Q1 :

A : List three different benefits of building a DS.

- connecting users & reference
- Openness
- Transparency
- Scalability
- Enhance availability

B : What is the different between Load balance & Fault tolerance in modern DS.

- Fault tolerance : Failure of a component should not result in failure of the whole system
- load balance : Distribute load among different computers.

C : Briefly describe the difference between a fault and failure that my occur in a DS.

- **Failure:** service no longer complies with its specification
- **Fault:** cause of a failure (crash of a component)

D : Briefly describe the difference between user authentication & user authorization while securing a DS

- Authentication : telling the system who you are by providing username and password.
- Authorization : things you can do according to who you are

Q2 :

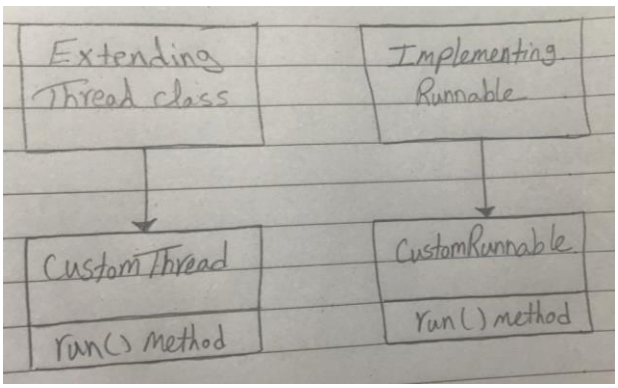
A : What is the information is required to create JAVA Socket ?

- IP address of server
- Port number

B : Draw a diagram that illustrate two possible ways to create threads in java, describe the difference between the two ways and when to use each of them

Key differences and when to use each approach:

Extending Thread Class:
it gains full control over the thread
Use when: Customizing the thread's behavior
Implementing Runnable Interface:
does not give developers any control over the thread
Use when: Flexibility is required



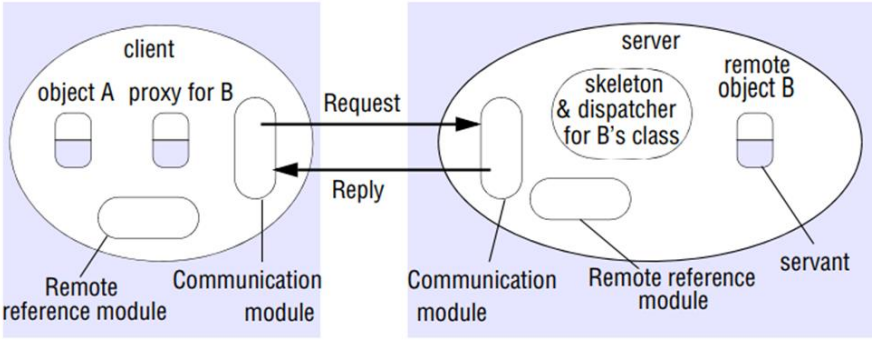
C : In Java Multithreading, when multiple threads try to access the same object, for example a bank account, an unpredictable behavior can happen to the shared object, Describe by an example how this unpredictable behavior can occur & how to resolve this issue

In the bank account, if two threads concurrently wants to withdraw funds from the same account, One thread may read the account balance, and before it updates the balance, another thread may read and modify the same balance, the final balance may not be what was expected

To resolve this issue : synchronization mechanisms, such as using the 'synchronized'

D : Briefly describe the purpose of java RMI. draw a sample diagram that shows the java RMI architecture and describe briefly all its components

RMI is used in the transmission of data between client and remote server



Components:

- **remote reference module:** provide addressing to the proxy object
- **proxy:** used to implement stub and provide transparency to the client
- **communication module:** responsible for networking
- **dispatcher:** select the proper skeleton and forward message to it
- **skeleton:** un-marshal the request and call the remote object

E : Briefly describe the difference between a faulty node and a malicious node in the Blockchain network

Faulty nodes are not malicious and typically result from software bugs, network failures, or glitches
A **malicious node** automatically behaves in a harmful or adversarial way.

F : Draw a diagram that shows the generic structure of a blockchain network