

Instructor	Mike Mulder (mmulder10@bcit.ca)
Total Marks	15
Due Dates	Part A – End of Class Part B – Sunday, June 6 th at midnight

PART A – In-Class (Due End of Class), 5 marks

1. Create a Python Project with the name Lab6
2. Create a Python file called lab6.py
3. In file lab6.py Create function main in the format:

```
def main():  
    pass  
  
if __name__ == "__main__":  
    main()
```

Create the following functions:

- Function `weight_converter()`, the function converts weight in kilos to weight in pounds, the function displays a table of the weight from 30 kilos to 100 kilos in increments of 10 and displays the corresponding weight in pound up to two digits after the decimal point for each value.

Hint: 1 Kg = 2.20 pounds.

Here is an example of the expected output

```
weight in Kilos: 30 is 66.00 in Pounds
weight in Kilos: 40 is 88.00 in Pounds
weight in Kilos: 50 is 110.00 in Pounds
weight in Kilos: 60 is 132.00 in Pounds
weight in Kilos: 70 is 154.00 in Pounds
weight in Kilos: 80 is 176.00 in Pounds
weight in Kilos: 90 is 198.00 in Pounds
weight in Kilos: 100 is 220.00 in Pounds
```

Process finished with exit code 0

- Function `get_divisible()`, the function prompts the user to enter three numbers, first number, second number and divisor.
 - If the first number is less than the second number then the function will display all the numbers between first and second number inclusive that are divisible by the given divisor.
 - If first number was greater than second number then the function will display all the numbers between the first and the second number that are divisible by the given divisor in a descending order.
 - The divisor cannot be 0. If the divisor was 0 the message “ the divisor cannot be 0” will be displayed and the function will terminate.
- Function `get_list_stats()`, the function prompts the user to enter positive integers, space to end the input . If the user entered anything other than an integer it will be ignored. If the user entered and integer it will be added to a list.

The function displays the list of integers, the length of the list, number of even numbers, min and max numbers. **Do not use min and max functions.**

Note that 0 is a positive number.

- In the main function call methods `weight_converter`, `get_divisible` and `get_list_stats`

Demo to your instructor on Slack (a screenshot of your output) the output form lab6.py. Upload the screenshot to the Lab 6 dropbox on the Learning Hub as well.

Part B – Take Home (due June 6th at midnight), 10 marks

In file lab6.py add the following functions:

- Function `calculate_pay()`, the function takes two parameters number of employees and hourly rate. The function creates a list of lists. Each inner list consists of the number worked hours and the pay of each employee. The size of the list should be the same as the employees' number. The function will prompt the user to enter the number of worked hours for each employee, calculates the pay of the employee, creates a list of the number of hours and the pay and adds it to the employee list. The pay is calculated as follows.
 - If number of hours is less than 40, then the pay is the number of hours times the hourly rate.
 - If the number of hours is greater than 40, then the pay is the hourly rate is 1.5 for the hours over 40 hours.

The function displays the number of hours and the pay of each employee in the given format:

If the hourly rate is 10 and the number of employees was 4 the output of the function would be:

```

C:/Users/Rana/PycharmProjects/session#6Lab/lab#6.py
input the number of worked hours:60
input the number of worked hours:20
input the number of worked hours:40
input the number of worked hours:30
the employee worked for 60 hours and earned 700.00 $
the employee worked for 20 hours and earned 200.00 $
the employee worked for 40 hours and earned 400.00 $
the employee worked for 30 hours and earned 300.00 $

```

- Function `is_prime()`, the function takes one parameter which is a positive number and returns `True` if the number was prime, `False` otherwise
- Function `get_prime_numbers()`, the function takes one argument which is a positive integer greater than 2, the function generates a list of integers from 2 to the given number inclusive. The function then iterates through the list and displays all the prime numbers in the list. It should use your `is_prime` function.

Sample output of the function:

If the passed number was 20, the output would be:

```

C:/Users/Rana/PycharmProjects/session#6Lab/lab#6.py
[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
the number 2 is prime number
the number 3 is prime number
the number 5 is prime number
the number 7 is prime number
the number 11 is prime number
the number 13 is prime number
the number 17 is prime number
the number 19 is prime number

```

- In function `main()`:
 - Prompt the user to input number of employees and hourly rate, pass the values to function `calculate_pay()`
 - Prompt the user to enter a positive integer greater than 2 and pass it to the function `get_prime_numbers()`

Best Practices

- Variable names are descriptive and should be lower_snake_case
- function names should be lower_snake_case
- All functions include Doc-String comments

Part B Submission

Submit a zipfile with you lab6.py file to the Lab6 dropbox on the Learning Hub before June 6th at midnight.