COMPUTER NETWORK - Homework Assignment 1

IRC Robot - Due Date : 23:59, November 12, 2016

1. Description

IRC is an application layer protocol that facilitates communication in the form of text. The chat process works on a client/server networking model.

This assignment is based on the knowledge of "Socket programming".
You need to "Connect" to socket and "Send / Receive" message to/from socket.

2. Directory Structure

Your program has to read **"config"** to get the information of IRC channel.

[Folder]	[config]
- config # IRC Configuration - main # Main program	CHAN='**#CN_Demo**' CHAN_KEY='**ILoveTA**'

3. Grading Policy

1. **Implementation (80%)**
   - (a) Connection to Channel (20%)
   - (b) Automatic Introduction Message & 'Repeat' Message (20%)
   - (c) Calculator (10% + 5%)
   - (d) Game - Guess Number (20%)
   - (e) Help (5%)
2. **Report (10%)**
3. **Demo (10%)**
4. **Bonus - HTML Parser (10%)**

4. How to Submit the Assignment ?

Please **compress** all of your file into an archive. (Format : zip/rar)
Then email to ntu.cnta@gmail.com before due date.
Penalty for late submission is **"20% per day"**.

5. What Should Your Robot Do ? (Language : c/c++/python)

(a) Connection to Channel (20%) :

Using "Socket" API to connect to the IRC channel.

Once connection is successful, robot's name will be shown on the channel.

(b) Automatic Introduction Message & 'Repeat' Message (20%) : "@repeat"

Automatically send introduction message when robot enters the channel.

```
15:40 -!- yylou [~yylou@140.112.28.135] has joined #CN_Demo
15:40 [Users #CN_Demo]
15:40 [@Ubuntu_Yylou] [ yylou]
15:40 -!- Irssi: #CN_Demo: Total of 2 nicks [1 ops, 0 halfops, 0 voices, 1 normal]
15:40 -!- Channel #CN_Demo created Sat Oct  8 03:47:58 2016
15:40 -!- Irssi: Join to #CN_Demo was synced in 4 secs
15:40 -!- TA_Yylou [~TA_Yylou@host-219-70-186-230.dynamic.kbtelecom.net] has joined #CN_Demo
15:40 < TA_Yylou>  Hi, I'm TA_Yylou
```

```
15:43 < yylou> @repeat Teacher'  'is beautiful
15:43 < TA_Yylou> Teacher'  'is beautiful (yylou)
```

(c) Calculator (10% + 5%) : "@cal" (*NOTE: We won't test overflow situation)

Operator : +, -, *, /, (), ^(Power) ; Operand : Positive Integer/Floating point/0

➔ Implement "Exceptions Handling" with wrong format of equation.

```
15:46 < yylou> @cal ((1.1+1.1)-(2*2))/3^3
15:46 < TA_Yylou> -0.0666666666667 (yylou)
15:49 < yylou> @cal ((1.1+1.1+a)-(2*2))/3^3
15:49 < TA_Yylou> ERROR - Please check input (yylou)
```

(d) Game - Guess Number (20%) : "@play", "@guess"

Guess number from 1 to 100 in 5 tries. Robot needs to provide the hints.

Robot only plays with a single user, so specify robot's name to start the game.

NOTE : Other functions should still be provided during the game.

```
16:08 < yylou> @play TA_Yylou
16:08 < TA_Yylou> Start ! (0-100 with 5 times) (yylou)
16:08 < yylou> @repeat Can you repeat during the game ?
16:08 < TA_Yylou> Can you repeat during the game ? (yylou)
16:08 < yylou> @cal ((1.1+1.1)-(2*2))/3^3
16:08 < TA_Yylou> -0.0666666666667 (yylou)
16:08 < yylou> @guess 50
16:08 < TA_Yylou> Higher (4) (yylou)
16:08 < yylou> @guess 75
16:08 < TA_Yylou> Higher (3) (yylou)
16:08 < yylou> @guess 94
16:08 < TA_Yylou> CORRECT (2) (yylou)
```

(e) Help (5%) :

```
15:52 < TA_Yylou> @repeat <String> (yylou)
15:52 < TA_Yylou> @cal <Expression> (yylou)
15:52 < TA_Yylou> @play <Robot Name> (yylou)
15:52 < TA_Yylou> @guess <Integer> (yylou)
15:52 < TA_Yylou> @youtube <Song> (yylou)
```