PROGRAMMING FOR ALL – SPRING 2021 - WEEK #1 – 210329 & 210331

Topics for the week:

- Introduction to the class – get familiar with CANVAS, ZOOM

INTRODUCTION: **read chapter 1**

- Introduction to computers and programming
- Python 3.7 - Idle – interactive & script mode

PROGRAMMING START: **read chapter 2**

- Introduction of print function and its use
- Introduction of variables, type of variables, variable names and styles
- Printing text and values of variables
- Basic math calculations
- Formatting, rounding
- Input from the keyboard

CLASS EXERCISES:

Calculations

- **Daily production**

A company to produce one face mask needs 7 seconds.
Write a program that calculates and displays a daily production of masks produced in this company.
(Assume that the company works 24 hours each day.)

- **Total number of books**

A bookstore company has 17 bookstores in a city. In one of these stores are 50,000 of books, and it represents 14% of all books in all stores.
Write a program that calculates and displays the number of books in all bookstores of this company.

- **Average speed**

Tony left his/her house 11:45am and drove to his/her parent's house which is distanced 168 miles from Tony's house. Tony arrived at 2:15pm.
Write a program that calculates and displays:

- Tony's driving time in hours and minutes;
- Tony's average speed (in miles per hour) when driving.

Calculations with inputs

- **Average of two**

Write a program that asks a user to enter two numbers and displays an average value of these numbers.

- **Secret word**

Write a program that asks a user to provide his/her secret word. Then the program displays the secret form in the following form:

Your secret word is "……."; it is not a secret anymore!"

- **From kilograms to pounds**

Write a program that asks a user to provide a weight in kilograms and displays the weight in lb.

Note: 1 Pound (lb) is equal to 0.45359237 kilogram (kg).

Pounds=kilograms/0.45359237

- **Apple sharing**

Write a program that asks the user to provide two numbers - the number of students and the number of apples in a basket. Students split apples evenly and the rest of apples remain in the basket. The program displays:

- The number of apples each student gets;
- The number of apples which remain in the basket.

Example:

If there are 18 apples and 7 students, each student gets 2 apples and 4 apples remain in the basket.

If there are 100 apples and 29 students, each student gets (100//29) 3 apples and (100%29) 13 apples remain in the basket.

- **Sum of five consecutive integers**

Write a program that asks the user for an integer. Then the program displays five consecutive integers starting from the entered integer and their sum.

Example:

If the user provides the integer 7, the program displays values 7, 8, 9, 10, 11, and the sum (45 in this case).

- **Time units**

Write a program that asks the user to provide a time in seconds. The program displays the entered time in hours, minutes, and seconds.

Example:

If the user enters 123,456 for number of seconds, then program displays:

"The entered time represents 34hr 17min 36sec."

PRACTICE EXERCISES:

- **Printing**

Use print function to print following texts:

a) The name of the book is 'Python'.

b) The file address is C:\my_file.com.

- **Digits in float numbers**

Write a program that takes from the user a number with three decimal digits and displays the first and the second digit to the right of the decimal point of this rational number taken from the user.

Example: If the user enters 142.357, the program displays 35.

- **Time difference**

Write a program that asks the user to provide three numbers as information about the starting time and the three numbers as information about the final time of a process within one day. The program should display the number of seconds the process lasted and then the number of hours, minutes and seconds representing the process to last.

Example:

If the process started at 10:15:21 and ended at 13:10:01, then the process time was 10,480 seconds it means 2 hours 54 minutes and 40 seconds.

Note:

We will use European time form.

- **Amount in bills**

Write a program that asks a user for an amount of money to pay (in dollars). The program should display how many $100, $50, $20, $10, $5 and $1 bills the customer needs to pay the entered amount.

- **Returning time**

A neighbor runs every morning and leaves the house at 6:50. The neighbor runs two to seven miles each morning, it depends on the weather and his/her schedule, and he/she can run 1 mile in 8 minutes. Write a program that asks the user how many miles the neighbor ran in the morning and displays the time when he/she returned back home.

Example:

If the neighbor ran 3.4 miles in the morning, he/she returned at 7:18. Note: if the running time has decimal places, use the closest higher number of minutes for the return.