CPSC 441- Assignment 2singm, Name: Farica Mago, UCID:30111924

Working of the program:

After sending the request to the server, I have opened a buffered input stream object called **in** and an integer variable called **byte_read** which reads the response of the server byte by byte using the command **in.read()** in the while loop. This while loop goes until the end of the response header. To identify the end of the file header, I have used a bunch of nested if else statements, so whenever consecutive **\r\n\r\n** is encountered, the **break** statement is encountered and the program comes out of the while loop. If the end of the header is not reached, the while loop keeps on going on and prints the bytes read from the response**. \r** and **\n** do not mean the end of the header file, **\r** means carriage return and **\n** means a new line and **\r\n** means a new line. Inside the print statement in each of the if statements, the ASCII code of the byte is converted into its corresponding character before printing. The while loop stops when the end of the header is encountered which is **\r\n\r\n** not anything else. Basically we print the part until response header ends and after that we put a loop to copy the contents of the data that is actually to be store on our machines.

After this I have created a buffered outputstream object called **out** to write the contents (file body) from the input stream to a new file on the local system. I have taken the **byte_read_out** variable to read bytes using the buffer. A while loop is set until the end of file is reached. The chunks of the contents are read from the input stream and written into the output stream. We use **out.flush()** inside the while loop to clear the contents of **out.** the size of each chunk is taken as 4096 for its maximum efficiency.

All of the streams and sockets are closed after this. The exception handling is done at the end which handles the exceptions such as UnknownHostException and IOException.