Exercises Loops and Exceptions

Exercise 1

1. using the file `sum_of_squares.py` from the previous exercise as a starting point (included here).
2. Create a file named `squares_calculator.py` in this folder.
3. In that file create a function named `calculate_sum_of_squares` that takes in a number as a parameter and returns the sum of squares of the integers from 1 to that number.
4. In the `sum_of_squares.py` file import function `calculate_sum_of_squares` from the `squares_calculator.py` file.
5. use the `calculate_sum_of_squares` function to calculate the sum of squares of the integers from 1 to `my_square`, where `my_square` is a variable that is input by the user. E.g. user enters 4 then return 1x1 + 2x2 + 3x3 + 4x4 = 30
6. import that function into the `sum_of_squares.py` file and use it to calculate the sum of squares, and remove the existing code to use this `calculate_sum_of_squares` function instead.
7. The output should look like the following:

```
$ python sum_of_squares.py
Enter a number to sum the squares: 4
The sum of squares is 30
```

Exercise 2

1. using the file `price_is_right.py` from the previous exercise as a starting point (included in the folder.)
2. Create a new file named `price_is_right_games.py` in this folder.
3. In that file create a function named `guess_items_price` that has one parameter named `guess` (this will be a number). If the guess is in the list of prices then return `You win!`, otherwise return `Sorry you lose!`
4. In that function use the `random_prices` from the function and include it in the `guess_items_price` function.
5. Import the function `guess_items_price` into the `price_is_right.py` file and use and continuous loop to play. The user should be able to enter `n` to quit the game.
6. The output should look like the following:

```
$ python price_is_right.py
Welcome to the Price is Right!
Guess the price of one of the items (done): 1
The prices were: [3, 3, 3]
Sorry you lose!
Do you want to play again? (y/n): y
Guess the price of one of the items (done): 2
The prices were: [8, 8, 2]
```

```
You win!
Do you want to play again? (y/n): n
```

Exercise 3

In this exercise, you're going to create a game where the user has to guess the price of a car within $1000.

1. Create a file named `car_price_guessing_game.py` in this folder. In this file create a loop that will ask the user to guess the price of the car. The user should be able to enter `n` to quit the game.
2. In the file `price_is_right_games.py` create a function named `guess_car_price` that has one parameter named `guess` (this will be a number).

- If the guess is the same as the actual price then return `You win! That's exactly the price! You're a cheater!`
- If the guess is within $1000 of the actual price then return `You win!`
- If the guess is more than $1000 but less than $5000 of the actual price then return `You're close!`
- If the guess is more than $5000 of the actual price then return `Way off!`

3. Import the function `guess_car_price` into the `car_price_guessing_game.py` file and use it to play the game and print the result of the function.
4. The output should look like the following:

```
$ python car_price_guessing_game.py
Welcome to the car price guessing game!
Guess the price of the car: 10000
The price of the car was: 25000
Sorry you lose!
Do you want to play again? (y/n): y
Guess the price of the car: 22000
The price of the car was: 25000
You're close!
Do you want to play again? (y/n): y
Guess the price of the car: 24500
The price of the car was: 25000
You win!
Do you want to play again? (y/n): y
Guess the price of the car: 25000
The price of the car was: 25000
You win! That's exactly the price! You're a cheater!
Do you want to play again? (y/n): n
```

Exercise 4

1. Create a file called `golf.py` in this folder.
2. In that file continue to ask for the user to enter the scores for their games. Until the enter `q` to quit and calculate the score. Save these scores to a list.
3. Create a function called `calculate_average` scores in a file named `score_calculator.py`, that's going to take a nonspecific number of grades as parameters (using `*args`).

4. Using this function calculate average score for the user by importing into your function in the `golf.py` file.
5. the output should look like the following:

```
$ python golf.py
Golf Score Calculator
Enter another score (q to quit and calculate)?: 98
Enter another score (q to quit and calculate)?: 96
Enter another score (q to quit and calculate)?: 90
Enter another score (q to quit and calculate)?: 85
Enter another score (q to quit and calculate)?: q
Your average golf score is 92.25
```

6. Create a function in your `score_calculator.py` file called `calculate_handicap` that takes a nonspecific number of grades as parameters (using `*args`). It's going to calculate the handicap by taking the average of the best 5 scores and then subtracting `72` from that average. O
7. import that function in your `golf.py` file and use it to calculate the handicap under the average score.
8. The output should look like the following:

- if less than 5 scores are entered:

```
$ python golf.py
Golf Score Calculator
Enter another score (q to quit and calculate)?: 90
Enter another score (q to quit and calculate)?: 95
Enter another score (q to quit and calculate)?: q
Your average golf score is 92.5
Your handicap is: Need at least 5 scores to calculate a handicap
```

- if 5 or more scores are entered:

```
$ python golf.py
Golf Score Calculator
Enter another score (q to quit and calculate)?: 98
Enter another score (q to quit and calculate)?: 95
Enter another score (q to quit and calculate)?: 94
Enter another score (q to quit and calculate)?: 94
Enter another score (q to quit and calculate)?: 92
Enter another score (q to quit and calculate)?: 100
Enter another score (q to quit and calculate)?: 120
Enter another score (q to quit and calculate)?: 130
Enter another score (q to quit and calculate)?: q
Your average golf score is 102.875
Your handicap is: 22.599999999999994
```

Note: Try to figure out how to round the handicap to 2 decimal places!