Q. Difference between is-a rule and has-a rule.

Check page number 25 of 'C++_Note_Part5'. It is same as 'is-a-relationship' and 'has-a-relationship'

Q. Advantage and disadvantages of inheritance.

Advantages of inheritance:

- Inheritance supports reusability. The features that are defined at parent class can be reused in child class.
- The codes of the base class will already be tested and debugged. Hence, reusability via inheritance will enhance reliability.
- Inheritance will avoid duplication of code.
- The methods of the base class can be overridden in the child class and hence better implementation of the base class methods can be implemented in the derived class.
- The development and maintenance costs will be reduced.
- The time and efforts to develop the software will also be reduced.
- The codes are easy to debug.
- It is easy to partition the work.

Disadvantages of inheritance:

- One of the major drawbacks of inheritance is that the base and child classes are tightly coupled. One cannot be used independently of each other.
- A change in base class will affect the child class.
- The execution speed is reduced as it takes time to jump across various levels of inheritance.
- The data members in the base class are often left unused which leads to memory wastage.
- Inappropriate use of inheritance will cause complexity in the program.

Q. Advantages and disadvantages of friend function:

Advantages of friend function:

- It can access the private data member of class from outside the class.
- It acts as the bridge between two classes by operating on their private data.
- It allows sharing private class information by a non-member function.
- It is able to access members without the need of inheriting classes.

Disadvantage of friend function:

- It violates the law of data hiding principle by allowing access to private members of the class from outside the class.
- The data integrity is breached.
- It is conceptually complex and messy.
- Size of the memory occupied by the objects will be high.

Q. Do you find any advantages of adopting RDD? Explain with the help of suitable examples.

Responsibility-Driven Design (RDD) is an object-oriented design technique that is driven by an emphasis on behaviors at all levels of development. It focuses on: what action is the object responsible for and what information does the object share. It identifies what action must be accomplished and which object will accomplish them.

RDD helps to maximize the abstraction mechanism. It categorizes the responsibilities of objects and initially hides the distinction between data and behavior. It tries to avoid dealing with the details.

Consider a system as a room in which objects live in. When a request is sent to the system for performing some tasks, the objects work together to fulfill the request. The RDD identifies which object is suitable for which task and it is given some responsibility. The RDD design also emphasize on what information the object will share. So, the objects living in the room are given responsibilities and they do what they are capable of doing. These objects then share the information and delegate the remaining things to its collaborators (other objects/components).

There are many passive concepts from the real world which do not have any behavior. For e.g. a reservation ticket of a movie theatre is a passive concept which becomes active if a real world client books and pays for it. The responsibilities are needed to make the passive things active and so the responsibilities are assigned to them.

While any system might be complicated, the objects within the system and their associated behaviors shall remain easy to understand. The RDD helps to achieve this and as a result, any updates or changes can easily be made.

In addition to that, by the help of RDD, a low coupling and high cohesion is achieved. Cohesion is used to indicate the degree to which an object has a single, well-focused purpose. Higher the cohesiveness of the object better is the design. Coupling indicates how the objects interact with each other. So, if there is low coupling between objects, it will be easier for modification and reuse and software development becomes better.

Q. Difference between the concept of computation as simulation and responsibility implies non-interference.

Check page number 1 of 'C++_Note_Part6' and page number 6 of 'C++_Note_part7'.

You can explain these two topics in separate paragraphs. If you can relate any points that can illustrate their difference then focus on those points.

Q, Explain about implementation and integration of component with real word example.

Check Page 7 and 10 of 'C++_Note_Part6'. Include one or more real world examples in its explanation.

For example, you can take examples which are given in the topic 'CRC Cards'.