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

**CSC130 Computing Fundamentals II**

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

LAB 01A **BASIC PYTHON PROGRAMMING (PART 1)**

# Objectives

- Perform simple input and output operations

- Perform operations with numbers such as arithmetic and comparisons

- Perform operations with Boolean values

- Implement an algorithm using the basic constructs of sequences of statements, selection statements, and 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.

## Program 1

An employee’s total weekly pay equals the hourly wage multiplied by the total number of regular hours plus any overtime pay: Overtime pay equals the total overtime hours multiplied by 1.5 times the hourly wage. Write a program that takes as inputs the hourly wage, total regular hours, and total overtime hours and displays an employee’s total weekly pay.

Example:

Enter hourly wage: $10.50

Enter the regular hours: 20

Enter the overtime hours: 12

The total weekly pay is $399.0

## Program 2

A standard science 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 form a height of 10 feet bounces 6 feet high, the index is 0.6 and the total distance is 16 feet after one bounce. If the ball were to continue bouncing, the distance after two bounces would be 10 ft + 6 ft + 6 ft + 3.6 ft = 25.6 ft. Note that distance traveled for each successive bounce is the distance to the floor plus 0.6 of that distance as the ball comes back up. Write a program that lets the user enter the initial height of the ball and the number of times the ball is allowed to continue bouncing. Output should be the total distance traveled by the ball.

Example:

Enter the height from which the ball is dropped: 10

Enter the bounciness index of the ball: 0.6

Enter the number of times the ball is allowed to continue bouncing: 5

Total distance traveled is: 36.8896 units.

## Problem 3

A company gives volume discount to customers of their software product. Unit price depends on number of copies purchased:

|  |  |
| --- | --- |
| Number of copies purchased | Unit price |
| 1 - 10 | $99 |
| 11 - 50 | $89 |
| 51 - 100 | $79 |
| 101 or more | $69 |

Write a program to do the following. Ask the customer how many copies he is buying. Display the unit price. Calculate and display total price. For example, if a customer is buying 20 copies, unit price will be $89 and total price will be $1780. The following are two examples.

How many copies are you buying? 25

Unit price: $89

Total price: $2225

How many copies are you buying? 100

Unit price: $79

Total price: $7900

## Problem 4

In a rock-paper-scissors game, paper beats rock, rock beats scissors and scissors beat paper. Write a program to play rock-paper-scissors. This game is played by two players. Player 1 enters his choice first, followed by player 2. Ask the players to enter R for rock, P for paper or S for scissors. The players can enter either uppercase or lowercase letter. If any user enters something other than those three letters, the game is canceled for invalid entry. Otherwise, determines and displays the outcome (player 1 wins, player 2 wins or tie). The following are three examples.

Player 1: Enter R for rock, P for paper or S for scissors: R

Player 2: Enter R for rock, P for paper or S for scissors: S

Player 1 has won the game.

Player 1: Enter R for rock, P for paper or S for scissors: r

Player 2: Enter R for rock, P for paper or S for scissors: R

Tie

Player 1: Enter R for rock, P for paper or S for scissors: p

Player 2: Enter R for rock, P for paper or S for scissors: X

Game canceled because of invalid entry

# Grading rubric

Program 1:

Getting input [5 points]

Processing [15 points]

Displaying output [5 points]

Program 2:

Getting input [5 points]

Processing [15 points]

Displaying output [5 points]

Program 3:

Getting input [5 points]

Processing [15 points]

Displaying output [5 points]

Program 4:

Getting input [5 points]

Processing [15 points]

Displaying output [5 points]