# COMP 6321: Distributed Systems Design

# Lab Instructions

RMI

May 31, 2022

## 1 General Information

**Lab Date:** Tuesday, May 31$^{th}$, 2022.

Your lab instructor will provide you with instructions on how to do this lab activity.

## 2 Overview

The purpose of this lab is to get a hands-on experience on Remote Method Invocation (RMI) technology using Java. In this lab you develop a simple RMI which counts number of words inside the dictionary by a remote procedure call technology, where it is used for java object serialization in the transferring of data.

## 3 Reminders on RMI

RMI (Remote Method Invocation) is an API used to access objects running on another JVM(Server-side). It is mainly used for the creation of distributed systems and is provided in Java Rome. Stub and Skeleton are the two objects used for handling communication between client and server. The following figure shows an overview of RMI.

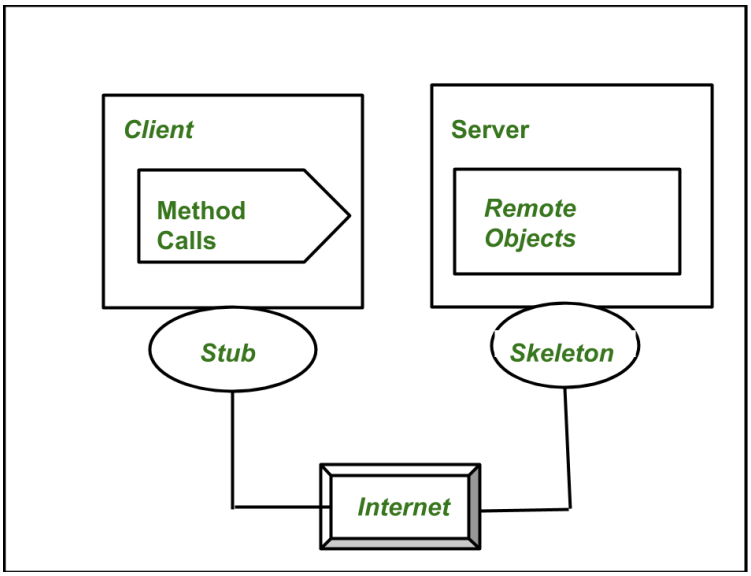


Figure 1: Working of RMI

- **Stub Object:** The stub object on the client machine builds an information block and sends this information to the server.
- **Skeleton Object:** The skeleton object passes the request from the stub object to the remote object. RMI contains a rmiregistry that holds all the server objects. The server binds all the objects to the registry and then the client fetches the object from the respective registry after which the client invokes the methods using the fetched objects.

# 4 Instructions

Before we give you the details of the exercise, we will first tell you which files you have to implement for a RMI client/server program. So, to write a RMI client/server program you need to write the following:

- An interface for the remote object: **Dictionary.java**
- An implementation of the remote object: **DictionaryImpl.java**
- A server that will run the remote object: **Server.java**

- A client to access the server: **Client.java**

## 4.1 Project Setup

Clone the template from the first tutorial on Github:

`https://github.com/COMP6231/T04`

See `Dictionary`, `DictionaryImpl`, `Server` and `Client` for details.

## 4.2 Interface

In this exercise, you will write a Dictionary RMI Client/Server program. The first thing that you wanna to write is of the interface of the Dictionary. In our case, the interface can only perform counting the words.

- The interface has to adhere to the following rules:
- The interface must extend java.rmi.Remote All methods specified in the interface must throw the java.rmi.RemoteException exception

## 4.3 Dictionary Interface

Now that you have the interface you need the implementation of the remote object. Inside the **DictionaryImpl.java** you have implementation of the interface for the remote object. This class is provided for you.

## 4.4 Server

The next item on the list of the files that we have to implement is the server that will run the remote object. You need to complete **stub** object. This object must inherit from the **java.rmi.server.UnicastRemoteObject** class. Then you need to create a new registry object from the **java.rmi.registry.Registry** class and bind the remote object's stub in it.

### 4.5 Client

The last class that we have to implement to complete our exersie is the **client**. You need to create object from the **java.rmi.registry.LocateRegistry** class and try to connect to the registry on the **port: 6231** and **host: localhost**. After that, you must look up a stub with name "Dictionary". (**Hint:** you can get help from lookup() method in **java.rmi.registry.Registry** class).

## 5 Running the Project

Compile and Run the Server first. Then you can run the client. Your program must pass the two test cases and the expected result must be like below:

- response (String line): Concordia=2, at=2, student=1, I=1, This=1, course=1, is=1, am=1, my=1, favourite=1, University.=2
- response (String[] lines): Concordia=2, at=2, student=1, I=1, This=1, course=1, is=1, am=1, my=1, favourite=1, University.=2

# References


1. Getting Started Using Java™ RMI:

   https://docs.oracle.com/javase/7/docs/technotes/guides/rmi/hello/hello-world.html

2. Getting Started with Java RMI:

   https://www.baeldung.com/java-rmi

3. Remote Method Invocation in Java:

   https://www.geeksforgeeks.org/remote-method-invocation-in-java/?ref=lbp