QS1;

a)

NO, Here University is a superclass and COMSATS is a subclass. Here university cannot be COMSATS but COMSATS is a university.

b)

YES, here currency is a superclass and rupee is a subclass. Rupee is a currency.

c)

NO, Here social media Account is superclass and Facebook account is subclass. Facebook account can’t be Social media account social media account is a Facebook account.

Qs1(b):

a)

Instance

b)

BOTH

c)

Instance

Qs2:

The existence of a new class in the project with 5 classes can be justified on the following grounds:

* We can define a class within another class which is called the nested class.
* We can define a new subclass with inheritance relationship to an existing superclass.
* We can define a new class with aggregation relationship to an existing class.

QS3:

Instances of the UI and architectural classes are created when a program runs, and closed when a program ended. The vast majority of these classes will in general be straightforwardly reused from class libraries, or else represent subclasses if library classes.

QS4:

Services:

* Allows customer to contact with customer service
* Allows system to handle data of frequent flier
* Provide a loyalty card to regular customers.

SLOTS:

* Allows the system to add data of customer
* Allows the system to update/edit the data
* Allows the system to remove data of customer
* Allows the user to view profile(data) of his personal Info

HOOKS:

* Cardholders will be promoted first.
* Cardholder will get free home delivery

Qs5:

a)

"Delegation Pattern"

* In delegation pattern you have components that behave identically. As delegatin pattern keeps from unnecessary interconnection between various pieces of framework. So we use Delegation design here on the grounds that we don't need unnecessary interconnections.

b)

“Adapter Pattern”

By Adapter design we will make a adaptor class that will acquire the class we wish to utilize and adjust its code in our class and afterward by polymorphism we utilize the adaptor class and utilize the changed code in the other class without obliterating the first code