Sault college

Name: Gurpreet Singh

Student ID: 23059357

Assignment

Professor : Nawaz Chowdhary

Course : Introduction to Programming

#### **Program 1**

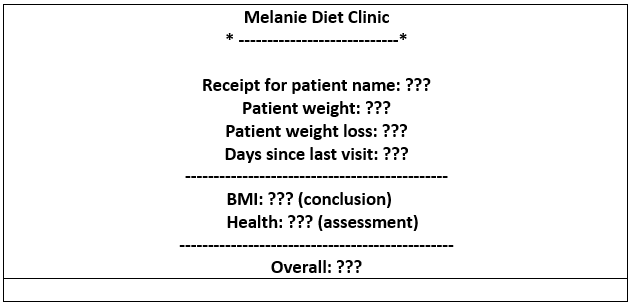
Create a Python script for the Melanie Dental Clinic. Take the following inputs from the user on console:

Melanie Dental Clinic

* Patient's name
* Has the patient been here before? (y/n)
* What is the patient’s height (in cm)? (int)
* What is the patient’s weight (in kg)? (float)
* When was the patient last weighed in (1/1/2000 if never weighed)? (date)
* What was the patient’s weight (in kg, -1 if never weighed)? (float)
* Practitioner’s overall assessment of the patient’s health (-5=very poor to +5=excellent, 0 for neutral) (int)

**Instructions:**

1. Take the above inputs on console from the user.
2. Use the data collected for the calculations in your program.
3. Calculate the patient’s health score based on:
   * Height
   * Weight
   * Change from previous weigh in
   * Practitioner’s assessment of health
     1. Calculate the patient’s BMI **(weight/ (height\*\*2)) \* 10,000**, rounded to 1 decimal point.
     2. Based on BMI calculation, BMI conclusion is as follows: over 30 is obese; 25-29.9 is overweight; 18.5-24.9 is healthy; 17-18.4 is underweight; under 17 is too thin.
     3. Intermediate health score is 0 for obese or thin, 3 for overweight or underweight, and 5 for healthy.
     4. Next, consider the change in weight. For a first weigh-in, add 1 to the intermediate health score. When weight does not change more than 1kg, subtract 1 from the intermediate score (and move to the next step). For an underweight person, if they lose more weight, subtract 3; if they gain more than 1kg, add 2. If they are thin, apply the same logic as for underweight but the change is either subtract 5 or add 5. For overweight, if they gain subtract 3, and if they lose more than 1kg add 2. For obese, apply the same logic as for overweight but the change is either subtract 5 or add 5.
     5. Finally, add practitioner’s health assessment (-5 to +5 value).
     6. Using the final score, patients should be ranked as: ‘Great condition!’ (score over 5); ‘Need a little work’ (score 3-5); ‘Need a lot of work’ (score 1-3); or ‘At risk!’ (score under 1).
4. Write up a report to share with the patient. If this is the first visit, weight loss is replaced with “NEW” and “Days since last visit” with “First visit.”



#### **Program 2**

Write a Python script for the following scenario by using function.

Program Logic and Output:

Create two functions: **takeInput()** and **displayResult().**

The function takeInput() is for taking the input from the user, and the function displayResult() is for implementing the calculations and displaying the result.

In the first function definition, get two numbers and an operator (+,-,\*,/) from the user.

In the second function definition, based on the user input, execute the arithmetic calculations. Display the formula and its result.

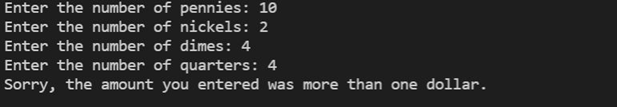
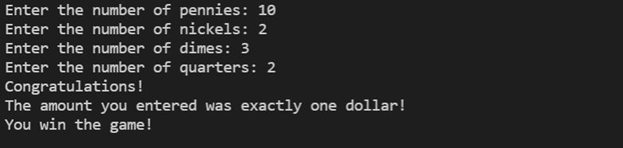
**For Example: 4 \* 2 = 8**

#### **Program 3**

Create a Python script for the “COUNT A DOLLAR” game. Follow the below instructions for this game:

1. Use the following constants as global variables:  
   PENNY\_VALUE = 1  
   NICKEL\_VALUE = 5  
   DIME\_VALUE = 10  
   QUARTER\_VALUE = 25  
   PENNIES\_IN\_DOLLAR = 100
2. Take the number of pennies, nickels, dimes, and quarters from the user.
3. Find the “totalCent” by multiplying the number of every type with its fixed value defined above.
4. Find the “totalDollars” by dividing “totalCent” with PENNIES\_IN\_DOLLAR (value is 100).
5. If the “totalDollars” value is greater than $1, then print the following message:  
   “Sorry, the amount you entered was more than one dollar.”
6. If the “totalDollars” value is less than $1, then print the following message:  
   “Sorry, the amount you entered was less than one dollar.”
7. Otherwise, print the following:  
   “Congratulations!  
    The amount you entered was exactly one dollar!  
    You win the game!”

**Sample Output1:**

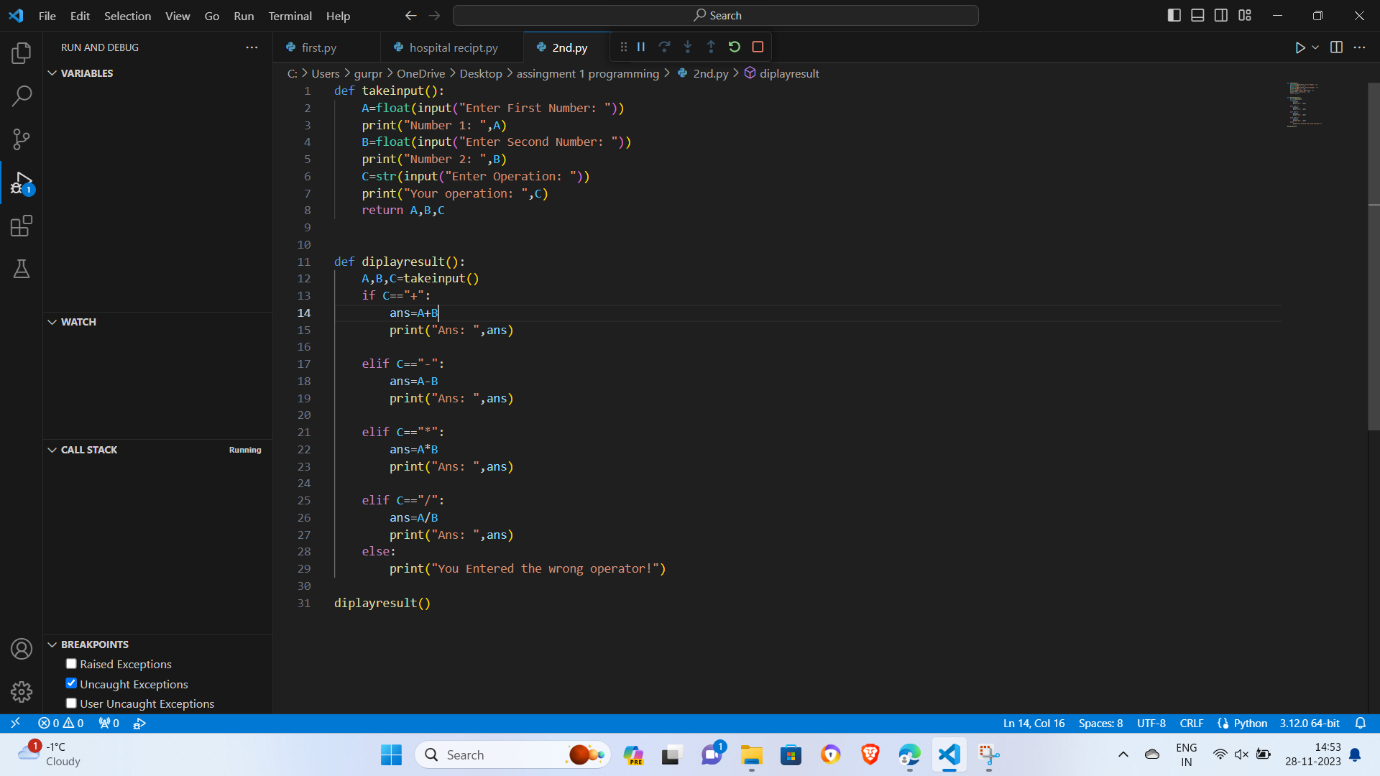
  
   
**Sample Output2:**  
 

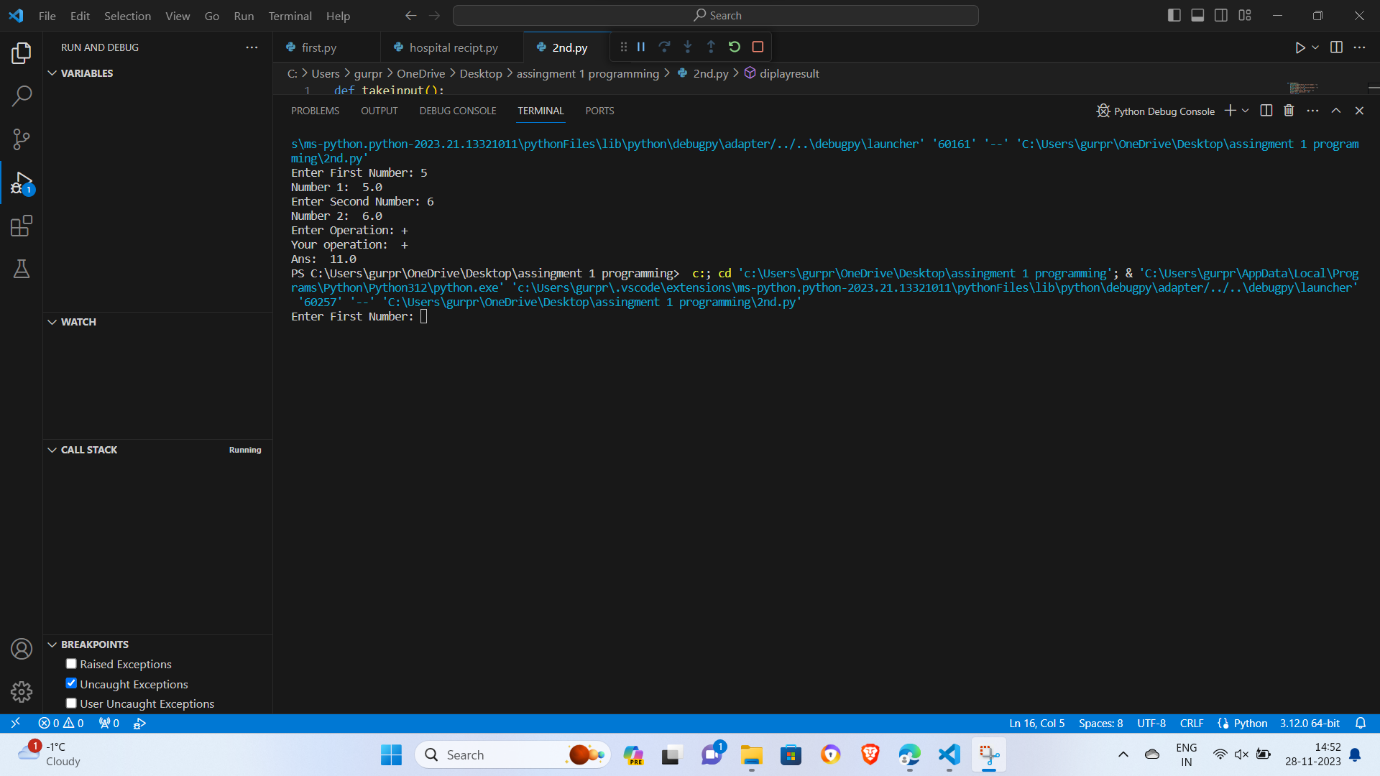
Contents

[Program2 6](#_Toc152076609)

[Program3 7](#_Toc152076610)

# Program2





# Program3

