

COMP 141

CS1: Programming Fundamentals
Python language, output, variables



1

Announcements

- **Complete 1st zyBooks Assignment for next time**
 - Information on course website

2

Designing a Program

- **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

3

Designing a Program

- **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

4

Designing a Program

- **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: set of well-defined logical steps that must be taken to perform a task**

5

Pseudocode

- **Pseudocode: fake code**
 - Informal language that has no syntax rule
 - Not meant to be compiled or executed
 - Used to create model program
 - No need to worry about syntax errors, can focus on program's design
 - Can be translated directly into actual code in any programming language

6

Write an Algorithm

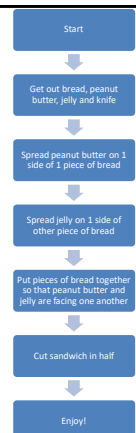
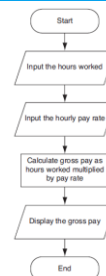
- **Work in groups of 3-4 and write down an algorithm for making a peanut butter and jelly sandwich**

7

Flowcharts

Flowchart: diagram that graphically depicts the steps in a program

Figure 2-2 Flowchart for the pay calculating program



8

Input, Processing, and Output

- Typically, computer performs three-step process
 - Step 1: Receive input
 - Input: any data that the program receives while it is running
 - Step 2: Perform some process on the input
 - Example: mathematical calculation
 - Step 3: Produce output

9

Displaying Output with the `print` Function

- **Function:** piece of prewritten code that performs an operation
- **Argument:** data given to a function
- Statements in a program execute in the order that they appear
 - From top to bottom
- **Example:**

```
- print ("Hello World")
```



Function that displays output on the screen



Data that will be displayed

10

Data Types

- Integers (int)
 - Whole numbers; may be negative.
- Floating point numbers (float)
 - Any number with a decimal point; may be negative.
- Strings
 - Any sequence of letters, numbers, or punctuation.
 - String literals are always surrounded by quotation marks, single or double.

11

Quick Demo



12

Activity

- **Open IDLE**
 - Start -> All Programs -> Computer Science and Math Programs -> Python 3.7.4-> IDLE (Python 3.7.4 GUI)
 - If you want to save this file for later use, in IDLE, go to File -> New Window and then Save that new file
 - Don't save it under Python, save it on your shared drive space or Box.com space!
 - To run your programs, go to Run -> Run Module (or hit the F5 key on your keyboard)
 - If you don't want to save the file, you can type the following statements using interactive IDLE mode
- **Write a statement that displays your name.**
- **Write a statement that displays "I am 100 years old!"**
- **Write a statement that displays the value of multiplying 322 * 35.**

13

Comments

- **Comments**: notes of explanation within a program
 - Ignored by Python interpreter
 - Intended for a person reading the program's code
 - Begin with a # character

14

Code with Comments

```
# Catie Welsh
# COMP 141
# This program display's a person's
# name and address.

print("Jane Doe")
print("123 Main Street")
print("Memphis, TN 38104")
```

15

Variables

- **Variable**: name that represents a value stored in the computer memory
 - Used to access and manipulate data stored in memory
 - 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 (=)

16

More on Variables

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

17

Variable Naming Rules

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

18

Displaying Multiple Items with the print Function

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

```
>>> dogName = "May"
>>> print("My dog's name is", dogName)
My dog's name is May
```

19

Numeric Data Types, Literals, and the str Data Type

- **Data types:** categorize value in memory
 - e.g., 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
- **Example:**

```
>>> a = 5           >>> a = "5"
>>> b = 7           >>> b = "7"
>>> print(a + b)   >>> print(a+b)
12                  57
```

20

Reassigning a Variable to a Different Type

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

```
>>> x = 90
>>> x = "Take me to your leader"
```
- **If you're using the same variable name for different uses, Python will assume you mean the most recent use**

21