Attach the assignment to your HW/Lab ENGR 102 Section **Lab 5a**
Date: _9/19/21 **DUE DATE: 9/2122**
You name _____Team # (table)_____

ENGR 102 Sect Lab 5a-team

70+38=108 points Canvas
ZyBook 12 points 100%

Reading assignment:

Lecture Slides	**L05**
zyBook	**Chapter 5**

Attention!!

This is a team assignment. For submission use a team header.

There are two deliverables for this team assignment. Please submit the following files to ***Canvas and zyBooks:***

For activity 1, submit pdf-word report file. – on ***Canvas (one submission per team)***

For activity 2,

submit your Py-files • ***diabetes_risk.py*** ***on ZyBook (individual submission on ZyBook)***

Submit your word/pdf file report and tests outputs. ***On Canvas (one submission per team)***

You will submit 3 files in total. ***2word/pdf files on Canvas and py-file on ZyBook***

For submission you may use this file as a template: rename file including your name. Do not forget to put your name inside of this file as well. If you are submitting py- files, make sure that they have headers.

Assignment: Writing and testing a larger program

In engineering and science, it is common that we want to calculate the effect of some complex behavior. To make this possible, we create a **model** of the behavior. A model attempts to describe the behavior in a way that is understandable and computable. Some models are based on physical laws and principles, some are based on replicating observations, and many are a combination of these. Once you have a model, you can use it to analyze and predict the performance or behavior of some system or phenomenon.

In this lab you will be using a model that has been created previously and you will generate a program that uses that model to give a user a prediction

In this case, you will use a model that has been developed to estimate someone's 10-year risk for likelihood of a heart attack, developed by the NIH. This model was constructed from data analysis of various factors that have been demonstrated to contribute to heart attacks for those ages 20-70.

The process is outlined on the final page of this document:

Risk calculation for type 2 diabetes

Early detection and treatment of type 2 diabetes is important to help reduce the risk of serious complications such as premature heart disease and stroke, blindness, limb amputations, and kidney failure. This score was developed to identify people at risk of having undetected diabetes. It is based on routinely collected information (age, sex, BMI, steroid/antihypertensive medication use, family history of diabetes, and smoking history).

Parameters used in the calculation:

Sex	Female: 0.879	Male: 0		
BMI	Under 25: 0	25 to 27.49: 0.699	27.5 to 29.99: 1.97	≥30: 2.518
Hypertension medication	On meds: 1.222	Not on meds: 0		
Steroids	On meds: 2.191	Not on meds: 0		
Smoker	Non-smoker: 0	Used to smoke: -0.218	Smoker: 0.855	
Family history	None: 0	Parent or sibling: 0.728	Parent and sibling: 0.753	

$$n = 6.322 + sex - (0.063 * age) - BMI - hypertension - steroids - smoker - history$$

$$risk = \frac{100}{1 + e^{n}}$$

The score was created from a notional population that was formed by random selection and pooling of two separate data sets. One data set came from a population-based sample of 1077 people, aged 40 to 64 years, without known diabetes, from a single Cambridgeshire general practice who underwent clinical assessment including an oral glucose tolerance test. The other data set came from a 12-month study in which 41 practices in southern England reported clinical details of patients aged 40 to 64 years with newly diagnosed Type 2 diabetes. Data were entered into a regression model to produce a formula predicting the risk of diabetes.
The performance of this risk score in detecting diabetes was tested in an independent, randomly selected, population-based sample. In the test population at 72% specificity, the sensitivity of the score was 77% and likelihood ratio 2.76. The area under the receiver-operating characteristic curve was 80%.

Reference:
Griffin SJ, Little PS, Hales CN, et al., Diabetes risk score: towards earlier detection of Type 2 diabetes in general practice, Diabet Metab Res Rev. 2000; 16: 164-71.

Note that the model works by taking several factors, assigning “point” values to them, and calculating an overall risk based on the sum of those points.

Lab5a. Activity 1 [50points]: Thinking about the program. Submit Word/Pdf file

Begin by putting together a document that you will use to analyze the problem. This should be done BEFORE any coding. This document should be saved as a Word/PDF and turned in before writing the program Activity 2.

First, you should review the table on the last page of the document referenced above, and ensure you understand how it works.

Begin by considering what values you need to store, and the general steps you will need to follow in your program.

- **[10 points]** Make a list of the variables you believe you are likely to need, and the names you will use to store each.

- **[10 points]** Create a sequence of steps that you will follow
    - Each step should be a short description of the goal of the step. The steps should not be code, or simple versions of code, but rather a description of the purpose of an action. Each step should be the equivalent of a few lines of code.
        - E.g. "Compute points based on age" might be a good description of a step.
    - If you have a conditional statement (and you should have several…), you might want to indicate each part of the condition as a separate action.

Next, determine a set of test cases. **[50 points]**
You will create a table with test cases. Example of the table is below. You may modify the table for your use.

- Create a list of test cases that you will use in your program. Be sure to handle both "typical" and "edge" cases. Do this before writing the program itself!
- You can create "intermediate" test cases. That is, your test cases do not have to be for the entire program, but you could write a test to verify that a particular part of your program is working.
- For this program, a "good" set of test cases would involve well **over 100 tests**, since many of the values to test are just to be pulled from tables. Since this is rather cumbersome, you do not have to give a full set, but
    - **[0.5*40=20 points]** You should have at least 40 different test cases (more is fine)
    - **[0.5*20=10points]** Your test cases should clearly include at least 20 "Edge/Corner" cases (points at boundaries)
    - **[0.5*10= 5 points]** Your test cases should clearly include at least 10 "typical" cases (not at edge/corner points)
    - **[7 points]** At least some cases test output edge/corner cases (i.e. boundaries between various output %)
    - Your test cases should address each part of the program (e.g. don't just test the age-based points for men).
- **[8 points]** Test cases labeled with purpose, input, expected output

For each test case, you should give a VERY BRIEF statement of what the test is covering (i.e. what part of the program is it testing and is it a "typical" or "edge/corner" case, what the data is you are testing on, and then what result you expect.
  - E.g. "Age-based score (typical); Sex: Male, Age: 62; Age-based score: 10"
  - E.g. "Entire program (extreme low); Sex: Female, Age: 20, etc. etc.; 10-year risk: <10%)"
- You will probably find it easiest to divide up the work for creating test cases, and then putting them all into a table.

When you have done this, save the document in Word/PDF format. Submit it.
Case test table, example.

#	Case Type	Case description, case base, purpose	Input	Expected result
1	Typical	Age-based		
2	Interm			
3	Edge	Extreme low		
4	Conner			

Activity 2: Constructing your program [50 points]. Submit your Py-file on ZyBook. Submit your word/pdf file with screenshots of your tests outputs
As a team, construct your program and name the file diabetes_risk.py.
As you do this, please be sure to do the following:

a) **[9 points]** Include comments for your program. You should probably begin by converting your list of steps into comments.
b) Develop incrementally. That is, write some code, and test it before writing the next section of code.
c) **[21 + 12 (ZyBook) points]** Be sure your program runs and passes all test cases. You should, in the process of developing, test every single one of the test cases you put together in Activity 1. You can submit your code as many times as you want to zyBooks to test it. Put your test results into a table
d) **[8 points]** Be sure to include specific instructions to the user for getting input and write a descriptive output.
e) Assume the user enters valid input only. Your code does not have to handle malicious users.
f) You may need additional input statements based on what the user enters.

For part c) use modified table from the previous task.

#	Case Type	Case description, case base, purpose	Input	Program Output	Program does not work
1	Typical				
2	Interm				
3	Edge				
4	Conner				

Example output (using inputs: female, 37, 26 BMI, not taking hypertension medication, not taking steroids, non-smoker, no family history of diabetes):

```
Enter your sex (M/F): F
Enter your age (years): 37
Enter your BMI: 26
Are you on medication for hypertension (Y/N)? n
Are you on steroids (Y/N)? n
Do you smoke cigarettes (Y/N)? n
Did you used to smoke (Y/N)? n
Do you have a family history of diabetes (Y/N)? n
Your risk of developing type-2 diabetes is 1.5%
```