*Turn on the Navigation Pane in Word when viewing this document:* **View tab > enable Navigation pane**

# Program Specification

* *Program title:* *Body Mass Index Analysis*
* *File name:* Ch3-HW-BMI-Analysis.py
* *Due Date:* see Canvas for due date
* *Main objective:* *Victorino Nutrition Center* is a walk-in center that provides basic nutritional information for local residents. They are requesting that you write a program that captures client data, determines their Body Mass Index, and displays feedback regarding their BMl. The system should prompt the user (an employee) to enter the Client’s data (name, age, height in inches, and weight in pounds), provide basic data validation of user input, and generate the output shown.
* Refer to the sample *Execution Screenshot* below for additional details.
  + Red, bold text is a calculated or determined value and thus a variable must be used to display that value.
  + Black, bold text is data entry.

# Read before starting

* The expectation is that students have read the *entire* chapter and watched *all* lecture recordings before beginning this assignment.
* Reminder: This is an *individual* assignment. Students are expected to complete this assignment independently and without the assistance of others. If you need help, contact your instructor via email or visit during office hours.
  + When emailing your instructor, please provide as much information as possible:
    1. State what the issue is and provide line numbers.
    2. Attach your \*.py file *zipped* as outlook does not permit attaching these files.
    3. Also, copy & paste the snippet of code where the issue is occurring within the email. If I am not at a computer when I see your email, I may be able to respond by seeing the snippet of code.
* I recommend that you read the entire assignment before beginning.
* Periodically review the ***Assignment Tips & Updates*** page in our module looking for tips as well as any ***required*** updates to this assignment.
* **This code must be free of syntax errors, or it will not be graded. Thus, comment out any code that crashes, or for a better grade, improve the code and submit it one day late with a 10% deduction.**

# Execution Screenshot

|  |
| --- |
| 1. Enter first name: **John** 2. Enter last name: **Doe** 3. Enter age: **20** 4. Enter height (in.): **70** 5. Enter weight (lbs.): **190.9** 7. Victorino Nutrition Center 8. ------------------- BMI Analysis ------------------- 10. Name: Doe, John 11. Age: 20 12. Height: 5'10" 13. Weight: 190.9 15. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 16. >> Client type: Adult 17. >> BMI: 27.4 (Overweight) 19. Due to your BMI Level of Overweight, you   should visit an internal medicine doctor   for further evaluation. 20. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

# Domain Information

*The Systems Analyst interviewed a Subject Matter Expert (SME - pronounced sm-ee) at Victorino and determined the following:*

## Age Categories

The Nutrition Center only sees Clients that are adolescents and older

* Adolescent 13-17
* Adult 18-44
* Middle Adult 45-64
* Older Adult 65+

## BMI Categories

* Optimal 18.5-25
* Underweight less than 18.5
* Overweight greater than 25

## BMI Calculation

* BMI = weight x 703 / height2

*Types of Doctors*

* Internal medicine doctor – treats adult patients
* Pediatrician – treats adolescent patients

*Output Message*

* When BMI is optimal, the system should generate the following message:

Congratulations, your BMI Level is Optimal!   
Return next year for your BMI Analysis.

* When BMI is not optimal, the system should generate the following message*.* The text in red will come from variables.

Due to your BMI Level of Overweight, you   
should visit an internal medicine doctor   
for further evaluation.

# Getting Started

* Open IDLE > open the *Editor Window* (File >NEW)
* Click File > Save As…
* Locate your Chapter 3 folder for this course
* Save your file as: **Ch3-HW-BMI-Analysis.py**

# Instructions

1. Include the required assignment header comment lines at the top of your code (as you did in the Chapter 2 assignment).
2. Add the comments lines shown below:
   * Notice we are now breaking down the general process comment into more detailed comments – these are all processing steps (i.e. doing calculations and comparisons (i.e. IF statements). Do not write a print statement in these 6 sections (in dark red); you will save each result to an appropriately named variable.

|  |
| --- |
| #Constants  #Input  #Data validation on age  #Convert height (in) to feet & inches  #Determine age category  #Calculate BMI  #Determine BMI level  #Determine Doctor  #Determine Message  #Output |

1. General rules to remember:
   1. Define all constants using the standard notation (ANY\_VALUE). For values that represent minimums appends the letters \_MIN or, if you are using maximums, append \_MAX.
   2. Readability – you will be graded on readability. At a minimum, make sure to leave 2 blank lines between each section. Also, leave a space between values, variables, operators, etc. – e.g. a = b + 10 print('hello', 'world')
2. Data validation
3. *Perform a single data validation to test the minimum age requirement* – use **a basic IF**; the client must be at least the age of an adolescent to receive a BMI Analysis; thus, if they are too young, display a message stating that they do not meet the minimum age requirement for BMI Analysis, and immediately stop/exit the application by coding the following:

**import sys**

**sys.exit()**

1. Once you have this completed, move this IF structure (along with the comment) immediately under the code to prompt for the user to enter an age, so that we can let them know immediately about the age issue.
2. Processing Requirements
3. *Convert height (in) to feet & inches* – the user entered height in total inches, however; you will need to separate that into *two* separate variables (feet, inches) by using the approach you learned in the Chapter 2 lectures on how to determine a quotient and a remainder.
4. *Determine age category* – Write this as an **IF-ELIF-ELSE** structure (see the top of Page 141 in the textbook). Use a single string variable to hold the appropriate *age category* to be displayed later in the output section. See lecture notes & recordings to code this efficiently.
5. *Calculate BMI* – this calculation is provided above. Store the result to a variable.
6. *Determine BMI level* – use an **IF-ELSE-IF** (i.e. test the series/range of values). Use a single string variable to hold the appropriate *BMI Level* to be displayed later in the output section. See lecture notes & recordings to code this efficiently.
7. *Determine Doctor Type* – set a variable to the correct type of doctor. See the notes on type of doctor above. By now, you should know what type of control structure you will need without being told – i.e. if something is true, it is one type of doctor; otherwise, it will be another type of doctor.
8. *Determine Message* – now it is time to generate the message (as shown in the output screen on Page 1 Line 20). Use a single variable to hold the appropriate *message* to be displayed later in the output section.
9. Output Requirements
10. Do not use string concatenation in this section as you need to practice writing arguments, and knowing how & when to use the *sep* argument.
11. Write the code to display lines 7-21 as shown in the screenshot on page 1.
12. Ensure all values are displayed. All values must update when a user enters new data.
13. Ensure all output is properly formatted (including spacing) as shown in the execution screenshot.
14. The final message that is displayed should not exceed past the line with 44 asterisks.

# Submitting the Assignment

* First, review the ***Assignment Tips & Updates*** page to see if there are any additional requirements to this assignment – these will be denoted with the tag: **[required].**
* To submit the assignment, go to Canvas and open the assignment link.
* Upload your \*.py file**.** You may upload more than once as long as it is before the due date & time. Only the final submission will be graded. Assignments will be accepted one day late albeit with a 10% deduction.