Scripting Languages: Workshop 5

- To complete these tasks you will need to place the following two (2) files into your current working directory:
    - `sampledata.txt`
    - `salesdata.csv`
- Ensure that all of the scripts you write are fully commented to clearly explain their functionality at every step

Task 1

1. Write a script named `ws5a.sh` that outputs to the terminal a count of the lines in `sampledata.txt` that **start** with the uppercase string `GET`
2. If you encounter an error, read the error message printed to the terminal carefully and attempt to resolve the issue and run the `fws5a.sh` script again or ask your tutor for assistance

Task 2

1. Write a script named `ws5b.sh` that outputs to the terminal those lines in `sampledata.txt` that end with the code `573`
2. If you encounter an error, read the error message printed to the terminal carefully and attempt to resolve the issue and run the `ws5b.sh` script again or ask your tutor for assistance

Task 3

1. Write a script named `ws5c.sh` that outputs to the terminal those lines in `sampledata.txt` that contain the file extension `jsp` or `py` or `asp` or `aspx`
2. If you encounter an error, read the error message printed to the terminal carefully and attempt to resolve the issue and run the `ws5c.sh` script again or ask your tutor for assistance

Task 4

1. Write a script named `ws5d.sh` that outputs to the terminal a count of all the lines in `sampledata.txt` that contain HTTP error `404` **and** that also end in code `506`
2. If you encounter an error, read the error message printed to the terminal carefully and attempt to resolve the issue and run the `ws5d.sh` script again or ask your tutor for assistance

Task 5

1. Write a script named `ws5e.sh` that retrieves all lines in `sampledata.txt` that end in the IP address `http://192.168.5.162/` and write these lines to a text file named `162attempts.txt`
2. If you encounter an error, read the error message printed to the terminal carefully and attempt to resolve the issue and run the `ws5e.sh` script again or ask your tutor for assistance

Task 6

1. Write a script named `ws5f.sh` that retrieves all lines in `sampledata.txt` that contain HTTP error code 404 **and** also end in code 506 **and** write these lines to a text file named `404messages.txt`
2. If you encounter an error, read the error message printed to the terminal carefully and attempt to resolve the issue and run the `ws5f.sh` script again or ask your tutor for assistance

Task 7

1. Write a script named `ws5g.sh` that outputs to the terminal a count of the lines in `sampledata.txt` that do **not** contain a *three digit code* that *starts* with `50`, i.e. *50x*
2. If you encounter an error, read the error message printed to the terminal carefully and attempt to resolve the issue and run the `ws5g.sh` script again or ask your tutor for assistance

Task 8

1. Write a script named `ws5h.sh` that checks that the input provided by the user is an integer and nothing else
2. Prompt the user for an integer of any length using a standard input prompt
3. Once the input has been provided, check that it is an integer; if it is, print to the screen a message that says `Congratulations, you have enter a valid integer` then exit the script, else print `Sorry, the value you have entered is not a valid integer`, then exit the script
4. Test your script by inputting both integer and non-integer values
5. If you encounter an error, read the error message printed to the terminal carefully and attempt to resolve the issue and run the `ws5h.sh` script again or ask your tutor for assistance

Task 9

1. Write a script named `ws5i.sh` that calculates the **total** sales for those products in *salesdata.csv* with a product code that **ends** in *XS* and display this to the terminal as an accurate currency value:

```
Total sales for product codes ending in XS is $944.95
```

2. Don't bother prompting the user for the file name *salesdata.csv*, just assign it to a variable
3. Keep your script as short as possible by using the *command substitution*, *piping* and *redirection* techniques demonstrated in the lecture
4. **TIP**: The *tr*, *cut* and *tail* commands may prove very useful in this script, so look these up on *SS64.com* or *gnu.org* if you are not yet familiar with them