Fitchburg State University

CSC 7014 Practice Computer Programming

Instructor: Nguyen Thai

Due: 10/14/2016 at 5:00 PM

Student:

**CSC 7014 Assignment 5: Body Mass Index**

The purpose of this assignment is to learn how to program classes and objects. Your program is to be written in the Python language. You will be graded for output correctness, code comments, code indentation, descriptive variables and source code file header completeness.

As you work through the assignment be sure to answer all questions (type your answers into this document) and take all screenshots as requested (copy them into the document). For the screenshots, you can use the Snipping Tool that is built-in to Windows to capture the important parts of the lab as highlighted in the document below. Do not delete the contents of this file. When finished, you will submit the document source code file and associated data files to the instructor via Blackboard. DO NOT SUBMIT ZIP FILES OR INDIVIDUAL IMAGES. If you have any questions or need any clarification, email the instructor *before* the deadline.

1. In this lab you are to write a program in Python called *bmi.py* to implement the body mass index (BMI) class.
2. You are also to write another program in Python called *testBmi.py* to test the bmi class.
3. Body mass index (BMI) is a measure of health based on weight. It is calculated by taking a person’s weight in kilograms and dividing it by the square of person’s height in meters:
4. The interpretation of BMI for people 16 years and older:

|  |  |
| --- | --- |
| **BMI** | **INTERPRETATION** |
| Below 18.5 | Underweight |
| 18.5 – 24.9 | Normal |
| 25.0 – 29.9 | Overweight |
| Above 30.0 | Obese |

1. The UML of the BMI class is defined as following:



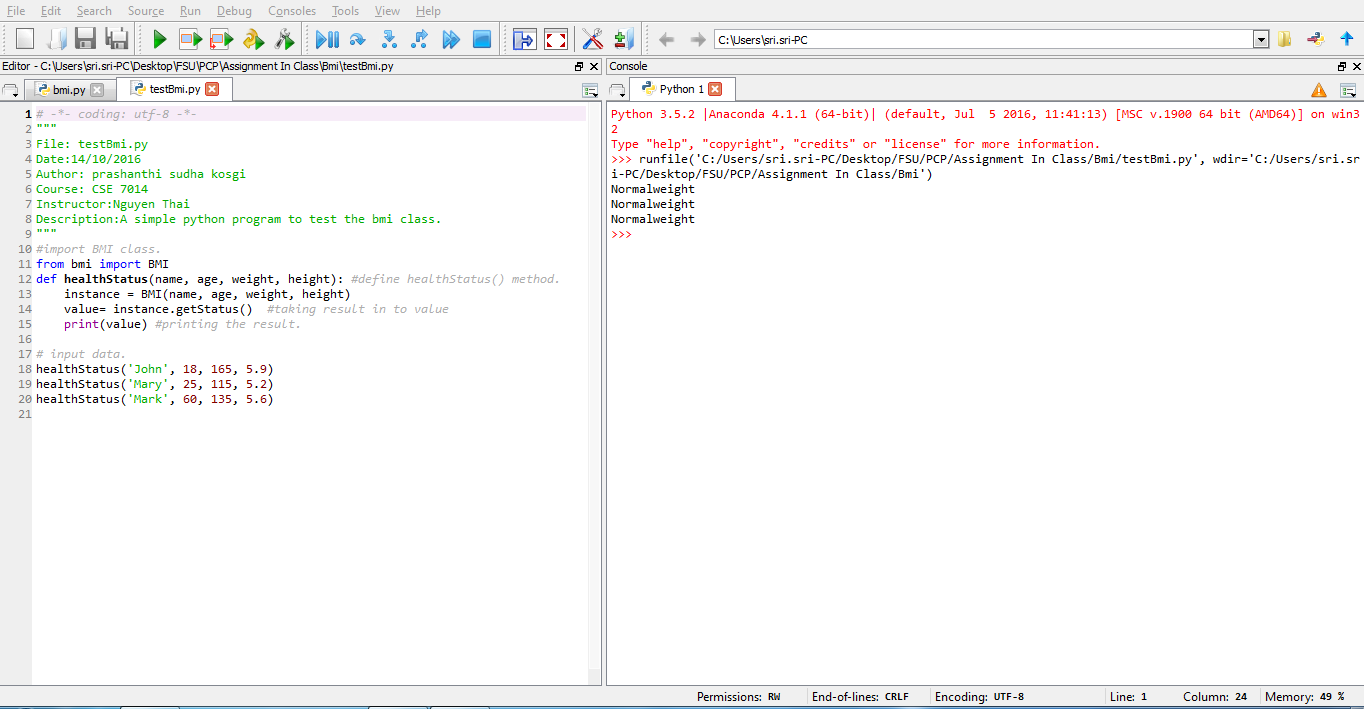
1. You need to convert units from English to Metric measurements.
2. In the testBmi.py, write a function called healthStatus() to take in appropriate parameters (name, age, weight, height) and return the health status of the person, i.e., Normal, Obese, etc.
3. In the testBmi.py, write test cases to test your BMI class, using the following numbers:
4. Test case 1:
   * Name: John
   * Age: 18
   * Weight: 165 lbs
   * Height: 5.9 ft
5. Test case 2:
   * Name: Mary
   * Age: 25
   * Weight: 115 lbs
   * Height: 5.2 ft
6. Test case 3:
   * Name: Mark
   * Age: 60
   * Weight: 135 lbs
   * Height: 5.6 ft

Print the test results to a console.

1. Before coding, think how you are going to tackle this problem, and write a short description of the logic of your program.

I am going to proceed with the following steps.

1. I will first create a BMI class and will create a constructor with parameters Name, Age, weight, and height.
2. I will create another method named getBMI which takes in values for the height and weight and computes the BMI according to the given formula.
3. In the same class I will create another method called getStatus(), if-else is used to check the condition for age. If the age is less than 16 years, system prompts the user to enter the valid date else program continues and executes conditional statements to compare the bmi value and returns the BMI status.
4. Next I will write another program to test the bmi class. In this program I will import the BMI class and will provide the values for the parameters(name,age,height,weight) which will return the health status of the person.
5. **INSERT YOUR DESCRIPTION HERE.**
6. In bmi.py program file, BMI class is created with a constructor and parameters (name, age, weight, height)(lines 9 to 15)
7. getBMI() method is created ,which contains a formula to calculate the bmi according to the input values and returns the value.(line 17 and 18).
8. getStatus() method is created . If-else is used to check the condition for age. If the age is less than 16 years, system prompts the user to enter the valid date else program continues and executes conditional statements to compare the bmi value and returns the BMI status. (lines 20 to 29).
9. Now another program tesiBMi.py is created to call BMI getStatus() method. After importing BMI class a healthStatus() method with the parameters is created (lines11 to 12).
10. Now the BMI values for name,age,height,weight are taken in to Instance and the status result is taken into value and the result is printed(lines13 to 15).
11. Method healthStatus() takes in appropriate parameters (name, age, weight, height) and returns the health status of the person(lines 18 to 20).
12. **TAKE A SCREENSHOT** of your input and output, and paste them here. Do not paste your source code in this document.



1. Submit your source code (*bmi.py, testBmi.py*) and this document to Blackboard for grading.