Exercise - RMI

- Implement a simple RMI in Java, C#, or Python. 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!)
- Whatever language you use, no libraries that implement RPC/RMI should be used (e.g. Java RMI, Python PYRO, etc.)

Exercise – continued

- Multiple clients should be able to connect to remote object
- Should use inheritance to enforce method implementation and consistency at client and server sides.
- Follow good programming practices in designing your application.
- The binding of client and server should be done **dynamically**.
- Dynamic binding mechanism should be based on **class name** and **class implementation version**.

Exercise – continued

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

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

- I suggest running a simple RMI program before designing your own

Exercise – continued

- After implementing the necessary classes/interfaces, create a sample application. In this application, the remote object has information about some entity (e.g. bank accounts). Don’t use the given example (i.e. bank account). Produce a different example for your implementation.
- In the bank example, clients should be able to deposit, withdraw, and check the balance (getBalance method) for a specific account at the remote object. In your application, at least three methods should be implemented. (doing writing and reading operations among them)

Report

- Submit report by 23:55 on Saturday 12th Azar
- Your report should describe your design and your evaluation of it (its strength and weaknesses)
- Furthermore, describe your sample application briefly.
- PDF file of the report as well as a zip file containing the source code should be submitted before the deadline.