Problems 2020

Programming and Scripting

Due: last commit on or before April 3rd, 2020

This document contains the instructions for Problems 2020 for Programming and Scripting. The lecturer will indicate which problems you should solve on different weeks. Your solutions will be assessed towards the end of the semester and will be worth 50% of your marks for this module.

The marking scheme is given below, and please note that one of the categories is for a pragmatic attitude to work. This involves working on the exercises throughout the semester. It is not expected that you get every program right first time. So long as an attempt is made when indicated by the lecturer, this will count as a good approach. It is important that you keep working on any incomplete problems until the deadline.

Please note that all students are bound by the Quality Framework [3] at GMIT which includes the Code of Student Conduct and the Policy on Plagiarism. The onus is on the student to ensure they do not, even inadvertently, break the rules. A clean and comprehensive git [1] history (see below) is the best way to demonstrate that your submission is your own work. It is, however, expected that you draw on works that are not your own and you should systematically reference those works to enhance your submission.

How to submit

During the semester you will learn how to use the version control software git [1] to track your work. You will sync your work with GitHub [2] (you may use another provider if you wish) and provide the URL to your GitHub repository using the submission form on the Moodle page. You should consult the Moodle page for information on how to do this.

I suggest you consider making your repository publicly available (the default) so that prospective employers may view it. Should you wish to, you may make it private so long as you make it available to the lecturer. You can submit your URL at any time as git will track what work was done before the deadline and what work was completed afterwards. In general, only work completed before the deadline will count. However, in borderline cases or exceptional circumstances work done after the deadline may be considered, with or without penalty.

Marking scheme

This problem set will be worth 50% of your marks for this module. The following marking scheme will be used to mark your submission out of 100%. The examiner's overall impression of the assignment may influence marks in each individual component.

25%	**Research**	Investigation of each problem and its solution, as evidenced by clean and well-commented code.
25%	**Development**	Clear, well-written, and efficient code.
25%	**Consistency**	Good planning and pragmatic attitude to work as evidenced by commit history.
25%	**Documentation**	Concise descriptions of solutions in comments and README.

Advice for students

- You must be able to explain your assignment during and after its completion. If you had trouble understanding something in the first place, you will likely have trouble explaining it a couple of weeks later. Write a short explanation of it into your submission, so that you can jog your memory later.
- Everyone is susceptible to procrastination and disorganisation. You are expected to be aware of this and take reasonable measures to avoid them.
- Students have problems from time to time. Some of these are unavoidable, such as acute family issues or illness. In such cases allowances regarding assignments can sometimes be made. Students should be able to show that up until an issue arose they had completed a reasonable and proportionate amount of work and took reasonable steps to avoid preventable issues.
- Go easy on yourself — this is one assignment in one module. It will not define you or your life. A higher overall course mark should not be determined by a single assignment.

Questions

For some of the following questions, examples of running the program are given in grey. Lines beginning with a dollar sign are typed at the command line, and the other lines are part of the programs.

1. Create a GitHub account and create a new repository in GitHub to contain the programs you will write for this problem sheet. Add a README, gitignore, and a license to your repository. Then add a program that just prints "Hello, world!" to the screen when run. Finally, submit the GitHub repository using the form on the Moodle page.

```
$ python hello-world.py
Hello, world!
```

2. Write a program that outputs whether or not today is a weekday. An example of running this program on a Thursday is as follows.

```
$ python weekday.py
Yes, unfortunately today is a weekday.
```

An example of running it on a Saturday is as follows.

```
$ python weekday.py
It is the weekend, yay!
```

3. Write a program that asks the user to input any positive integer and outputs the successive values of the following calculation. At each step calculate the next value by taking the current value and, if it is even, divide it by two, but if it is odd, multiply it by three and add one. Have the program end if the current value is one.

```
$ python collatz.py
Please enter a positive integer: 10
10 5 16 8 4 2 1
```

4. Write a program that takes asks a user to input a string and outputs every second letter in reverse order.

```
$ python secondstring.py
Please enter a sentence: The quick brown fox jumps over the lazy dog.
.o zletrv pu o wr cu h
```

5. Write a program that that takes a positive floating-point number as input and outputs an approximation of its square root.

```
$ python squareroot.py
Please enter a positive number: 14.5
The square root of 14.5 is approx. 3.8.
```

6. Write a program that outputs today's date and time in the format "Monday, January 10th, 2019 at 1:15pm".

```
$ python whenisit.py
Monday, January 10th, 2019 at 1:15pm
```

7. Write a program that reads in a text file and outputs the number of e's it contains. The program should take the filename from an argument on the command line.

```
$ python es.py moby-dick.txt
116960
```

8. Write a program that displays a plot of the functions $f(x) = x$, $g(x) = x^2$ and $h(x) = x^3$ in the range $[0, 4]$ on the one set of axes.

References

[1] Software Freedom Conservancy. Git.
`https://git-scm.com/`.

[2] Inc. GitHub. Github.
`https://github.com/`.

[3] GMIT. Quality assurance framework.
`https://www.gmit.ie/general/quality-assurance-framework`.