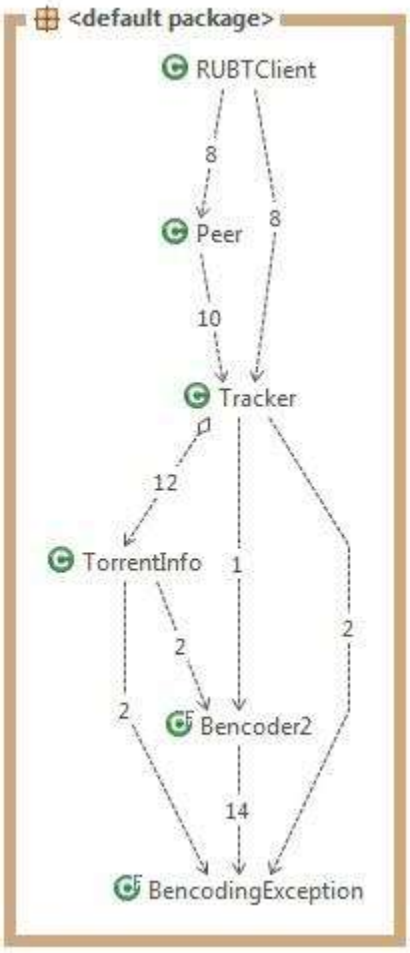
Our classes kind of depend on each other but can easily be changed in later steps of the project. We have two classes peer and tracker. Tracker sets up the tracker and all that data is passed on to peer which uses that information to send a handshake to the peer, get the return message and then when everything is sorted out, begin the download process.



Tracker class:

The tracker class is used to store all the information from the torrent file using the bencoder by hashing the data. Once we have all the information we can create the URL to connect to the tracker. Once connected to the tracker we are able to see a list of peers and then we distinguish the appropriate one for this assignment. From there we are able to get the interval and the port and IP of the peer. This port and IP allows us to connect to a socket which is used in the peer class.

Peer class:

In the peer class have two fields that are Data input and output streams used for communicating between the client and the peer. To begin we have a method that sets up the bit protocol and we send that to the peer. Once that is sent and everything worked the way it should we receive a bitfield. From there we request from the peer. If the appropriate message is returned then we unchoke and start the steps to download. We then have the communicationSeed method that just verifies that everything is right and that we can start the download if the method returns true. Then we start downloading and keep downloading each different part.

RUBTclient:

This is just the main class where we call all the methods in the peer and tracker class.