\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CSC121 PYTHON Programming**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

LAB 04 **ITERATIVE CONTROL STRUCTURES**

# Objectives

In this lab assignment, students will learn:

- How to use loops to solve problems

- How to write while statements

- How to write indefinite loops

- How to write definite loops

# Goals

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

- Use loops to solve problems

- Write while statements

- Write indefinite loops

- Write definite loops

# Instruction and Problems

Write a Python program for each of the problems in this lab.

Please use PyCharm to type and test your programs. Submit the Python files to Blackboard for credit. In this lab, you should submit 4 Python files, one for each problem.

## Problem 1

Diabetes is diagnosed by measuring patient’s fasting plasma glucose level (FPG). If FPG is higher than 125, the patient has diabetes. If FPG is higher than 100 but not higher than 125, the patient has pre-diabetes. FPG 100 or lower is considered as a healthy level. Write a program to do diabetes diagnosis for multiple patients. For each patient, enter FPG and determine whether the patient has diabetes, pre-diabetes or a healthy FPG level. Ask the user to answer a yes/no question to indicate whether he/she wants to do diagnosis for another patient.

The following is an example:

Enter fasting plasma glucose level: 126

This patient has diabetes

Checking diabetes for another patient? [y/n] y

Enter fasting plasma glucose level: 97

This patient has healthy fpg level

Checking diabetes for another patient? [y/n] y

Enter fasting plasma glucose level: 125

This patient has pre-diabetes

Checking diabetes for another patient? [y/n] y

Enter fasting plasma glucose level: 100

This patient has healthy fpg level

Checking diabetes for another patient? [y/n] y

Enter fasting plasma glucose level: 101

This patient has pre-diabetes

Checking diabetes for another patient? [y/n] n

Save your Python program in a file named **Lab04P1.py**. Submit the file to Blackboard for credit.

## Problem 2

In Lab 03 we wrote a program to calculate the number of seconds since midnight. Modify the program by adding error checking loops. Hour must be a number from 1 to 12. Minute and second must be numbers from 0 to 59. Also check whether “AM” or “PM” is entered. Every time an invalid value is entered, display an error message and ask the user to re-enter a valid value immediately.

The following is an example:

Enter hour: 0

Hour must be from 1 to 12.

Enter hour: 13

Hour must be from 1 to 12.

Enter hour: 12

Enter minute: -1

Minute must be from 0 to 59.

Enter minute: 60

Minute must be from 0 to 59.

Enter minute: 14

Enter second: -1

Second must be from 0 to 59.

Enter second: 60

Second must be from 0 to 59.

Enter second: 47

Enter AM or PM: FM

Please enter AM or PM

Enter AM or PM: PM

Seconds since midnight: 44087

Save your Python program in a file named **Lab04P2.py**. Submit the file to Blackboard for credit.

## Problem 3

A simple physics experiment is to drop a ball and see how high it bounces. Once the “bounciness” of the ball has been determined, the ratio gives a bounciness index. For example, if a ball dropped from a height of 10 feet bounces 6 feet high, the index is 0.6. If the ball were to continue bouncing, the height it reaches in the second bounce would be 3.6 feet (6 feet \* 0.6 = 3.6 feet). Write a program that asks the user to enter the initial height of the ball, the bounciness index and the number of times the ball is allowed to continue bouncing. Display the height the ball reaches in each bounce.

The following is an example.

Enter initial height: 8

Enter bounciness index: 0.5

Enter number of times the ball is allowed to bounce: 6

Number of bounces: 1 Height: 4.0

Number of bounces: 2 Height: 2.0

Number of bounces: 3 Height: 1.0

Number of bounces: 4 Height: 0.5

Number of bounces: 5 Height: 0.25

Number of bounces: 6 Height: 0.125

Save your Python program in a file named **Lab04P3.py**. Submit the file to Blackboard for credit.

## Problem 4

Instructors in a community college are paid on a schedule that provides a salary based on their number of years of teaching experience. For each year of experience after the first year, up to 10 years, the instructor receives a 2% increase over the preceding value. Suppose the initial salary of an instructor is $50,000. In the second year, this instructor’s salary will be $51,000 ($50,000 + $50,000 \* 0.02 = $51,000). In the third year, the salary will be $52,020 ($51,000 + $51,000 \* 0.02 = $52,020), and so on. In addition, the instructor is required to deposit 5% of the salary each year into a retirement fund account. For example, if the salary is $50,000 in a year, $2500 ($50,000 \* 0.05) will be deposited into his/her retirement fund account. Write a program to do the following. Ask the user to enter the starting salary. Calculate and display the salary each year in the first 10 years. Also calculate and display a running total of the instructor’s retirement fund after each year.

The following is an example:

Enter starting salary: 50000

Year: 1 Salary: 50000.0

Retirement Found Total So Far: 2500.0

Year: 2 Salary: 51000.0

Retirement Fund Running Total: 5050.0

Year: 3 Salary: 52020.0

Retirement Fund Running Total: 7651.0

Year: 4 Salary: 53060.4

Retirement Fund Running Total: 10304.02

Year: 5 Salary: 54121.608

Retirement Fund Running Total: 13010.100400000001

Year: 6 Salary: 55204.040160000004

Retirement Fund Running Total: 15770.302408000001

Year: 7 Salary: 56308.1209632

Retirement Fund Running Total: 18585.708456160002

Year: 8 Salary: 57434.283382464004

Retirement Fund Running Total: 21457.4226252832

Year: 9 Salary: 58582.969050113286

Retirement Fund Running Total: 24386.571077788867

Year: 10 Salary: 59754.628431115554

Retirement Fund Running Total: 27374.302499344645

Save your Python program in a file named **Lab04P4.py**. Submit the file to Blackboard for credit.

# Grading rubric for Each Problem

Writing loops [15 points]

Other statements [10 points]