Timothy Choi	150001490
Michael Shafran	145000003
Prasant Sinha	144007660

Phase 1:

The objective of this project is to create a BitTorrent client that can interface with a tracker and a single peer while downloading a file from that same peer. This client will also read a .torrent file containing metadata for the file requested to be downloaded. When the file has completed downloaded, it will then be saved to the local hard disk.

RUBTClient.java:

This is where the main class is held and where the client knows how to respond to the user based on the arguments that are given to the program. This class looks for 2 arguments, the location of the torrent file and the name of which the file will be downloaded. A parser is initialed using TorrentInfo and the client makes a request to retrieve the list of peers available. From there the client can connect to the peers and establish a download for the requested file.

Parser.java:

This is where a lot of the responses are stored for the end user such as informing the user if the file has completed downloading, shows the number of peers to the user, how much downloaded/uploaded, etc. All of this information is valuable because it shows the progress of the .torrent file in use.

Peer.java:

This class simply just stores the peer ID, the port used, and the peer's IP address.

Typechange.java:

This class converts some of the strings into bytes so the client can understand the given keys.

Handshake.java:

This class handles the sockets for sending the handshake and receiving the handshake back from the other peer. A new handshake is sent per peer where a handshake is established with. For the handshake to occur, there needs to be an open socket, the torrent info needs to be available, and the peer id.