# CA3 – 35% Socket Programming

**Objective**: To gain practical experience in developing a simple java client using sockets.

**Environment**: You may use any IDE that you would like e.g., NetBeans, IntelliJ, Atom, Notepad etc

Preferred IDE Netbeans

**Instructions**:

Based on the last digit of your student number, you must develop an application that achieves the given task. The application developed must use java sockets in its implementation. You are free to decided on whether you required one class or more for the implementation (instantiable classes). You must adhere to java code best practices such as indentation, variable naming conventions, commenting etc.

|  |  |
| --- | --- |
| Last digit of student number | Task |
| 0-3 | Write a Java application that connects to the NIST Internet Time Server **(`time.nist.gov`)** to fetch the current date and time. The application should attempt the connection multiple times (e.g., 3 times), handle possible connection issues with retries, and log each attempt with the response status. If successful, print the current date and time. If unsuccessful after all attempts, display a failure message to the user.  TIP: You will need to handle potential connection errors gracefully, as the time server may not always be accessible. Implement multiple connection attempts with a delay between each and log each attempt. After three unsuccessful attempts, display a message informing the user that the server is currently unavailable. |
| 4-6 | Write a Java application that connects to the **(‘whois.internic.net’)** server on port 43 to retrieve "WHOIS" information for a specified domain name. The domain name should be passed as a parameter to the application. If no parameter is provided, default to querying ncirl.ie. Include error handling to manage cases when the server cannot be reached, or the domain information is unavailable.  TIP: You will need to pass the domain name to be looked up as a parameter to the application. As an alternative, you may statically define that the request performs a lookup for ncirl.ie. |
| 7-9 | Write a Java application that sends an HTTP HEAD request to a specified web server URL. The server's response should be displayed to the user, confirming the connection was successful. The URL should be passed as a parameter, but if no URL is specified, the application should default to **`http://nginx.org/`.** Handle connection timeouts and invalid URLs and log the status of the connection attempt.  TIP: You will need to pass the URL to send the HEAD request to, as a parameter when running the application. As an alternative, you may statically define that the request to one of the URL’s below.  Sample Sites:   * <http://httpforever.com/> * <http://nginx.org/> |

**Marking:**

You will receive marks for all aspects of your code, not just on whether you have a functioning application or not. You must adhere to programming best practices such as appropriate indentation, naming conventions, commenting etc. You will be awarded marks for the appropriate use of these best practices in addition to being awarded marks on the actual code submitted.

**Plagiarism**

As per NCIRL plagiarism policy, all work must be your own. Anyone suspected of plagiarising content from other sources will receive a zero mark for this lab.

**Submission:**

You may submit your completed code via the CA3 link available via Moodle.

**Deadline:**

As per listed in the upload link on Moodle.