Question # 01:

Explain the concept of pure virtual functions, abstract base classes and interfaces. Discuss the role of abstract classes in designing class inheritance.

Answer:-

Pure Virtual Function:-

A pure virtual function is a function that has no body and is assigned to the 0 value in the base class. It enables polymorphic behaviour.

Abstract Base Class:-

Abstract class is a class that has one or more pu virtual functions. It cannot be instantiated, but it can be used for polymorphism.

Interface Class:-

It is a special kind of abstract base class that has only ~~one~~ virtual function which is pure and other data members. It defines a common beha which can be implemented by different classes.

Role of Abstract class:-

The role of Abstract class in designing class inheritance is to provide a common interface a behaviour for the derived classes. It also enfor the derived classes to override the pure virtual f Hence abstract classes behaves as a contract be the base class and the derived class.

Part # c

Explain the concept of association, aggregation and composition in object-oriented programming. Provide an example for each type of relationship in a real world context.

Answer:-

Association:-

Association is a relation between two or more classes that are otherwise unrelated. According to this context one class can use another class without depending on it. It can be one-to-one, one-to-many, many-to-many, or many-to-one. It is a weak relationship.

Example:-

A doctor and patient are related and both can exist independently. A doctor can have many patients and a patient can visit multiple doctors.

Aggregation:-

It is a special type of relationship. It is represented by a "has-a" or "a-part-of" relationship. It means one class can has another class as member variable, but they have their own implementations.

Example:-

An employee and a department of a university are aggregated with eachother. A department can have many employees but employee does not belong to a single department. Both can exist separately.

Composition:-

It is a type of association in which there is a

"whole part" or "owns a" relationship. It means one class can have other class as member variable and hence they both share same life-cycle. It is unidirectional

Example:

A car and its engine both are composed with eachother. Both cannot exist without one another. A car has an engine and the engine belongs to only that car.

Question # 02:-

Provide a class diagram for given scenario on paper

Answer:

Class Diagram :-