Group Coursework Assignment Specification

(50 % of the module assessment)

Submission Deadline: 5pm, 15th December 2017.

# Problem description

Johnny Redknees is a world class Olympic record holding triathlete, famous for wearing his unique combination of technical material sports clothing.

During his diligent preparations, for each race Johnny and his team must decide on the best technical sports clothing. For footwear he has the choice of cycling shoes, running shoes or flippers. He also makes a choice between swimming goggles and sunglasses as eye protection. His unique approach, however, is that whatever he starts the event wearing, he does not remove or change it - this is what he keeps for the entire race (he claims changing them for each discipline affects his rhythm). The problem, however, is that one item may improve a discipline, but then have a negative effect on either or both of the other two.

Without detailed analysis, it is possible that Johnny may wear the wrong combination of clothing, reducing his chance of winning. As chief software engineer, your job is to give him and his team the essential critical data to help Johnny decide which combination of the optional technical clothing to wear at the start of each race to maximize his chance of winning. Johnny’s team has provided you with all of the necessary data in order to develop this decision making software.

# Background data

As a triathlete, Johnny completes the multi-stage competition through the three continuous and sequential endurance disciplines of swimming, cycling and running. The distances for each discipline are as follows:

|  |  |
| --- | --- |
| Swim | 1500m |
| Cycle | 40000m |
| Run | 10000m |

Table 1: Standard Triathlon Distances

When Johnny only wears his triathlon suit (no footwear or eye protection), he can race at the following average speeds:

|  |  |
| --- | --- |
| Swimming | 6.2km/h |
| Cycling | 52.8km/h |
| Running | 18.3km/h |

Table 2: Johnny’s basic average speeds, without technical clothing

The time taken (t) for Johnny to complete each discipline may be calculated using the following equation:

Through detailed experimentation, Johnny’s team has calculated the effect of each technical item of clothing item on his **speed** (and hence time) for each individual discipline as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Clothing | Swimming | Cycling | Running |
| Cycling Shoes | -10% | +12% | -25% |
| Running Shoes | -2% | +4% | +25% |
| Flippers | +60% | -5% | -30% |
| Swimming Goggles | +35% | -8% | -12% |
| Sunglasses | -10% | +8% | +5% |

Table 3: The effect on speed when wearing optional technical clothing items

# What you must do

**Question 1: 20 marks**

1. Write a Python program to apply the given formula above to complete the following table, for when Johnny DOES NOT wear any of the optional technical clothing. Rather than hard coding the distances travelled for each discipline, allow the user to enter the values (m) in Table 1 for each of the distances for the three disciplines.

|  |  |
| --- | --- |
| Discipline | Time Taken (s) |
|  |  |
|  |  |
|  |  |

Output the table in columns with column 1 aligned left and column 2 aligned right, with all numbers having decimal points lined up. Because of the importance of accuracy, calculate all results to 2 decimal places. Output a friendly message to the user to confirm the values entered, such as:

For swimming distance you entered: XXX metres

For cycling distance you entered: XXX metres

For running distance you entered: XXX metres

**NB**: ensure the correct (consistent) units are used to give the time output in **seconds**.

1. Output a friendly message to the user informing the team of how long it will take Johnny to complete an entire race (all 3 disciplines) from start to finish (in seconds to 2 decimal places).
2. Force the user to enter logical values for each discipline such as positive integer numbers.
3. You **must** define at least three appropriate functions and use those functions, the Python List and Loop structures in the program code.

**(save as q1.py)**

**(save the output screenshot after successfully running the code as q1capture.jpg)**

**Question 2: 30 marks**

1. Consider all the possible combinations of wearing footwear and eye protection. Modify the program in Question 1 (q1.py) to calculate the total time taken to complete the triathlon for all separate combinations of wearing an additional item of technical footwear and an additional item of eye protection. Output the results in tables of a design to your liking. Remember that the layout should be user friendly, give all the information needed and show accuracy to 2 decimal places.

You need to display several tables to show all the possible combinations of 3 possible footwear items with 2 possible eye protection items. You must use a loop when outputting the tables, with 1 table per iteration of the loop.

1. The key values of total time taken and chosen items of technical clothing must be held in a list.
2. Sort the list in (b) in the order of time taken, ascending. Save the contents of the ordered list with suitable messages into a Python text file in order to demonstrate to the team the time differences of using all combinations of optional clothing.

Options Time taken

===== ==========

…….. | .

…….. | .

……. | .

……. | .

.

1. Read data from the text file and Plot the times as a histogram (using pylab), including the additional time for Johnny to complete the race without any technical clothing. Save the graph to an image file called **q2d.png.** Remember to label the graph appropriately.
2. Define some more functions to increase the maintainability of the code and utilize the functions defined in q1.py

**(save as q2.py)**

**(save the output screenshot after successfully running the code as q2capture.jpg)**

**Question 3: 20 Marks**

1. Not all triathlons follow the same fixed distances for each discipline. Some (such as the Ironman event) are far longer. As such, allow the user to enter 3 different distances for each of the 3 disciplines (that will give 3 different lengths for the race in total). Applying the optimized clothing found in Question 2; display a user friendly message on the screen to show the time taken to complete the triathlon for each of the 3 total distances entered by the user.
2. Print a short discussion (max 200 words) on the screen in order to explain what you find. Could there be a particular discipline distance where the optimized clothing identified in Question 2 is no longer the best option?

**NB**: You can assume Johnny’s average speed is not affected by different discipline distances.

1. Utilize the appropriate functions defined in Q2.

**(save as q3.py)**

**(save the output screenshot after successfully running the code as q3capture.jpg)**

**Question 4: 20 marks**

As discussed, the lengths of the races could vary, depending on the difficulty of the triathlon. However, Johnny would like to know if there is a **swimming** distance at which wearing swimming goggles **and** flippers gives a faster overall time for the entire race when compared to just wearing the optimized clothing found in Question 2 (assuming the distances for cycling and running are fixed at 40000m and 10000m appropriately).

1. Your final task will be to produce 1000 random distances for the swimming phase (the upper limit is 20km and lower limit is 1km, uniformly distributed). For each of these random numbers, calculate the total time taken to complete the race and plot the results as a scatter diagram (total distance against time) in order to help show if there is a swimming distance most likely to make it beneficial wearing swimming goggles and flippers. Keep measurement units consistent and ensure the plot has suitable titles. Save it as **q4a.png.**
2. Store the random numbers in a list and use a loop to calculate each distance, and then find the total time taken to complete each race.
3. If you don’t find a swimming distance to be beneficial when wearing swimming goggles and flippers, could you increase the swimming distance beyond 20km? Would this make any difference to getting a faster time when compared to the optimum clothing? What is the effect on total time if you shorten the cycling and running distances? Print a short discussion on the screen (max 200 words); detailing any pivot points when wearing flippers and goggles does make Johnny race faster compared to if he wore the recommended optimum clothing found in Question 2.
4. Utilize the appropriate functions defined in Q2.

**(save as q4.py)**

**(save the output screenshot after successfully running the code as q4capture.jpg)**

# Report (10 marks)

In addition to the files mentioned you may write a short text file called README. This is to provide

* + any special instructions or warnings to the user (or assessor!) such as in which case code does not work ,
  + or to draw attention to any aspects of the program that you are particularly proud of (i.e. reusability, maintainability aspects of the code.)

The size of the report is two pages (1500 words, +/- 10 %).

# Notes on Expectations:

Below follows a qualitative description of some general expectation associated with this piece of coursework.

**Technical mastery of Python** Your programs should show mastery of what you have been taught.

**Design** Your programs should be well structured for the task in hand so that it is as easy as possible for:

* a user to use the program for any likely purpose,
* a programmer to understand the code structure and be able to develop it further,
* a programmer to be able to re-use as much as possible of the code in a related application.

**Clarity and Self-Documentation**

Given the structure of your programs, they should be as easy to read and understand as possible. Lay your code out so that it can be listed sensibly on a variety of devices: avoid having any lines longer than 80 characters as these may wrap (to reduce the number of “problem lines” you should use 4 spaces for indentation rather than tabs). Sensible names should be chosen for all variables, methods etc. Documentation strings should be included for each:

**Program** Fully explain what the program does and how it should be used.

Also state who wrote it and when.

**Function** State what each function does and explain the roles of its parameters.

In addition you should include occasional comments in your code; these may be (a) to introduce a new section in the code, or (b) to explain something that is not obvious. Bear in mind that pointless comments make your code harder to read, not easier.

# Version of Python:

Please write the code in Python 2.7x . Any other version of Python will **NOT** be accepted.

# Group Submission

All the files (including picture, README and the text file) should be compressed into a zip file and submitted electronically as directed on Learn.

# Individual Submission

In addition to the group submission, each group member is required to submit an individual assessment of how the project has gone by 5.00pm on the Friday of Week 11 (15th December 2017). This report assesses the way the group carried out the task and what contribution each member of the group made to the work (identify strengths and weaknesses of each individual including yourself). Include an estimation of the percentage contribution each team member contributed to the project. Use the cover sheet provided on Learn for this information.

The individual report will not be directly assessed as part of the group mark, but will be taken into consideration when individual marks are awarded. Individual mark will be calculated by using 70% of the group mark achieved and 30% of the contribution made towards to the project.

Firat Batmaz