# CSC121 Lab 02: Decision Structures and Boolean Logic

## Goals

In this lab assignment, students will demonstrate the ability to:

* Use selection control structures in Python programs
* Write if statements
* Use elif and else headers
* Write nested if statements
* Use logical operators and Boolean variables

NOTE: This lab document is a copyrighted work of Wake Tech Community College and the course instructor. Any posting of this document outside of Wake Tech is considered a copyright violation. Students who post these documents outside of Wake Tech are subject to academic and possible legal actions.

## Instructions

In this lab, you will demonstrate your mastery of decision structures and Boolean logic in Python.

Follow the instructions in each problem and submit the specified files. All problems will require that Python code be submitted as well as screenshots that prove the programs have been executed in PyCharm.

Problems 1 and 2 will start with code provided by the instructor that either needs to be fixed or completed. Problem 3 will consist of a program that you create from scratch that meets the problem specification.

## Before You Begin

* Since this is a new lab assignment, you should create a NEW PROJECT with an appropriate name. You can put this project in the same location where you placed Lab 1. If you have not done other work, PyCharm will likely remember the Location, and you just need to choose a different Name for your project.
* Nothing else should need to be updated in the New Project dialog.

## Problems

### Problem 1

In this problem, you are given a completed program that has logic errors. You need to fix the errors.

This flawed program is an Air Quality Classifier. The user enters an AQI value, and the program outputs a specific health impact. Here are the classifications:

| **AQI Value** | **Classification** |
| --- | --- |
| 0-50 | Good |
| 51-100 | Moderate |
| 101-150 | Unhealthy for Sensitive Groups |
| 151-200 | Unhealthy |
| 201-300 | Very Unhealthy |
| 301 or more | Hazardous |

* The instructor has provided a file called **Lab02P1-FixTheErrors.py**. Download that file and rename it **Lab02P1.py**.
* Copy that file into your PyCharm project.
* Change the program header to include your name and the date.
* Run the program AS IS and see if you can determine what is wrong with the program.
* Take a screenshot of the program running that demonstrates an error and name the screenshot **Lab02P1-error.jpg**.  
    
  HINT: Try using the AQI value of 100 or 200 and see how those are incorrectly classified. To make sure you have the program completely correct, test all BOUNDARY CONDITIONS.
* Review the code and correct all the errors. You should eventually be able to run the program that accurately classifies the ages with no errors.  
    
  NOTE: PyCharm may give you a hint to "Simplify chained comparison" with a yellow underline. That suggestion can be safely ignored.

Run this program using the PyCharm Terminal. Take a screenshot of the Terminal that includes the line showing where you started the program run with the results. Name the screenshot **Lab02P1-ouput.jpg**.

Submit all three files, **Lab02P1.py**, **Lab02P1-error.jpg**, and **Lab02P1-output.jpg**, to Blackboard for credit.

### Problem 2

In this problem, you are given a partially completed program, and you need to update and fill in the rest of the program to produce the desired output.

This program is designed for a card game that determines the action to take based on the color and number of a drawn card. The card color can be red or black. The card number can be 2-10. Here's how the actions are determined:

| **Color** | **Number** | **Action** |
| --- | --- | --- |
| Red | Even | Discard 2 cards |
| Red | Odd | Draw 1 card |
| Black | Even | Play another card |
| Black | Odd | Skip next player’s turn |

* The instructor has provided a file called **Lab02P2-FillThisIn.py**. Download that file and rename it **Lab02P2.py**.
* Copy that file into your PyCharm project.
* Change the program header to include your name and the date.
* Replace every instance of **FillThisIn** after line 9 with correct code that will enable the program to generate the correct actions for the selected card.
  + You should ignore the FillThisIn on line 9. That line of code is included in the file to reduce the number of errors PyCharm will flag when you first open the file.

Sample Outputs:

Enter the color of the card: Red

Enter the number on the card: 4

Action: Discard 2 cards

Enter the color of the card: Black

Enter the number on the card: 5

Action: Skip next player's turn

Enter the color of the card: Red

Enter the number on the card: 12

Action: Card number out of range

* Run this program using the PyCharm Terminal.
* Take a screenshot of the Terminal that includes the line showing where you started the program run with the results.
* Name the screenshot **Lab02P2-ouput.jpg**.

Submit both files, **Lab02P2.py** and **Lab02P2-output.jpg**, to Blackboard for credit.

### Problem 3

Trish at Bargain Swap Shop has decided to add a gift card program for her loyal customers and for customers that buy a lot. Here's how it works:

* If a customer is in her loyalty program and they buy $50-100, they get a $15 gift card.
* If a customer is in her loyalty program AND they buy over $100, they get a $25 gift card.
* If a customer is not in her loyalty program but they buy over $100, they get a $5 gift card.

Note: Some customers aren’t eligible for a gift card. The program should indicate that.

With every purchase, Trish also must charge 6.5% sales tax on the total. The gift card is NOT included in the total and is not considered when calculating the tax.

Create a file named **Lab02P3.**py. Write a program that calculates the cost for the total purchase:

* Ask the user to enter the total purchase amount. NOTE: This number could be a floating point number.
* Ask the user if the customer is in the loyalty program (see sample output).
* Calculate the amount of sales tax on the total.
* Calculate the total with tax.
* Output the sales tax and the total with tax. The program should format all money amounts that are output with 2 digits after the decimal point.
* Announce to the user if the customer receives a gift card, and what amount of gift card they receive.

Sample Outputs:

Enter the total purchase amount: 120

Is the customer a loyalty program member (y/n): n

Sales tax: $7.80

Total after tax: $127.80

Gift Card Awarded: $5

Enter the total purchase amount: 95.45

Is the customer a loyalty program member (y/n): y

Sales tax: $6.20

Total after tax: $101.65

Gift Card Awarded: $15

Enter the total purchase amount: 65

Is the customer a loyalty program member (y/n): n

Sales tax: $4.23

Total after tax: $69.22

Gift Card Awarded: $0

NOTE: The last example in the Sample Output looks like a math mistake, but the total after tax result is due to how Python is doing rounding.

Run this program using the PyCharm Terminal. Take a screenshot of the Terminal that includes the line showing where you started the program run with the results. Name the screenshot **Lab02P3-ouput.jpg**.

Submit both files, **Lab02P3.py** and **Lab02P3-output.jpg**, to Blackboard for credit.

## Grading Rubric

### Grading rubric for Problems 1 (20 points)

* Program is well-formatted and has a correct header [5 points]
* All errors in the program were corrected [10 points]
* Screenshots demonstrates student executed the faulty program and the corrected program [5 points]

### Grading rubric for Problem 2 (30 points)

* Program is well-formatted and has a correct header [5 points]
* Program does execute correctly and produces correct results [20 points]
* Screenshot demonstrates student executed the program [5 points]

### Grading rubric for Problem 3 (50 points)

* Program is well-formatted and has a correct header [5 points]
* Program does execute correctly and produces correct results [40 points]
* Screenshot demonstrates student executed the program [5 points]