

COSC 1306 - Prog for Non-Majors

I/O Processing-1

Dr. Mohan

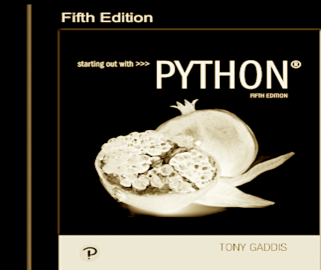
McMurry University

September 5, 2023



Lesson Topics Overview

- Designing a Program.
- Input, Processing, and Output.
- Displaying Output with print Function.
- Comments.
- Variables.
- Reading Input from the Keyboard.

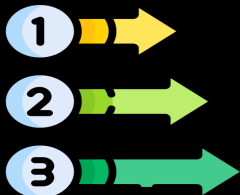


Chapter 2

Input, Processing, and
Output

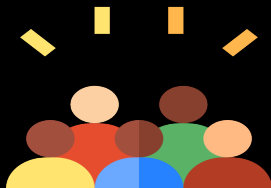
Designing a Program (1 of 3)

- Programs must be designed before they are written.
- Program development cycle:
 - Design the program.
 - Write the code.
 - Correct syntax errors.
 - Test the program.
 - Correct logic errors.



Designing a Program (2 of 3)

- Design is the most important part of the program development cycle.
- Understand the task that the program is to perform.
- Work with customer to get a sense what the program is supposed to do:
 - Ask questions about program details.
 - Create one or more software requirements.



Designing a Program (3 of 3)

- Determine the steps that must be taken to perform the task.
 - Break down required task into a series of steps.
 - Create an algorithm, listing logical steps that must be taken.
 - Algorithm is a set of well-defined logical steps that must be taken to perform a task.



Pseudocode

- Pseudocode: Informal language that has no syntax rule and not meant to be compiled or executed. A.k.a fake code.
- Mainly used to create model program and no need to worry about syntax errors, can focus on program's design.
- Can be translated directly into actual code in any programming language.

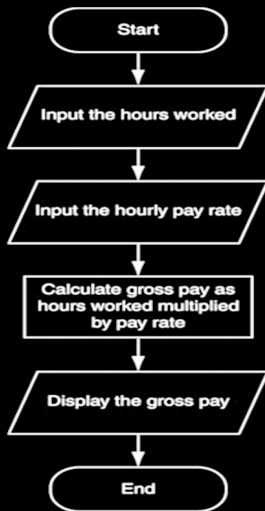


Flowcharts (1 of 2)

- Flowchart: diagram that graphically depicts the steps in a program.
- Ovals are terminal symbols.
- Parallelograms are input and output symbols.
- Rectangles are processing symbols.
- Symbols are connected by arrows that represent the flow of the program.

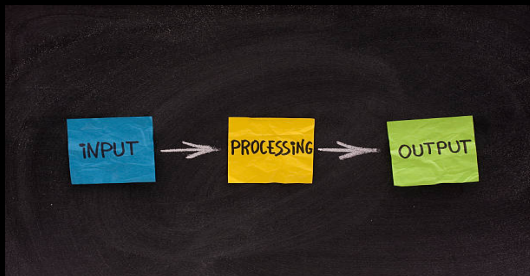


Flowcharts (2 of 2)



Input, Processing, and Output

- Typically, computer performs three-step process:
- **Receive input**: Input is any data that the program receives while it is running.
- **Perform** some process on the input. Example: mathematical calculation.
- **Produce** output.



Displaying Output with the print Function

- Function: piece of prewritten code that performs an operation.
- print function: displays output on the screen.
- Argument: data given to a function. Example - data that is printed to screen.
- Statements in a program execute in the order that they appear from top to bottom.



Strings and String Literals

- String: sequence of characters that is used as data. Must be enclosed in single (') or double (") quote marks.
- String literal: string that appears in actual code of a program. Can be enclosed in triple quotes (''' or ''').
- Enclosed string can contain both single and double quotes and can have multiple lines



Comments

- Comments: notes of explanation within a program. Ignored by Python interpreter.
- Intended for a person reading the program's code.
- Single-line comment begin with a # character. Multi-line comments are wrapped using three single quotes (""").
- End-line comment: appears at the end of a line of code. Typically used to explain the purpose of that line.



Variables

- Variable: name that represents a value stored in the computer memory.
- Used to access and manipulate data stored in memory. In other words, a variable references the value it represents.
- Assignment statement: used to create a variable and make it reference data.
- General format is variable = expression.

Example: age = 29

- Assignment operator: the equal sign (=)



Variables (cont'd.)

- In assignment statement, variable receiving value must be on left side.
- A variable can be passed as an argument to a function and the variable name should not be enclosed in quote marks.
- You can only use a variable if a value is assigned to it.



Variable Naming Rules

- Rules for naming variables in Python:
 - 1 Variable name cannot be a Python key word.
 - 2 Variable name cannot contain spaces.
 - 3 First character must be a letter or an underscore.
 - 4 After first character may use letters, digits, or underscores.
 - 5 Variable names are case sensitive and should reflect its use.



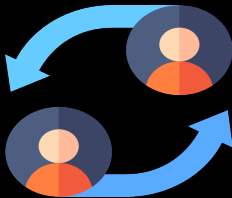
Displaying Multiple Items with the print Function

- Python allows one to display multiple items with a single call to print:
 - 1 Items are separated by commas when passed as arguments.
 - 2 Arguments displayed in the order they are passed to the function.
 - 3 Items are automatically separated by a space when displayed on screen.



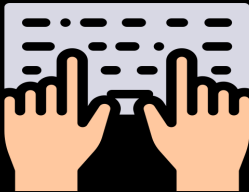
Variable Reassignment

- Variables can reference different values while program is running.
- Garbage collection: removal of values that are no longer referenced by variables. This process is carried out by Python interpreter.
- A variable can refer to item of any type. Variable that has been assigned to one type can be reassigned to another type.



Data Types and Literals

- Data types: categorize value in memory.
- **Example:** int for integer, float for real number, str used for storing strings in memory.
- Numeric literal: number written in a program. No decimal point considered int, otherwise, considered float.
- Some operations behave differently depending on data type.



Reassigning a Variable to a Different Type

- A variable in Python can refer to items of any type.



Reading Input from the Keyboard

- Most programs need to read input from the user.
- Built-in input function reads input from keyboard and Returns the data as a string.

Format: variable = input(prompt)

- **prompt** is typically a string instructing user to enter a value - Does not automatically display a space after the prompt.



Reading Numbers with the input Function

- **input** function always returns a string.
- Built-in functions convert between data types.

int(item) converts item to an int.

float(item) converts item to a float.

- Nested function call: Function call another Function.

Example - value returned by f2 is passed to f1.

general format: f1(f2(argument))

- Type conversion only works if item is valid numeric value, otherwise, throws exception.



Performing Calculations

- Math expression: performs calculation and gives a value.
- Math operator: tool for performing calculation.
- Operands: values surrounding operator.
Variables can be used as operands with the resulting value typically assigned to variable.



Performing Calculations (Cont'd)

- There are two types of division:
 - / operator performs floating point division.
 - // operator performs integer division.

What is the output of the following program?

```
x = 5
y = 2
print (x/y)
print (x//y)
```

Operator Precedence

- Python operator precedence: Operations enclosed in parentheses forces operations to be performed before others.
- The order of precedence is:
Exponentiation (**), Multiplication (*),
division (/ and //), remainder (%),
Addition (+) and subtraction (-).
- Higher precedence performed first.
- Operators with the same precedence get executed from left to right.



The Exponent & Remainder Operator

- Exponent operator (**): Raises a number to a power

$$x ** y = x^y$$

- Remainder operator (%): Performs division and returns the remainder This is also known as modulus or mod operator.

$$4\%2 = 0, 5\%2 = 1$$

- Modulus operator is typically used to convert times and distances, and to detect odd or even numbers.



Math to Programming Statements



- Operator required for any mathematical operation.
- When converting mathematical expression to programming statements. we may need to add multiplication operators and insert parentheses as required. For example:

$e(ab + cd) \rightarrow (e)*((a*b)+(c*d))$

Mixed-Type Expressions

- Data type resulting from math operation depends on data types of operands.
- The result of two float values is a float. While dealing with int and float, int temporarily converted to float resulting in a float.
- The result of two int values is an int.
- In Mixed-type expression, the type conversion of float to int causes truncation of fractional part.



Long Statements

- Long statements cannot be viewed on screen without scrolling and cannot be printed without cutting off.
- Hence, Multiline continuation character (\) is used to allow breaking a statement into multiple lines.

```
result = var1 * 2 + var2 * 3 + \  
var3 * 4 + var4 * 5
```



Long Statements (Cont'd)

- Any part of a statement that is enclosed in parentheses can be broken without the line continuation character.

```
print("Monday's sales are", monday,  
      "and Tuesday's sales are", tuesday,  
      "and Wednesday's sales are", Wednesday)  
total = (value1 + value2 + value3  
        + value4 + value5  
        + value6)
```

Lesson Summary

This chapter covered:

- Designing a Program.
- Input, Processing, and Output.
- Displaying Output with print Function.
- Comments.
- Variables.
- Reading Input from the Keyboard.



Things to do

- Complete Activity for Week-2.
- Do more progress in Assignment-1.
- Read Textbook Chapter-2.



Questions?

Please ask your Questions to clarify!