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 |

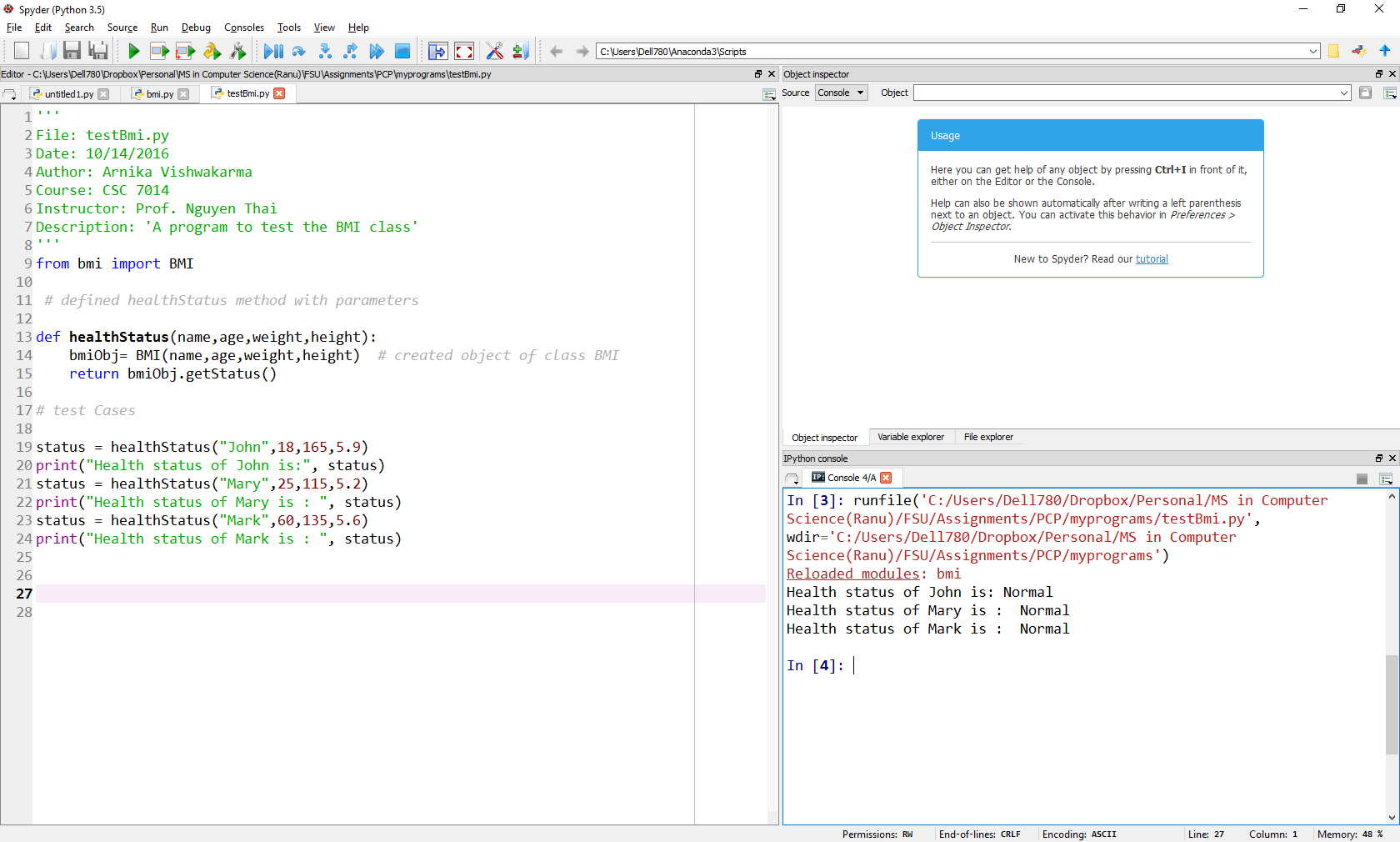
1. The UML of the BMI class is defined as following:



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.
2. **INSERT YOUR DESCRIPTION HERE.**
3. Firstly, we created file named bmi.py, now we will create a class named BMI. Inside this class we will initialize the initializer(\_\_init\_\_()) which get called automatically when an object is created for the BMI class.
4. Next, we will define a method named getBMI() inside which we will do the metric conversion of Pound to Kg i.e.(1 Pound = 0.453Kg) and Foot to Meters i.e.(1 ft= 0.3048 m), by applying the formula to calculate BMI , we will save the converted value in variable named ”bmi”. As, the converted output value is in float type so, to round it up upto 2 decimal we used format().
5. In next stage we defined getStatus() in which we used if-else statement to interpret the BMI value and give the health condition in result.
6. Next, we created another file with name testBmi.py. We imported our module Class BMI in this file using import statement.
7. After importing the class BMI, we defined the method healthStatus() and passed parameters (name, age, weight, height), inside this method we created an object (bmiObj) of class BMI and called the getStatus() with this object and returned the value.
8. Finally we called the function healthStatus() with arguments and stored the health status in variable ”status” printed the health status by using print().
9. **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 (primeNumber.py) and this document to Blackboard for grading.