Intro to Python – Lesson 22 and 23

Along with menu-based programs, functions are also used to reduce repetitive code by placing it in its own block and calling it when needed. Check out the following video. There may be some repetition from yesterday, but the video adds more for today.

https://www.youtube.com/watch?v=_ypAw_pCOt8

Here are a couple of examples that you can try. Write each function, and then, in the main program, set up a statement that will call each function.

1. Create a function that accepts a grade between 0 and 100 and returns a letter grade based on the following:

80 – 100	A
70 – 79	B
60 – 69	C
50 – 59	D
less than 50	F

2. Create a function to calculate and return the weekly gross pay of an employee. The function should accept the number of hours worked and the hourly pay. Overtime is paid on all hours over 40 and is paid at a rate of 1.5 times the hourly pay.

3. Create a function that calculates and returns the ideal Training Heart Rate of a person. The function will accept the age, and the resting heart rate of a person. To calculate the THR, first subtract the age from 220, subtract the resting heart rate from the first result, and finally, multiply result 2 by 60%, and add the resting heart rate.

4. Write a function that accepts the total sales for a salesperson and returns their bonus. The bonus is based on 1% of sales. If sales are less than $5000.00, the commission will be 0, but if the sales are greater than 100000, then an extra 500.00 is added to the bonus value.

5. Write a function that requires an integer value and will return the sum of all the numbers up to and including the values passed.

For each of the following, write a function that can be used for the required calculations. Test each function by setting up a calling statement with possible values. For example if you create a function called DoStuff() with 3 parameters, test it using

DoStuff(5, “Text”, 78) OR First = 5
Second = ““Text”
Third = 78
DoStuff(First, Second, Third)

6. Write a function that will accept a Purchase date as a parameter and return payment date as the first day of the next month. If the day of the month is 25 or greater, add an extra month. For example, is the date is 2022-02-14 then return 2022-03-01. However, if the date is 2022-02-26 then return 2022-04-01.

7. Write a function that will calculate the gross pay based on a base salary plus commission based on a draw against commission. NOTE – a draw against commission is when an employee does not meet their sales quota and the base salary is deducted based on a percentage under the quota.

   All employees earn a base salary of $350.00 and have a sales quota of $5,000.00. If their sales are below the quota, calculate 10% of the amount they are under and subtract it from the base salary. If they have exceeded their quota calculate the commission at 4% of sales and add it to their base salary. In addition, if the employee has sales more than $10,000.00 add an extra $200.00 to the gross pay. If the employee has sales more than $20,000.00, add an extra $500.00 to the gross pay.

8. Write a function that will determine the customer status. The status will be “OK” if the customer is under their credit limit, “CHECK” if they are not over but within $200.00 of the credit limit, and “OVER” if they are over their credit limit. If the number of days since the last purchase is more than 60 days concatenate the word – PURCH (including the -) to the status. If the number of days since the last payment is more than 30 days add the text – PAY to the status. For a really bad customer the status may read OVER – PURCH – PAY. Return the Status and determine required parameters.

9. Come up with a function on your own that you feel would be useful when writing general programs. Include parameters and a return value.

See you at 1.