[Index | Exercise 1.3 | Exercise 1.5]

Exercise 1.4

Objectives:

- Review of how to define simple functions
- Exception handling

Files Created: None

Files Modified: `pcost.py`

(a) Defining a function

Take the program `pcost.py` that you wrote in the last exercise and convert it into a function `portfolio_cost(filename)` that takes a filename as input, reads the portfolio data in that file, and returns the total cost of the portfolio as a floating point number. Once you written the function, have your program call the function by simply adding this statement at the end:

```
print(portfolio_cost('Data/portfolio.dat'))
```

Run your program and make sure it produces the same output as before.

(b) Adding Error Handling

When writing programs that process data, it is common to encounter errors related to bad data (malformed, missing fields, etc.). Modify your `pcost.py` program to read the data file `Data/portfolio3.dat` and run it (hint: it should crash).

Modify your function slightly so that it is able to recover from lines with bad data. For example, the conversion functions `int()` and `float()` raise a `ValueError` exception if they can't convert the input. Use `try` and `except` to catch and print a warning message about lines that can't be parsed. For example:

```
Couldn't parse: 'C - 53.08\n'
Reason: invalid literal for int() with base 10: '-'
Couldn't parse: 'DIS - 34.20\n'
Reason: invalid literal for int() with base 10: '-'
...
```

Try running your program on the `Data/portfolio3.dat` file again. It should run successfully despite printed warning messages.

(c) Interactive Experimentation

Run your `pcost.py` program and call the `portfolio_cost()` function directly from the interactive interpreter.

```
>>> portfolio_cost('Data/portfolio.dat')
44671.15
>>> portfolio_cost('Data/portfolio2.dat')
19908.75
>>>
```

Note: To do this, you might have to run python using the `-i` option. For example:

```
bash % python3 -i pcost.py
```

We are going to be writing a lot of programs where you define functions and experiment interactively. Make sure you know how to do this.

[Solution | Index | Exercise 1.3 | Exercise 1.5]
