

# Software Dev. & Problem Solving I

GCIS-123

**Due Date:** Monday Oct. 7, 2021 At 23:30

## Assignment 1

### Goals of the Assignment

This assignment provides an opportunity to practice all of the materials learned including Git, TDD, conditionals.

#### *Instructions:*

- *You will be using a specific repo for this assignment.*
- *You are required to commit after each step.*
- *A file header and doc strings for each function is required.*

### Assignment Problem:

Health Monitoring System (**HMS**) is a sophisticated technology, consists of a wearable wireless device like a bracelet with sensors that are paired with an application for a doctor to access the medical information. Among these important measurements are: pulse, oxygen levels, CO2 levels, thermometer, blood pressure...



Figure 1: <https://www.informationweek.com/mobile/remote-patient-monitoring-market-to-double-by-2016>

A. **Write python script for each of the following program skeleton functionalities:**

- ✓ Reading information for four hours, ***read\_pulse()***: this function reads pulse levels three times for each hour. Hint: you have to call this function 4 times.

- ✓ Measuring pulse levels, ***measure\_pulse()***: this function accepts three pulse values for each hour then calculate and return their average.
- ✓ Detecting abnormal levels, ***abnormal()***: this function accepts averages for the four hours and return the hour number in which the average is less than ***min\_level*** or greater than ***max\_level*** . (in this assignment we will assume that only one hour is abnormal) or return *None* for all normal levels.
- ✓ Warning doctors, ***warning\_doctor()***: this function print on screen warning messages about the hour number that has abnormal levels detection
- ✓ Drawing column chart, ***draw\_chart()***: (using turtle) to show pulse averages during the four hours.
- ✓ main function.

## B. Remember to use TDD:

- a. Write test
- b. Stub out function
- c. Run test (fail)
- d. Implement the solution from problem solving
- e. Run test (pass)

Since we are following the TDD process, start by creating the following test functions

1. ***test\_measure\_pulse()***. The test should assert all of the following conditions after the ***measure\_pulse*** function has been called.
  - All three pulse values are positive.
2. ***test\_abnormal()***. The test should assert all of the following conditions after the ***abnormal()*** function has been called.
  - Return *None* when all values within the interval `[min_value,max_value]`.

3. Stub out the new `measure_pulse()` function and use `pytest` to verify the test fails. **Don't forget to commit/push your code.**
4. Complete the `measure_pulse()` function. Upon completion, run the tests again and verify it passes.
5. **Commit/push your code.**
6. Add a call to `measure_pulse()` to `HMS.py`'s main function. Visually verify it returns the correct answers. **Commit/push your code.**

## Submission Instructions & Grading

Be sure that you have submitted the following two item:

- committed all your solution files to your GIT repository **before** the deadline.
- Submit the following sheet on MyCourses' respective assignment **before** the deadline. it includes the rubrics that will be used for grading the assignment.

GCIS123 - Software Dev. & Problem Solving I					
Assignment 1					
Exceptional Performance	Competent Performance	Acceptable Performance	Developing Performance	Beginning Performance	Unacceptable Performance
5 (100%)	4 (95%)	3 (88%)	2 (75%)	1 (50%)	0 (0%)
The solution is complete and correct. Instructions	The solution is functionally complete other than a small	The solution includes at least 80% of the required	The solution includes at least 50% of the required	The student began the solution, and there is	Little or no effort. Almost all of the

<p>were followed completely. or The student made a small number of minor errors for the first time, e.g. wrongly named files, output that does not match specifications, etc. A warning should be issued regarding any errors.</p>	<p>number of functional errors. Instructions were followed completely. or The solution is completely correct but the student made a small number of errors that they have made at least once before, e.g. wrongly named files, output that does not match specifications, etc. An explanation should be issued regarding any of these errors. Instructions were followed completely.</p>	<p>functionality and all instructions have been followed. or The student's effort is obvious based on the volume of work and commit history, but there are a modest number of errors that range from relatively minor to significant that can be easily fixed by a member of the course staff with minimal effort.</p>	<p>functionality and all instructions have been followed. The student's effort based on volume of work and commit history is obvious. There is at least some attempt to implement the missing functionality. Errors can be easily fixed by a member of the course staff in a short amount of time.</p>	<p>obvious effort, but more than 50% of the required functionality is missing or broken. This may include code that exists but does not compile and run, but could be fixed with significant effort.</p>	<p>required functionality is missing. The commit history shows very little activity on behalf of the student.</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------