

# CS5001 - Object Oriented Modelling Design & Programming

## Practical3 – Networking Report

(Student ID- 190026870)

### Abstract:

This report covers the advanced requirements of the third practical on networking. Apart from the basic requirements given

- The main method for your server should take two command-line arguments, the directory from which your server will serve documents to clients and the port on which your server should listen.
- The server should support and respond correctly to HEAD requests.
- The server should support and respond correctly to GET requests.
- The server must be able to return HTML documents requested by a client.
- The server should respond with appropriate error messages when non-existent services or resources
- are requested

Three advanced requirements are attempted in this practical which are displaying the binary images, multi-threading and logging.

### Displaying binary images:

The logic to display binary images lies in the ConnectionHandler. Java class where content type for the image files is checked using getMimeType() method. If the method is GET, then the images are returned to the client. This is tested manually by accessing localhost:12345/beer.jpg and localhost:12345/tp\_it.jpg using browser as the client.

After completing the basic requirement this did not work as I'm having line breaks between the content and the images because of using (“\r\n”) and initially getMimeType( ) method supported only text and html files but later on for the advanced requirement I have included some other image formats such as jpg/jpeg, png, gif.

### Multithreading:

To implement the concept of multithreading a class Threading.java which extends Thread class is created and run () method is overridden. This class is instantiated in Server.java. Then,

ConnectionHandler.java is instantiated in the run () method. For testing this manually two different browsers are used as clients and the connection request is given at the same time. To identify some timing gap between these clients Thread.sleep(-) method is invoked.

## **Logging:**

To store logging details log.txt file is created. This concept is implemented using various class in java.util.logging package. For storing the logging details in a file FileHandler class is used. Info () method is used to store the data. This logging data can also be seen on the terminal for that ConsoleHandler class is used. Once the request is given to the server from a client all these details are stored in the log.txt file and also displayed on the terminal.