Distributed Data Analytics Project
(Max team members: 4)
(Longest and Shortest Word in a File using RMI)

RMI Setup:

- Create a folder for all needed files
- Create a interface file with extension .java
- Create an implementation file with extension .java
- Create a server file with extension .java
- Create a client file with extension .java

Example for the interface file code:

```
import java.rmi.Remote;

public interface AddI extends Remote
{
public int add(int x,int y) throws Exception;
}
```

Example for the implementation file code:

```
import java.rmi.server.*;

public class AddC extends UnicastRemoteObject implements AddI
{

public AddC() throws Exception
{

super();

}

public int add(int x,int y){

return x+y;
}
}
```

Example for the Server file code:

```
import java.rmi.*;


public class Server
{
public static void main(String a[]) throws Exception
{


AddC obj = new AddC();
Naming.rebind( "ADD" ,obj);
System.out.println("Server Started");

}
}
```

Example For the Client file Code:

```
import java.rmi.*;


public class Client
{


public static void main(String a[]) throws Exception
{

AddI obj = (AddI)Naming.lookup("ADD");
int n = obj.add(5,4);
System.out.println("addition is : " + n);

}
}
```

Then

- **Open a terminal or cmd and set directory to the folder containing your files path's**
- **Try the command javac**
- **If you get an unknown command error then try the following command: set path="c:\Program Files\Java\jdk1.8.0_20\bin" or wherever it's located on your system, make sure you have java on your system**
- **Compile all files using javac *.java**
- **You will find 4 new files created with the same names as the previous files but with a .class extension**
- **Use the following command rimc yourimplementationfilename to create the stub and skeleton without extension you will find a new file created and now you have total of 9 files**
- **Start the registry by the command start rmiregistry on windows or rmiregistry & on mac/linux**
- **Now open 2 new terminals/cmds and set directory again to the same folder and now it's time to run each of server and client using the command java server in one terminal and java client in the other one**
- **After setting and running the example now you are ready to change the example into the required project which is**

Finding the longest or shortest word in a file

- **Open a text file using I/O on files and parse the file and find the longest and the shortest word.**