# CSC121 Lab 03: Repetition Structures

## Goals

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

* Use loops to solve problems
* Create and use condition-controlled loops
* Create and use count-controlled loops
* Calculate a running total
* Create and use input validation loops
* Create and use nested loops

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 repetition structures.

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.

## Problems

### Problem 1

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

This flawed program is supposed to estimate the inventory of books, DVDs, and games in Trish's Bargain Swap Shop.

* Initially the program asks for the current number of books, DVDs, and games.
* The program then assumes that there will be a net gain of 45 books per month, 32 DVDs per month and 15 games per month, and it shows the amounts of each item in stock each month for the next 3 months.

The instructor has provided a file called **Lab03P1-FixTheErrors.py**.

* Download that file and rename it **Lab03P1.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. Note: You do NOT need a screen shot of the error for this problem.
* Correct all the errors. You should eventually be able to run the program with no errors that accurately produces a table like the sample output.

Sample Output:

What is the current number of books? 20

What is the current number of DVDs? 100

What is the current number of games? 35

Month 1

Books: 65

DVDs: 132

Games: 50

Month 2

Books: 110

DVDs: 164

Games: 65

Month 3

Books: 155

DVDs: 196

Games: 80

Run this program. Take a screenshot showing where you started the program run with the results. Name the screenshot **Lab03P1-ouput.jpg**.

Submit both files, **Lab03P1.py** and **Lab03P1-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 should ask the user to enter an odd number for the size of the pattern. The program will then print an "X" pattern based on the user's input. The pattern follows these rules:

* The pattern should be a square grid with dimensions equal to the odd number entered by the user.
* The main diagonals of the square should be filled with asterisks (\*).
* All other positions should be filled with space characters.

Sample output 1:

Enter an odd number for the size: 5

\* \*

\* \*

\*

\* \*

\* \*

Sample output 2:

Enter an odd number for the size: 7

\* \*

\* \*

\* \*

\*

\* \*

\* \*

\* \*

HINT**:** Be careful about the positioning of the diagonals. You can determine the positions using conditions based on the row and column indices.

* The instructor has provided a file called **Lab03P2-FillThisIn.py**. Download that file and rename it **Lab03P2.py**.
* Copy that file into your PyCharm project.
* Change the program header to include your name and the date.
* Ignore the FillThisIn on line 8.
* After line 8, replace every instance of FillThisIn with correct code that will enable the program to generate the patterns shown above. Do not change any other code or comments.
* Run this program in either the PyCharm Terminal or using the Run button.
* Take a screenshot showing where you started the program run with the results.
* Name the screenshot **Lab03P2-ouput.jpg**.

Submit both files, **Lab03P2.py** and **Lab03P2-output.jpg**, to Blackboard for credit.

### Problem 3

Trish at Bargain Swap Shop has been using the programs you developed for her and really likes them. Recently however, she hired a new cashier who isn't very good at data entry. In particular, she needs the program to catch data entry errors that are made and give the user a chance to reenter the data.

To help her out, you'll be taking the first program you developed for her (Lab 01 Problem 4) and rewrite it so that it blocks out some data entry errors. Here's the changes that she'd like to see:

* A user should never be able to enter a negative number for the number of books, DVDs, or games that are being purchased. Zero is a valid entry.
* It is extremely rare that someone buys a LOT of a particular item, so Trish wants the program to limit the number of each item type someone can purchase. These are the limits she wants to set:
  + Books: Maximum of 30
  + DVDs: Maximum of 15
  + Games: Maximum of 10

Other than these checks on the data entry, the program should execute as it did before.

NOTE: We are **not** handling issues where a user types an invalid integer, that is, an entry which cannot be converted using the int() function. We will handle that in a future lab.

Remember, you are only allowed to use concepts we have learned in class up through this lesson. We have not learned about exception handling which would be needed to handle conversion issues.

The costs of each item type are the same as in Lab 01:

* Books are $2.25 each.
* DVDs are $4.35 each.
* Games are $5.00 each.

Create a file named **Lab03P3.**py. Write a program that calculates the cost for the total purchase:

* Ask the user to enter the number of workbooks, textbooks, and magazines being purchased.
* If the user enters a negative number, or goes above the maximum for a particular item, the program should continue to ask the user to enter a number until it is valid.
* Calculate the total before tax.
* Calculate the amount of sales tax on the total. (Tax is 6.5%.)
* Calculate the total after tax.
* Output the total before tax, the sales tax, and the total after tax.

Sample Output:

Enter the number of books: -10

Number of books must be between 0 and 30.

Enter the number of books: 60

Number of books must be between 0 and 30.

Enter the number of books: 10

Enter the number of DVDs: 18

Number of DVDs must be between 0 and 15.

Enter the number of DVDs: 5

Enter the number of games: -3

Number of games must be between 0 and 10.

Enter the number of games: -2

Number of games must be between 0 and 10.

Enter the number of games: 3

Cost before tax: $59.25

Sales tax: $3.85

Cost after tax: $63.10

Remember to format monetary amounts with 2 digits after the decimal.

NOTE: There are no discounts in this program.

Hint: If you need help with validating user input, review section 4.6 in your textbook.

Run this program. Take a screenshot showing where you started the program run with the results. Name the screenshot **Lab03P3-ouput.jpg**.

Submit both files, **Lab03P3.py** and **Lab03P3-output.jpg**, to Blackboard for credit.

## Grading Rubric

### Grading rubric for Problem 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 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]