course:	**Object-oriented Analysis & Design**	**Course Code:**	**CS-309**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2018**
Duration:	**180 Minutes**	**Total Marks:**	**60**
Paper Date:	**18-Dec-18**	**Weight**	
Section:	**All**	**Page(s):**	**8**
Exam:	**Final**	**Reg. No.**	

Initially, participants submit their registration application. Once all the registration requirements for the concerning event are fulfilled and registration fee is received, the registration is confirmed.

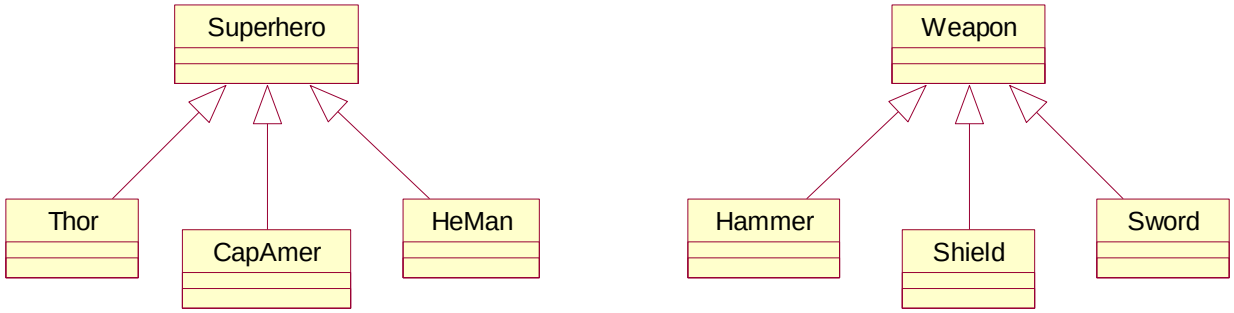
Participants are notified of the confirmation through an email.

Develop a use-case diagram for Softec Registration System, illustrating the actors and their respective use-cases. Show important inclusions and extensions, if applicable.

Course:	**Object-oriented Analysis & Design**	**Course Code:**	**CS-309**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2018**
Duration:	**180 Minutes**	**Total Marks:**	**60**
Paper Date:	**18-Dec-18**	**Weight**	
Section:	**All**	**Page(s):**	**8**
Exam:	**Final**	**Reg. No.**	

Question 3 **10 points**

Consider the following superheroes and their weapons:



The following code assigns a weapon to each superhero:

```
#include <typeinfo>

Weapon* createWeapon(Superhero* ptr) {
      string n = typeid(*ptr).name();
      if (n == "Thor")
            return new Hammer();
      else if (n == "CapAmer")
            return new Shield();
      else if (n == "HeMan")
            return new Sword();
}
```

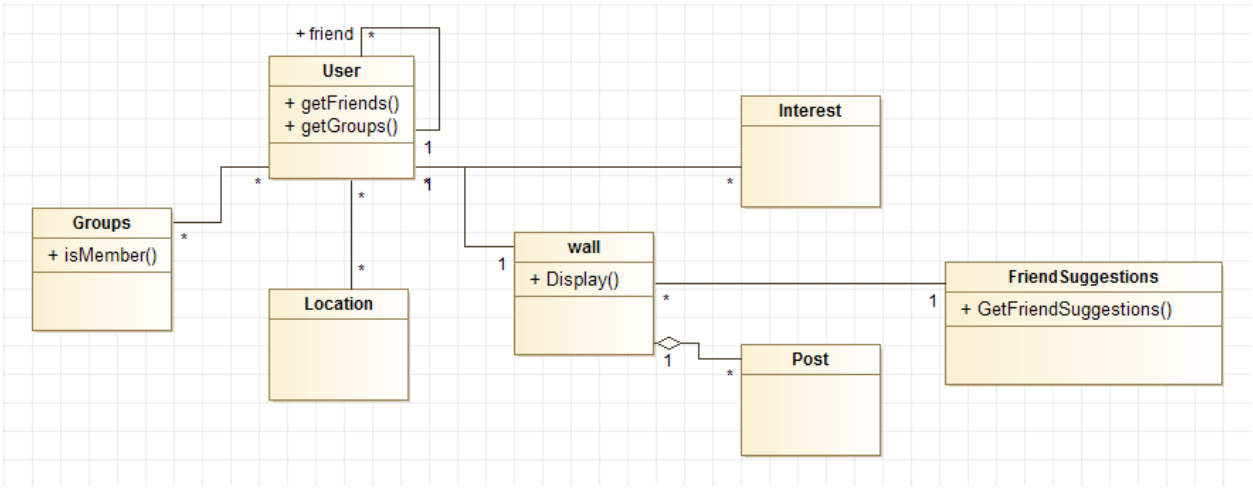
Rewrite / Refactor the above program to improve the program design. If you add any new function(s) during the process then show their code as well.

Course:	**Object-oriented Analysis & Design**	**Course Code:**	**CS-309**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2018**
Duration:	**180 Minutes**	**Total Marks:**	**60**
Paper Date:	**18-Dec-18**	**Weight**	
Section:	**All**	**Page(s):**	**8**
Exam:	**Final**	**Reg. No.**	

Course:	**Object-oriented Analysis & Design**	**Course Code:**	**CS-309**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2018**
Duration:	**180 Minutes**	**Total Marks:**	**60**
Paper Date:	**18-Dec-18**	**Weight**	
Section:	**All**	**Page(s):**	**8**
Exam:	**Final**	**Reg. No.**	

Question 4 **10 points**

Consider following partial class diagram of an example "Facebook" system



There is a "Display Wall" use case which <<includes>> another use case "Get Friend Suggestions"

Prepare a sequence diagram for the use case "Get Friend Suggestions". You have to use only the provided functions. You are not allowed to create new functions in any class.

- getFriends(): returns a list of friends for a user
- getGroups(): returns a list of groups for a user
- isMember(): takes one parameters (user) and return True if the user is a member of the group otherwise returns False

You have to provide complete flow for the public method GetFriendSuggestions(). This method accepts an object of the current user. It creates a list of friend suggestions based on the following rule:

- A person is suggested as a friend if he is a friend of an existing friend of the current user, and
- He shares a common group with the current user, i.e. both the user and the person are members of a single group

Remember that sequence diagram shows interaction between objects of classes. There can be multiple object of the same class in a sequence diagram if required. Show arguments and return types of each message passed between the objects in the sequence diagram.

Course:	**Object-oriented Analysis & Design**	**Course Code:**	**CS-309**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2018**
Duration:	**180 Minutes**	**Total Marks:**	**60**
Paper Date:	**18-Dec-18**	**Weight**	
Section:	**All**	**Page(s):**	**8**
Exam:	**Final**	**Reg. No.**	

Course:	**Object-oriented Analysis & Design**	**Course Code:**	**CS-309**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2018**
Duration:	**180 Minutes**	**Total Marks:**	**60**
Paper Date:	**18-Dec-18**	**Weight**	
Section:	**All**	**Page(s):**	**8**
Exam:	**Final**	**Reg. No.**	

Question 5 **15 points**

Consider a software for managing discussion forums. A forum is a collection of topics that may have sub-topics. For instance, an automobiles related forum (e.g. PakWheels) can have several topics e.g. Technical, Buy & Sell, Vintage (old) cars, etc. Each topic may have sub-topics, for instance, Technical discussion can relate to Electrical, Mechanical, Body work, etc. Such categorization may be done to any level of depth. A registered user can post a question on a topic. Others may post reply. When a reply is made, all the users participating in the discussion on that question are notified through email.

Develop a class diagram for the above software, making use of relevant design patterns. There is no need to show the attributes and methods, unless required to show the applicability of relevant design pattern.