P1 Grading - Comments and Status

- We’re working on it!
- Some common issues:
  - Jar files are not a substitute for build instructions
  - Misnamed handin files
  - Please compile and test your code on GHC/Andrew machines
RMI Overview
Major Components

- Remote object references
- Registry
- Stubs
- Dispatcher/Skeletons
- Client/Server
Remote Object Reference

- The information you need to “find” an object
- All objects which implement “Remote” are passed by reference
- Other objects are passed by value (serialized)
Registry

- `bind()` - Adds a remote reference for an object to the registry
- `lookup()` - Gets a remote object reference for an object based on a “name”
- `rebind()` - Updates a reference in the registry
- `unbind()` - Removes a reference from the registry
Stubs

- Responsible for marshaling method invocations
- Can be created by a stub compiler or use an invocation handler (optional for you for the RMI)
- Clients need the .class files for the stubs
Dispatcher/Skeletons

- Both are responsible for unmarshaling the method invocation
- Skeletons - like stubs, one for each method
- Dispatcher - single object which can unmarshal any method invocation
- Up to you which to implement
Client/Server

- Client
  - Looks up objects from server
  - Requests remote method invocations
- Server
  - Hosts some objects by adding references to the registry
  - Runs a registry, dispatcher
- Please provide working, runnable examples of a client and a server in your project
Where do I start?

- Do try out using Java RMI first to get an idea of how your library should look (in terms of the interfaces it provides, usage, etc)
- Follow the plan of action presented in the writeup
What is expected

- Remote object references
- Registry
- Dispatcher/Skeletons
- Stubs/Invocation handler
- Example server and client
- A report with instructions on how to run the examples
Talk about your major design decisions, things that are likely different from project to project. What are the tradeoffs of your decisions?

Tell us about bugs, unimplemented pieces.

Tell us how to build, run, test your code. This should work on GHC/Andrew machines. Please make this easy for us! :)

Tell an application programmer how to set up servers and clients using your RMI facility.
What is not required

- Stub compilers
- Mechanism for downloading .class files
- Garbage collection of remote objects
XML RPC
Other FAQ

- Allowed to work with one partner
- Code can use libraries as long as it does not do RMI/RPC for you ;)
- Report is an important part of your grade
Questions