Exercise

- Implement a simple RMI in Java or a similar object-oriented programming language. If you want to use a different language confirm that with me.
- The implemented RMI should have the following features:
- Obviously should allow calling of remote methods!
- To the programmer, the remote method calling should look like a local method call
- You can not use classes in the Java RMI package (only exception to this is the RemoteException!)

Exercise – continued

- Multiple clients should be able to connect to remote object
- Should use inheritance to enforce method implementation at client and server side.
- Follow good programming practices in designing your application.
- The binding of Client and Server could be hardcoded or read from a config file.
- Having a dynamic binding mechanism has bonus marks

Exercise – continued

- Should support at least the following primitive types for arguments and return type: integer, boolean, char.
- Support for additional types and objects is optional with bonus marks.
- If you are making assumptions, clearly specify them.
- Read the following to better understand the Java RMI:

https://docs.oracle.com/javase/6/docs/technotes/guides/rmi/hello/hello-world.html

Exercise – continued

- After implementing the necessary classes/interfaces, create a sample application. In this application, the remote object has information about bank accounts. Clients should be able to deposit, withdraw, and check the balance (getBalance method) for a specific account at the remote object.
- Whatever language you use, no libraries that implement RPC/RMI should be used (e.g. Java RMI, Python PYRO, etc.)
- Submit report by 23:55 on 20th Dey