UECM3033 Numerical Methods (Assignment #1)

DEADLINES:

- This assignment will contribute to 5 marks of the coursework.
- Assignment submission: 29/1/2016 (Friday) by 12:00 noon.
- Peer Reviews: 5/2/2016 (Friday) by 12:00 noon.

All deadlines are final and no extensions are allowed. Delay in submission of the assigments will delays the peer review process. Thus a penalty of 1 mark per day will be imposed for one day late in submission of assignment. Each student will need to review two peers. No mark will be awarded if a student failed to review the two peers.

You could always edit and change your submission as many time as you like. However, you are not suppose to change your submission when the peer reviews starts. Your reviewers will be asked to check the last modified dates of your files. If the dates is beyond the stipulated due dates, you will be considered as late submission and same penalty will be applied.

There are a total of 3 tasks for this assignments:

- setup a github repository
- setup Python
- modify and run a given Python script.

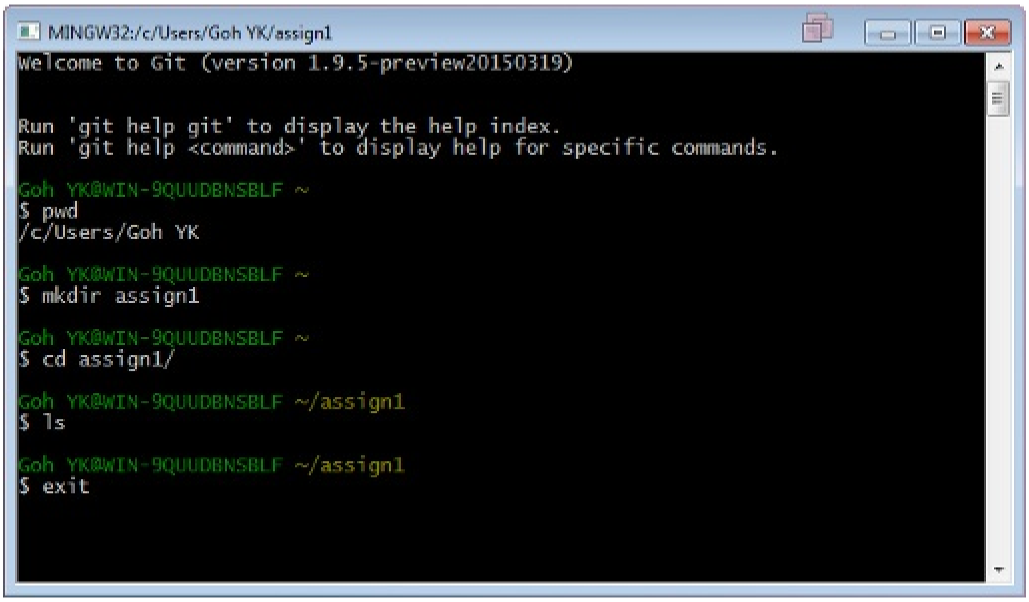
Task 1 -- setup a github repository

Version control is a system that tracks changes of files and source codes so that you could recover or recall a specific version in the future. It is especially important if sharing and editing files across a team.

Git is a popular opensource version control system that is widely used by developers. Many of opensource software is managed by git version control. (Git at Wikipedia.)

One of the useful features of git is that all files could be stored in the local repositories in your computer. Git is a command line based software, and thus there will be a bit of learning curve to overcome. On the other hand the installation of git is straight forward. Just point your browser to http://git-scm.com/download/ and download Git Bash for your operating system. Depending on your operating system, the procedure to install git will be a bit different, please consult the documentation. (If you are using Linux, usually git is already installed.)

The following is a screen shot of Git Bash session. Several basic folder navigation commands are as below:



```
Welcome to Git (version 1.9.5-preview20150319)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Goh YK@WIN-9QUUDBNSBLF ~
$ pwd
/c/Users/Goh YK

Goh YK@WIN-9QUUDBNSBLF ~
$ mkdir assign1

Goh YK@WIN-9QUUDBNSBLF ~
$ cd assign1/

Goh YK@WIN-9QUUDBNSBLF ~/assign1
$ ls

Goh YK@WIN-9QUUDBNSBLF ~/assign1
$ exit
```

- *pwd* - print current working directory.
- *mkdir* - make directory
- *cd* - change directory
- *ls* - list the content of current directory
- *exit* - exit Git Bash

To begin using git we navigate into the folder we would like to use as local repository, say *assign1*. Then, we can initialize the repository by

```
$ git init
```

Once the repository is created, user could create or modify files in the repository. When a certain milestone is reach and the user decide all the changes made should be updated in the repository, then the user should *commit* the changes in the repository. Each commit to git repository will be tagged with the username of the person who made the commit and a commit message that give an indication of what were changed. Hence, we will need to configure git first before we can make any commit.

```
$ git config --global user.name "Your Name Here"
$ git config --global user.email "your_email@1utar.edu.my"
```

You will only need to do this once in your computer. Of course you could reconfigure git again when you wish to change your details. You can also check the configuration for git by the following command:

```
$ git config --list
```

It won't be too useful If git is only to store files locally. Usually we would like sync the local repositories with the cloud based git server for backup or sharing purposes. Github is a web-based hosting service that provide git version control. User is allowed to "*push*" or "*pull*" their local repositories to and from remote repositories.

First thing first, let sign up a Github account.

- Go to Github homepage https://github.com/join.
- Enter a username, email and password and click "Sign up for Github".
- On the next screen, select the **free plan** and click "Finish Sign Up".

To complete **Task 1**:

- Login to your Github.
- While still stay logged in Github, go to https://github.com/yongkheng/UECM3033_assign1
- Click "fork" button at the top right hand corner. Now your Github will have a copy of the UECM3033_assign1 repository. Copy the URL.
- Go to the google form http://goo.gl/forms/vfTbe6w89K.
- Fill in your name, ID and the URL to your newly forked repository.
- Answer the question in the google form and submit.

Task 2 -- setup python

This part of the assignment consists of two parts:

1. Install and take a screen capture of your Python.
2. Save your screen shot onto Github repository.

For the installation of Python, you are encouraged to download and install Anaconda, a completely free Python distribution from Continuum Analytic.

1. Go to http://continuum.io/downloads
2. Choose "Python 3.5" suitable to your operating system.
3. Download and install.
4. Start Spyder, the IDE (Integrated Development Environment) of Anaconda distribution of Python.
5. At the Python shell, type the following command and take a screen capture of the entire Spyder IDE, and call it "`python.png`".

```
import sys
sys.version
import this
```

If you prefer other distribution of Python other than Anaconda, you could also take a screen capture of the Python of your choice with the 3 lines of code above. However, the version of the Python used, must be 3.1 and above.

Next is to merge your file `python.png` with your Github repository.

1. Clone the repository

```
$ git clone https://github.com/your_github_id/UECM3033_assign1
```

This will create a UECM3033_assign1 folder. (Remember you can check your current directory with ''pwd```* and *```ls```*.) Change into the

*UECM3033_assign1` folder.

2. Copy and replace your `python.png` into *`UECM3033_assign1`* folder.

3. Commit changes and push to Github.

```
$ git add -A
$ git commit -m "Added python.png"
$ git push
```

4. Verify at Github website that "python.png" is added into the repo.

Task 3 -- modify Python script and write Markdown report

In the repository that you just cloned, there is a `pyexercise.py` file and a `report.md` file. The `pyexercise.py` is a Python script that you will need to modify according to the instruction below. While the markdown file `report.md` is the template for you to report your assignment.

In `pyexercise.py`, there are 3 functions: `fun_1`, `my_integral` and `my_solution`.

1. Open the file and edit `pyexercise.py` to calculate the hexadecimal representation of your student ID.
2. Next, choose an definite integration that you think is interesting. The integration should not be simple polynomials or trigonometry functions that could be solved by hand. Even better if the integration has no simple close form solution.
3. Modify `my_integral` such that it calculate the integration of your choice. The example shown in the original `pyexercise.py` is to evaluate (see README.pdf if the mathematics equation doesn't render properly.) These integrals will be used as testing cases in your assignment 3.
4. Next, setup a system of 10 linear equations with unique solutions.
5. Modify `my_solution` such that it calculate the solution of the system of linear equations that you setup. The example in the original file is solving
6. Record all your findings into the `report.md` file.
7. Upload into Github together with your modified python script

You noticed that I have mentioned that to report your findings into `report.md` file. It is called a **Markdown** file. Markdown allows you to write using simple plain text format, which then Github can translates it into common webpage HTML. Markdown is especially useful for writing reports with codes, it save the hassle of formatting. More information on markdown language can be found here https://help.github.com/articles/markdown-basics/.

The `report.md` file you downloaded can be edited using any text editor like `notepad.exe`. Try to open `report.md` using a text editor and modify the file. Upload into Github and you can see the corresponding changes will appear in the preview page of `report.md`.

Remember that you could edit and change your `report.md` as many time as you like. However, you are not suppose to change anything when the peer reviews starts. Your reviewers will be asked to check the last modified dates of your files. If the dates is beyond the stipulated due dates, you will be considered as late submission and penalty will be applied.

Summary:

Task 1:

- install Git Bash
- create Github account
- fork UECM3033_assign1 repository
- submit URL of forked repository

Task 2:

- install Python, capture screen shot
- clone UECM3033_assign1 repository
- pull repository
- submit screen shot "python.png" to Github

Task 3:

- Edit and modify `pyexercise.py`
- Edit and modify `report.md`
- Submit `pyexercise.py` and `report.md` to Github.

---

last modified: 12 Dec 2015