I turned to Oracle, the owner of this technology itself, for an example tutorial and found what seemed like a good tutorial exercise. The website is:  
<http://docs.oracle.com/javase/tutorial/rmi/overview.html>

After setting up all necessary skeletons, RMI interfaces, server and client side implementation it was time to see it working. At that point, several things got in the way, the first of which turned out to be the security policy written by Oracle as part of their online tutorial. The security policy saved in a ‘server.policy’ file which look like:

grant codeBase "file/Users/tanvir/Documents/Developments/Workspace/RMISecurityInvestigation2ServerSide/security/" {

permission java.security.AllPermission;

};

would not work due access permission issue as following:

ComputeEngine exception:

java.security.AccessControlException: access denied (java.net.SocketPermission 127.0.0.1:1099 connect,resolve)

at java.security.AccessControlContext.checkPermission(AccessControlContext.java:376)

at java.security.AccessController.checkPermission(AccessController.java:549)

at java.lang.SecurityManager.checkPermission(SecurityManager.java:532)

at java.lang.SecurityManager.checkConnect(SecurityManager.java:1034)

at java.net.Socket.connect(Socket.java:522)

at java.net.Socket.connect(Socket.java:476)

at java.net.Socket.<init>(Socket.java:373)

at java.net.Socket.<init>(Socket.java:187)

at sun.rmi.transport.proxy.RMIDirectSocketFactory.createSocket(RMIDirectSocketFactory.java:22)

at sun.rmi.transport.proxy.RMIMasterSocketFactory.createSocket(RMIMasterSocketFactory.java:128)

at sun.rmi.transport.tcp.TCPEndpoint.newSocket(TCPEndpoint.java:595)

at sun.rmi.transport.tcp.TCPChannel.createConnection(TCPChannel.java:198)

at sun.rmi.transport.tcp.TCPChannel.newConnection(TCPChannel.java:184)

at sun.rmi.server.UnicastRef.newCall(UnicastRef.java:322)

at sun.rmi.registry.RegistryImpl\_Stub.rebind(Unknown Source)

at engine.ComputeEngine.main(ComputeEngine.java:30)

After hours of effort when nothing was found wrong while support information is very limited, one of the blogs in orcle-community came handy (<https://community.oracle.com/thread/1177999?start=0&tstart=0>) and this turned out to be a flaw in the security policy itself.

Once the security policy is rectified, the application was attempted to run again with appropriate JVM arguments. This time there was a *ClassNotFoundException* exception which didn’t make sense at all, as the exception suggests the compiled class doesn’t exist in the classpath which it superficially wrong when the class clearly exists. After much effort to identify the real problem and days of investigation, this turned out to be a problem with a configuration not being set, namely ***CodebaseOnly*** configuration which is only been introduced in java 7.

ComputeEngine exception:

java.rmi.ServerException: RemoteException occurred in server thread; nested exception is:

java.rmi.UnmarshalException: error unmarshalling arguments; nested exception is:

java.lang.ClassNotFoundException: compute.Compute

at sun.rmi.server.UnicastServerRef.oldDispatch(UnicastServerRef.java:419)

at sun.rmi.server.UnicastServerRef.dispatch(UnicastServerRef.java:267)

at sun.rmi.transport.Transport$1.run(Transport.java:177)

at sun.rmi.transport.Transport$1.run(Transport.java:174)

at java.security.AccessController.doPrivileged(Native Method)

at sun.rmi.transport.Transport.serviceCall(Transport.java:173)

at sun.rmi.transport.tcp.TCPTransport.handleMessages(TCPTransport.java:556)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run0(TCPTransport.java:811)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run(TCPTransport.java:670)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)

at java.lang.Thread.run(Thread.java:744)

at sun.rmi.transport.StreamRemoteCall.exceptionReceivedFromServer(StreamRemoteCall.java:275)

at sun.rmi.transport.StreamRemoteCall.executeCall(StreamRemoteCall.java:252)

at sun.rmi.server.UnicastRef.invoke(UnicastRef.java:378)

at sun.rmi.registry.RegistryImpl\_Stub.rebind(Unknown Source)

at engine.ComputeEngineStarter.main(ComputeEngineStarter.java:40)

Caused by: java.rmi.UnmarshalException: error unmarshalling arguments; nested exception is:

java.lang.ClassNotFoundException: compute.Compute

at sun.rmi.registry.RegistryImpl\_Skel.dispatch(Unknown Source)

at sun.rmi.server.UnicastServerRef.oldDispatch(UnicastServerRef.java:409)

at sun.rmi.server.UnicastServerRef.dispatch(UnicastServerRef.java:267)

at sun.rmi.transport.Transport$1.run(Transport.java:177)

at sun.rmi.transport.Transport$1.run(Transport.java:174)

at java.security.AccessController.doPrivileged(Native Method)

at sun.rmi.transport.Transport.serviceCall(Transport.java:173)

at sun.rmi.transport.tcp.TCPTransport.handleMessages(TCPTransport.java:556)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run0(TCPTransport.java:811)

at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run(TCPTransport.java:670)

at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)

at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)

at java.lang.Thread.run(Thread.java:744)

Caused by: java.lang.ClassNotFoundException: compute.Compute

at java.net.URLClassLoader$1.run(URLClassLoader.java:366)

at java.net.URLClassLoader$1.run(URLClassLoader.java:355)

at java.security.AccessController.doPrivileged(Native Method)

at java.net.URLClassLoader.findClass(URLClassLoader.java:354)

at java.lang.ClassLoader.loadClass(ClassLoader.java:425)

at sun.rmi.server.LoaderHandler$Loader.loadClass(LoaderHandler.java:1206)

at java.lang.ClassLoader.loadClass(ClassLoader.java:358)

at java.lang.Class.forName0(Native Method)

at java.lang.Class.forName(Class.java:270)

at sun.rmi.server.LoaderHandler.loadClassForName(LoaderHandler.java:1219)

at sun.rmi.server.LoaderHandler.loadProxyInterfaces(LoaderHandler.java:729)

at sun.rmi.server.LoaderHandler.loadProxyClass(LoaderHandler.java:673)

at sun.rmi.server.LoaderHandler.loadProxyClass(LoaderHandler.java:610)

at java.rmi.server.RMIClassLoader$2.loadProxyClass(RMIClassLoader.java:646)

at java.rmi.server.RMIClassLoader.loadProxyClass(RMIClassLoader.java:311)

at sun.rmi.server.MarshalInputStream.resolveProxyClass(MarshalInputStream.java:255)

at java.io.ObjectInputStream.readProxyDesc(ObjectInputStream.java:1558)

at java.io.ObjectInputStream.readClassDesc(ObjectInputStream.java:1514)

at java.io.ObjectInputStream.readOrdinaryObject(ObjectInputStream.java:1771)

at java.io.ObjectInputStream.readObject0(ObjectInputStream.java:1350)

at java.io.ObjectInputStream.readObject(ObjectInputStream.java:370)

... 13 more

It has been an eventful experience to get RMI working on my Macbook. After days of trying when it was finally deployed and tried on the university’s Unix box, it worked!