

Homework #3

Decision Structures and Boolean Logic (100 points)

Directions:

- Create the following programs in Python that perform the purpose described and produce the sample output (with user input where applicable).
- Include at least three lines of comments/remarks in each program, including your name, the date, and a description of the program.
- Upload your .py Python program to our Moodle web site for each of the following programs.
- Be sure to adhere to the college and course academic honesty policy.

Complete the following 6 programs for Homework #3

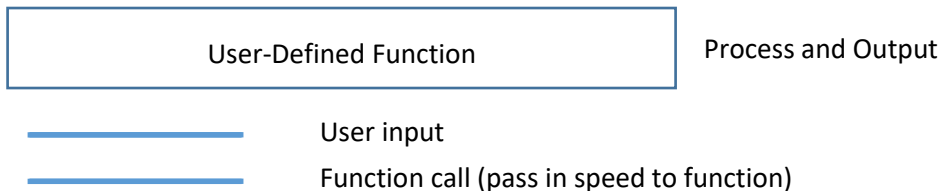
1. Wind Speeds (one function)

Create a Python program named **hw3-1.py** with the following functionality.

- The program will prompt the user to enter a sustained wind speed in the main part of the program and store it as an integer.
- The program will then call a user-defined function and pass the speed variable into it.
- The user-defined function will then use a decision (IF) statement to determine the level of a hurricane based on the sustained wind speed and output two things:
 - The message: This is a category X hurricane on the first line, where X is 1 through 5 (see NOAA web site link below. This line of output is to be blue color.
 - The paragraph describing the type of damage due to the winds from this web site. This line of output is to be black color.
 - See sample runs below.

<https://www.nhc.noaa.gov/aboutsshws.php>

Program organization:



Below is a sample run of the program (88 is user input from keyword):

Please enter sustained wind speed: 88

This is a category 1 hurricane

Very dangerous winds will produce some damage: Well-constructed frame homes could have damage to roof, shingles, vinyl siding and gutters. Large branches of trees will snap and shallowly rooted trees may be toppled. Extensive damage to power lines and poles likely will result in power outages that could last a few to several days.

Below is another sample run of the program (187 is user input from keyword):

Please enter sustained wind speed: 187

This is a category 5 hurricane

Catastrophic damage will occur: A high percentage of framed homes will be destroyed, with total roof failure and wall collapse. Fallen trees and power poles will isolate residential areas. Power outages will last for weeks to possibly months. Most of the area will be uninhabitable for weeks or months.

2. Create a tuition calculator for Henry Ford College (using one function)

Create a Python named **hw3-2.py** program that is a Henry Ford College tuition calculator.

It should perform similar to, and return amounts equal to this calculator on the college's web site:

<https://www.hfcc.edu/tuition-and-payment/calculator>

This calculator uses these tuition prices and fee structure:

<https://www.hfcc.edu/tuition-and-payment>

In the main part of the program:

- Your program should first ask the user if they are in-district, out-of-district, or international
- Then ask them how many credits of 100 to 200 level credits they plan to take
- Then ask them how many credits of 300 to 400 level credits they plan to take

A function will then be called with the three values described above passed into it. The function will calculate the student's tuition. You will need an IF-ELSEIF-ELSEIF statement to determine which type of student they are.

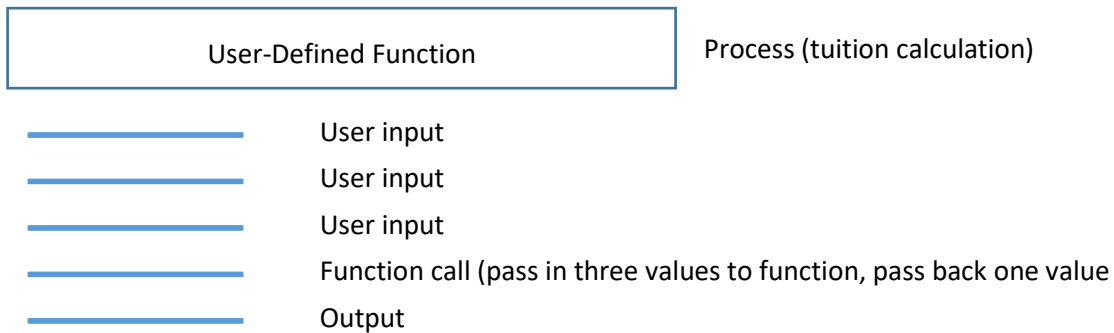
Pass the total tuition amount back the main part of the program and professionally display it.

Thus, the function will have three values passed in and one value passed back.

Here is a sample run of the program:

```
1 - In District
2 - Out of District Student
3 - Out of State / International Student
Choose one of the above (1-3):1
How many 100-200 level credits do you plan to register for?6
How many 300-400 level credits do you plan to register for?0
Your tuition cost will be $857.00
```

Program organization:



3. Password Strength Program (using one function)

Create a Python program named **hw3-3.py** that verifies the strength of a password that a user is entering is strong (complex/secure) enough.

- In the **main area** of the program, prompt the user to enter a password.
- Then, **call a function**, passing into it the password they entered.

In the function, use whatever Python commands or built-in functions are available to ensure:

- The password is at least 10 characters in length
- The password is mixed case (upper and lower)
- Includes at least one number
- You have at least one of these valid special characters in your password
\$! @ \$ % ^ & * #

The function will determine if the password is strong enough and then output the correct message. Here is a sample run and output from the program:

```

D:\python>py password.py
Enter a password: thisis
Your password length is too short. Please choose a password that is at least 8
characters long.
Your password is not mixed case. Please chose a password with mixed case.
You do not have a valid special character in your password. Plesae add at least
valid one special character.

D:\python>py password.py
Enter a password: thisismypassword
Your password is not mixed case. Please chose a password with mixed case.
You do not have a valid special character in your password. Plesae add at least
valid one special character.

D:\python>py password.py
Enter a password: Thisismypassword
You do not have a valid special character in your password. Plesae add at least
valid one special character.

D:\python>py password.py
Enter a password: thisis!myPassWORD
Thank you. Your password is valid.

```

4. Math Quiz (with three user-defined functions)

Create a program named **hw3-4.py** that provides a math quiz to the user. Your program must work as described below.

The program will contain **three user-defined functions**. The main part of the program will contain just three lines: a function call to each of the three user-defined functions.

- The **first function** will present the user with a **menu** of four math operations (addition, subtraction, multiplication, modulus) available for the quiz. The user will choose one of the options by entering a number 1 through 3. This choice will be passed back to the main part of the program. So, the function will have no data passed in, but will pass data back. **Name this function menu.**
- The **main program will then call the second function**, passing in the type of math operation the user chose from the first function. The second function will then generate and display a corresponding math problem using two random numbers between 3 and 20. Lastly, the second function will calculate and return (but not display) the correct answer to this math problem back to the main. So, this function will have one value passed in and one value passed back. **Name this function displayProblem.**
- The **third function will prompt the user for their answer, compare the users' response to the correct answers, and then display a message as to whether they got the answer correct.** If they got the answer incorrect, the program will display the correct answer. So, this function will have a value passed into it, but no data passed back. **Name this function checkAnswer.**

Your program and output should look/work like these two sample run examples:

```
1. Addition
2. Subtraction
3. Multiplication
Select math operation for quiz (1-3): 1
17
+ 18
Enter your answer: 35
Correct! Great job.

1. Addition
2. Subtraction
3. Multiplication
Select math operation for quiz (1-3): 2
9
- 9
Enter your answer: 2
Sorry. That is not correct. The answer is: 0
```

5. Medical Diagnosis (no functions needed)

Flowcharts are extremely common in business and life. They are commonly used to visualize and guide troubleshooting and diagnoses in a variety of fields: computers, automotive repair, health care, etc. Performing a Google image search for Simple Diagnostic flowchart -or- medical diagnosis flowchart returns hundreds of examples.

Network Flowchart

<http://www.ccri.edu/it/helpdesk/images/network-flowchart.gif>

Diagnostic flowchart for a car that won't start or stalls

<http://www.ifitjams.com/starting.htm>

Laptop Repair - Troubleshooting Toshiba, Dell, HP, IBM Acer and Sony laptops

<http://www.ifitjams.com/repair.htm>

<http://www.ifitjams.com/lappower.htm>

Troubleshooting Hardware Problems: Community College of Rhode Island

<http://www.ccri.edu/it/helpdesk/pc-troubleshooting.html>

Computer Repair with Diagnostic Flowcharts Third Edition

http://www.fonerbooks.com/ide_hd.htm

Diagnostic flowchart for diabetes insipidus

<http://www.usmle-forums.com/usmle-step-2-ck-forum/10195-diagnostic-flowchart-diabetes-insipidus.html>

Here is one for a dental/tooth diagnosis:

<http://dentistryandmedicine.blogspot.com/2011/12/apical-diagnosis-flowchart.html>

Create a Python program named **hw3-5.py** using IF statements that allow a user to respond to the questions in **this dental/tooth diagnosis flowchart** and then display the diagnosis

Here are a few sample runs of the program:

```
Is tooth tender to percussion? (y/n)n
Widened PDL or perialpical radiolucency?y
Is tooth necrotic? (y/n)y
Is sinus tract present?n
Diagnosis: Chronic apical abcess
```

```
Is tooth tender to percussion? (y/n)y
Is there swelling?y
Diagnosis: Acute apical abcess
```

6. Create a tuition calculator for Henry Ford College (using one function)

Make a copy of your program named **hw3-2.py** and name it **hw3-6.py**. Add decision logic and syntax after the user input to end the program, do not run the user-defined function, do not produce the tuition cost output if any of the three criteria exist and then produce an appropriate error message (see sample runs).

- a) Either 100-200 level credits value or 300-400 level credits value is below 0.
- b) Both 100-200 level credits value and 300-400 level credit value are 0.
- c) Either 100-200 level credits value or 300-400 level credits value are greater than 90.

Sample run #1:

```
1 - In District
2 - Out of District Student
3 - Out of State / International Student
Choose one of the above (1-3):1
How many 100-200 level credits do you plan to register for?111
How many 300-400 level credits do you plan to register for?1
Error. Please enter 90 or less credit hours for values. Please re-run program.
```

Sample run #2:

```
1 - In District
2 - Out of District Student
3 - Out of State / International Student
Choose one of the above (1-3):1
How many 100-200 level credits do you plan to register for?0
How many 300-400 level credits do you plan to register for?0
Error. Total credit hours (lower plus upper) must be greater than 0. Please re-run program.
```

Homework #3

Decision Structures and Boolean Logic

Extra Credit (10 pts)

1. Twonums (1 pt)

Create a Python program named **hw3-ec1.py**. In the main part of the program prompt the user to enter two integer numbers. Then, call a function and pass these two numbers into it. Use a Python IF - ELSE IF - ELSE statement to compare the two numbers and then display one of three messages:

The two numbers are equal.

The first number __ was larger than the second number __

The second number __ was larger than the first number __

(where __ is the number the user typed in)

Below is a sample run of the program:

```
Please first number: 2
Please second number: 3
The second number 3 was larger than first number 2
```

2. Create a roulette game. (4 pts)

Create a roulette game.

- The game will display the possible bet types. There will be 6 types of bets on this roulette table.

Even (payout: 1-1)	Odd (payout: 1-1)"
1st 12 (payout: 2-1)	2nd 12 (payout: 2-1)"
3rd 12 (payout: 2-1)	Specific number 1-36 (payout: 35-1)
- The game will allow the player to enter their bet type.
- The game will allow the player to enter the bet amount.
- If the player is playing on a specific number, the game will need to allow the player to enter the number they are placing their chip on.
- The game should prompt the player to press Enter to play.
- The game should then pause for a moment as the wheel spins.
- The game should then randomly choose a roulette number to land on (1-36).
- The game should then determine if the player won and display the appropriate output.
 - If the player did not win, indicate so, and what their number and the roulette number were.
 - If the player did win, indicate so, and what their winnings are.

Here are a few sample runs of the program:

```
1 - Even (payout: 1-1)          2 - Odd (payout: 1-1)
3 - 1st 12 (payout: 2-1)       4 - 2nd 12 (payout: 2-1)
5 - 3rd 12 (payout: 2-1)       6 - Specific number 1-36 (payout: 35-1)
How would you like to bet?1
How much money would you like to bet? $10
Press the Enter key to spin the roulette wheel.
Roulette wheel spinning...
Roulette wheel spinning...
You bet on even and the roll was 2 . You win $ 10.0
```

```
1 - Even (payout: 1-1)          2 - Odd (payout: 1-1)
3 - 1st 12 (payout: 2-1)       4 - 2nd 12 (payout: 2-1)
5 - 3rd 12 (payout: 2-1)       6 - Specific number 1-36 (payout: 35-1)
How would you like to bet?6
How much money would you like to bet? $10
What number do you want to place your chip on? 14
Press the Enter key to spin the roulette wheel.
Roulette wheel spinning...
Roulette wheel spinning...
You bet on the number 14 and the winning number was 12 . Sorry, you lose.
```

3. Change program: dollars and cents (3 pts)

Create a Python program named **hw3-ec3.py** that is a continuation of the change program from Homework #2. This program will allow a user to enter any dollar and cents amount and then will provide change. Your program must use at least one user-defined function to make the calculations. The program must allow the user to enter the dollar and cents as one input (variable) – not two different inputs. Here are several sample runs of the program:

```
Please enter the amount to make change for: $ 42.88
Number of hundreds:..... 0
Number of 50s :..... 0
Number of 20s:..... 2
Number of tens:..... 0
Number of fives:..... 0
Number of singles:..... 2
Number of quarters:..... 3
Number of dimes:..... 1
Number of nickels:..... 0
Number of pennies:..... 4
```



```
Please enter the amount to make change for: $ 1100.22
```

```
Number of hundreds:..... 11
```

```
Number of 50s :..... 0
```

```
Number of 20s:..... 0
```

```
Number of tens:..... 0
```

```
Number of fives:..... 0
```

```
Number of singles:..... 0
```

```
Number of quarters:..... 0
```

```
Number of dimes:..... 2
```

```
Number of nickels:..... 0
```

```
Number of pennies:..... 3
```

```
Please enter the amount to make change for: $ 867.64
```

```
Number of hundreds:..... 8
```

```
Number of 50s :..... 1
```

```
Number of 20s:..... 0
```

```
Number of tens:..... 1
```

```
Number of fives:..... 1
```

```
Number of singles:..... 2
```

```
Number of quarters:..... 2
```

```
Number of dimes:..... 1
```

```
Number of nickels:..... 0
```

```
Number of pennies:..... 4
```

4. Leap Year (2 pts)

Create a Python program named **hw3-ec4.py** that allows a user to enter a year (e.g. 1998, 1999, 2011, 2014, etc.), passes this value to a user-defined function and the function will determine and display whether that year is a leap year.

Below is two sample runs:

```
Enter year to check if leap year: 2004
```

```
2004 is a leap year
```

```
Enter year to check if leap year: 2001
```

```
2001 is not a leap year
```