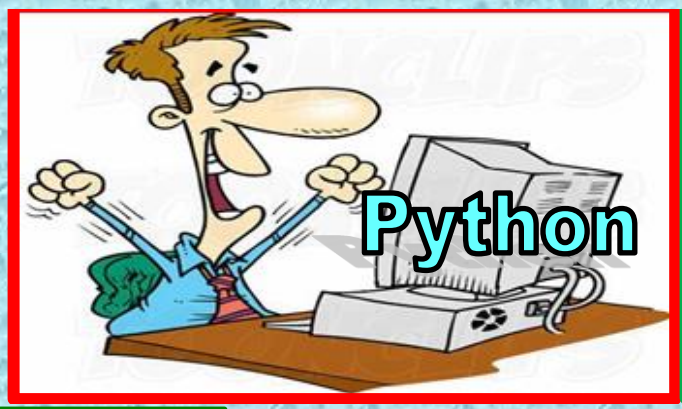
Wright College + Chapter 3

Conditional Statements: Using if...else and Logical Operators: ('and', 'or' & 'not') and Python Turtle Module

CIS103 Python Programming Language

Introduction to Computer Programming



'Hands-On' Mastering Computer Logic, Design and Programming Using Python Language

Written By:

OGAR HAJI

Master's Degree in Computer Science

DePaul University ✚ Chicago, Illinois, USA

Date Published: January 29, 2023

Wright College +

Chapter 3 Part A

Conditional Statements: Using if...else and Logical Operators: ('and', 'or' & 'not') and Python Turtle Module

CIS103 Python Programming Language

Introduction to Computer Programming

'Hands-On' Mastering Computer Logic, Design and Programming Using Python Language

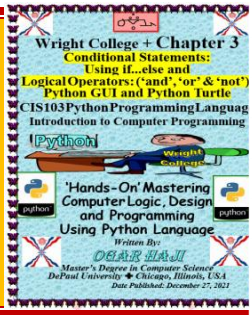
Written By:

OGAR HAJI

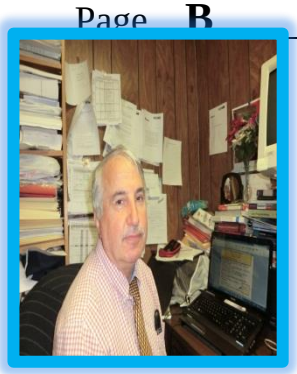
Master's Degree in Computer Science
DePaul University ✚ Chicago, Illinois, USA

Date Published: January 29, 2023

Instructor Hands-On Mastering Python Programming Language Chapters to Cover:

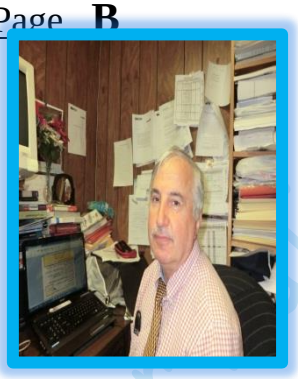


1) Chapter 01 + Introductions to Computers and Coding in Python Programming Language
2) Chapter 02 + Declaring Variables in Python and Turtle
3) Chapter 03 + Conditional Statements Using If Statement
4) Chapter 04 + Loops using For and While Statements
5) Chapter 05 + Introduction to Functions
6) Chapter 06 + Advanced Functions
7) Chapter 07 + Files and Exceptions
8) Chapter 08 + Python Lists and Tuples
9) Chapter 09 + Python Lists and 2D Lists
10) Chapter 10 + Python Strings and String Manipulation
11) Chapter 11 + Python Dictionaries and Sets
12) Chapter 12 + Object-Oriented Programming (OOP) in Pytho
13) Chapter 13 + Recursion Functions and Fibonacci Numbers
14) Chapter 14 + Graphical User Interface (GUI)
15) Chapter 15 + Data Base and Python



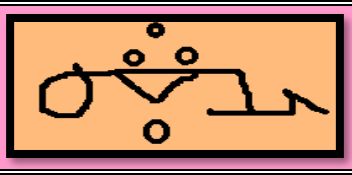
Computer Text Books Published by the Author: Ogar Haji

Page B



The Following is a List of Computer Text Books Published by the Author: Ogar Haji. He has an MS Degree in Computer Science from DePaul University, Chicago, Illinois - USA. Mr. Ogar Haji has over 30 Years of teaching experience at: The College of Office Technology, Oakton College, Washington College, Truman College, Wright College, Triton College, IIT Technical Institute, Phoenix University and East+West University in Chicago, Illinois, USA.

1) **'Hands-On' Mastering Microsoft Windows 10 and 7**
2) **'Hands-On' Mastering Microsoft Excel 2019 and 2016**
3) **'Hands-On' Mastering Microsoft Word 2019 and 2016**
4) **'Hands-On' Mastering Microsoft Access 2019 and 2016**
5) **'Hands-On' Mastering Microsoft PowerPoint 2019 & 2016**
6) **'Hands-On' Mastering Microsoft Publisher 2010**
7) **'Hands-On' Mastering MS Visual Basic .Net Language**
8) **'Hands-On' Mastering Java Programming Language**
9) **'Hands-On' Mastering HTML5 and CSS3 Web Page Design**
10) **'Hands-On' Mastering JavaScript Programming Language**
11) **'Hands-On' Mastering Ruby Programming Language**
12) **'Hands-On' Mastering QBasic Programming Language**
13) **'Hands-On' Mastering DOS (Disk Operating System)**
14) **'Hands-On' Mastering C# Programming Language**
15) **'Hands-On' Mastering Python Programming Language**



Ogar Haji (CIS Adjunct Instructor)
Wilbur Wright College
4300 N. Narragansett Avenue
Chicago, Illinois 60634
USA
www.ccc.edu/Wright

Computer Labs Rules

1- **No Drinks, Food, Headphones** allowed in Computer labs. And Please **Turn Off the Cell Phones**.

2- When **Lecturing** is in progress, you are Not allowed to work on the computer. Please **Pay Attention** and **Take Notes**.

3- **Attendance and Punctuality** are very **important**. If you are absent, it is your responsibility to make up for the missing work and assignments. **Attendance will be taken daily.**

4- Students should have a **USB Flash Drive** and **Save Projects** to it.

5- **Practice makes perfect**. Please keep practicing the new features or steps over and over again until the instructor tells you to stop.

6- You have to **Concentrate** on what you are doing. **Talking is Not Allowed** in the computer Lab.

7- Please **Study** the Lessons in your **Python Handout and Text Book Daily** and **review your notes before class**. **There will be a Quiz Once a Week**.

8- Please **Check Mark ☑ the Lessons** in the Handout that you **have completed.**

9- You **must do All Python works**, **Assignments and Tests** located at the End of each Chapter on Time.

CIS 103 Python Programming

Instructor: Ogar Haji

Chapter 3+Part A

Conditional Statements: Using if..else and Logical Operators('and','or' & 'not') Calculate Final Grades Project and Insert Computer Date and Time project

You will learn the following in Chapter 3 Part A:

- Named Constants in Python: SALES_TAX_RATE = 0.09
- Using Multiplication Operator (*) to print letter 'Z' 25 Time
- Using max() and min() functions to find the Maximum Number and the Minimum Number.
- Using Single **if** statement to Check if a Condition is **true.**
- Using Single **if…else** statements to Check if a Condition is **true** or **false** in Python
- Using the Short-Circuit Conditional 'and' and 'or' operators in Python Language.
- Calculate the Average of 2 Tests and Assign Final Grade. Lab Exercise
- Insert Computer Date and Time in Python + Lab Exercise.
- Calculate Monthly Payment of a Loan Python Project
- Python Language to Draw a Square with Colored Lines (red, green, blue, gold)
- Chapter 3 + Python Homework #3 (Due Next Week)
- Do Python Lab Assignments 3A + American Cars Rent

Flowchart Symbols

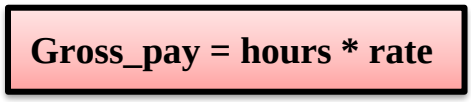
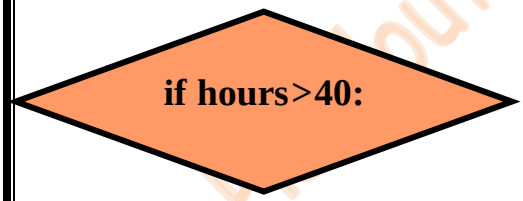
+++ Review +++

Lesson 100 Review + What are the Flowchart Symbols used in Python Language?

You should always **draw a Flowchart** when you Design, Code and Solve a problem in Python language.

Before you Code a program in Python Language, you have to **Draw a Flowchart** to solve the problem of the program you want to code.

The following symbols are used with Python Programming Language:

Symbol	Symbol Name	Usage
	**Oval (Beginning and Terminal)** symbol	Use **Oval (Beginning and Terminal) Symbol** at the Beginning of the Flowchart and at the End of the Flowchart. Use with Start and End statements.
	**Parallelogram (Input/Output)** Symbol	Use **Parallelogram (Input /Output)** or **I/O** symbol to Input Data, Read Input or Print Output
	**Rectangle** Symbol	Use the **Rectangle** Symbol for Calculating, Assigning Values
	**FlowLine** Symbol	Use **FlowLine** Symbols to show the Flow or Sequence of the flowchart.
	**Diamond (Decision)** Symbol	Use **Diamond (Decision)** Symbol with the If or Select statements when deciding if Hours is > 40. The Result will be either True or False.
	**Connector** Symbol	Use **Connector** Symbol to Connect the Flowchart rather than draw a long Arrow. Use 
	**Function or Method (Predefined Process)** Symbol.	Use **Function** or **Method (Predefined Process)** Symbol to call another Function or Method that contains coding statements.

Calculate Gross Pay of Employees Project

+++ (Do Lab Exercise) +++

+++ Review +++

Lesson 101 Review + How to Calculate Gross Pay of Employees Project?

Problem or Project: Design and Code in Python Language the project to Calculate Gross Pay of Employees in a company.

Do the following 12 Must Steps to Design, Code and Solve a project using Python Language.

Do Steps 1 thru 7 in your Note Book or on Paper.

Step 1) Purpose of the Program: State what Program will do: (5 Points)

a) This Program will calculate Gross Pay of Employees.
b) It will ask the User to Enter Employee's Full Name:
c) It will ask the User to Enter Number of Hours Worked and
d) It will ask the User to Enter Hourly Rate.
e) The program will then calculate Gross Pay.

Gross Pay = Hours * Rate

f) Display the Gross Pay

Step 2) Input: You should know how the Input looks like: (5 Points)

Enter Employee's Full Name: Ogar Haji
Enter Hours Worked: 40
Enter Hourly Rate: 10

Step 3) Processing and Calculation: The program will process each record and Calculate Gross Pay: (5 Points)

Gross Pay = Hours * Rate

Step 4) Output: You should know how output should look like: (5 Points)

********** Calculate Gross Pay Project **********

The Employee Ogar Haji Gross Pay is = $ 400

********** End of Program **********

Step 5) Flowchart: Draw a Flowchart for Gross Pay program.(5 Points)

Step 6) PseudoCode: print a PseudoCode for the Program.(5 Points)

1) Declare the variables to be used in the program
2) print to Console 'The Title or Heading of the program '
3) Prompt the user 'Enter Employee's Full Name: '
4) Read from Console full_name and store the String in a variable
5) Prompt the user 'Enter Hours Worked: '
6) Read from console hours worked and store in a String variable
7) Convert String hours variable to integer value
8) Prompt the user 'Enter Hourly Rate: '
9) Read from console the rate and store in a String variable
10) Convert String rate variable to float value
11) Calculate Gross Pay = Hours * Rate
12) print to Console 'The Employee Gross Pay is = $', gross_pay
13) print to Console '******* End of Program *******'

Step 7) Code the Program in Python by referencing the Flowchart or Pseudocode you designed above and Save it as CalculateGrossPay to USB.

a) Type the following Python code in IDLE IDE: c File, New File

Code for the first part of 'CalculateGrossPay' project follows:

```
# 1) Add Comments about the Calculate Gross Pay Project
'''
 ************************************************************
   Purpose of the Project:
   a) This Interactive Project will prompt the user to enter his/her (First Name,
   Last Name, and Hours Worked and Rate) then it will read the text entered
    and  store it in its variables. Then it Calculates the Gross Pay.
   b) Project Name: CalculateGrossPay
   c) Date: Saturday, December 28, 2016
   d) Programmer: Instructor – Ogar Haji
 ************************************************************
'''
# 2) Prompt the user to Enter his/her First Name and store it in a variable
first_name = input ('Enter your First Name: ')
# 3) print or echo First Name to the output
print ('Your First Name is: ', first_name)
# 4) Prompt the user to Enter his/her Last Name and store it in a variable
last_name = input ('Enter your Last Name: ')
```

5) print or echo Last Name to the output
print ('Your Last Name is: ', last_name)

6) Prompt the user to Enter Hours Worked and Store it in a variable
hours = input ('Enter Number of Hours Worked: ')

7) Convert the string hours to integer using int() function
hours = int (hours)

8) print or echo Hours Worked to the output
print ('Hours Worked: ', hours)

9) Prompt the user to Enter Hourly Rate, read it and store it
rate = input ('Enter Hourly Rate: ')

10) Convert the rate to float using float() function
rate = float (rate)

11) print or echo Hourly Rate to the output
print ('Hourly Rate: ', rate)

12) Calculate Gross Pay
gross_pay = hours * rate

13) print grossPay to the output screen using .format () function
print ('Gross Pay is = $ {:,.2f}'.format(gross_pay))

Step 8) Click **Run Project**  button to Start Running the program

The following output appears on the Left side of the screen with the Input you entered and the correct calculated GrossPay $400.

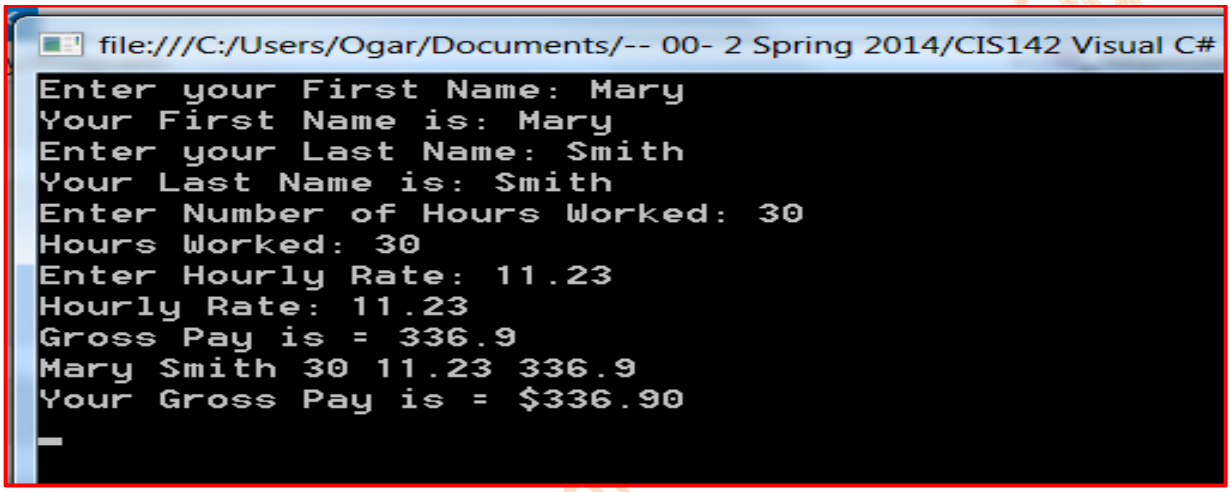
If any Syntax Errors Found Do Next Step 9:

```
Enter your First Name: Ogar
Your First Name is:  Ogar
Enter your Last Name: Haji
Your Last Name is:  Haji
Enter Number of Hours Worked: 40
Hours Worked:  40
Enter Hourly Rate: 10
Hourly Rate:  10.0
Gross Pay is = $ 400.00
```

Step 9) Debug the Program: **Debug or Correct any Syntax Errors until you have a clean Compiled program. (5 Points) (Clean compiled program means No Errors in the program).**

Step 10) Test the Program: **Test the Program with Test Data. (5 Points)**

Repeat Step 10) Test the program many Times and Test the Program again and again until All conditions are tested:



Step 11) Documentation (5 Points): You have to add more comments to the Program (like Comments about the Purpose of the Program, Your Name and the Date the Program was written.)

```
##################################################################
# Purpose of the Program:
#   a) This Program will calculate Gross Pay.
#   b) It will ask the User to Enter Employee's Full Name:
#   c) It will ask the User to Enter Number of Hours Worked
#   d) It will ask the User to Enter Hourly Rate.
#   e) The program will calculate Gross Pay.
#       Gross Pay = Hours * Rate
#   f) Display Gross Pay
##################################################################
```

Step 12) Print a Copy of Python Code along with screen printout of the Running program. Submit to your Instructor the Print Copy and the screen Printout (Snaps) along with the following: (Which you did on Paper)

Copy the Python Code and the result of the program and Paste it in Microsoft Word program:

1) Purpose of the Program.
2) Input: how the Input looks like
3) Processing and Calculations
4) Output: how the Output will look like
5) Flowchart 6) Pseudocode
7) Python Code and
8) Print out copy of Python code and Output after running the program.
Submit the Programs on Time.
Remember Points will be deducted (20%) for Programs submitted Late.

Important Note:
1) Do Steps 1 thru 7 on Paper.
2) Then Get into Python IDLE IDE Code Editor
3) Type the Python code.
4) Save All the Files
5) Run the Program and Test it with Test Data for All Conditions.

Modify the Project and add the following to print to the Console using Format Specifiers (%s, %d, %,.2f, %n) with print() method.

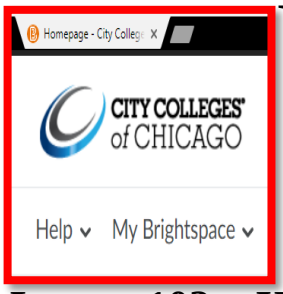
To print the Results using Format Specifiers: %s %d %f %.2f %n

```
#Using Format Specifiers (%s, %d, %f, %,.2f) to print out to output
  print ('%s       %s       %d    %f   $%,.2f ' % (
          first_name, last_name, hours, rate, gross_pay)  )
```

or

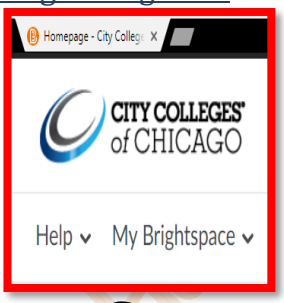
```
# print the Formatted Gross Pay to output console using .format () function
print ('Gross Pay is = $ {:,.2f}'.format( gross_pay) )
```

When Modifying a Project
Do Only 1 Modification at a Time



Upload Your Weekly Assignments: To Brightspace Correct Weekly Assignments Folder

+++ Review +++



Lesson 102 + How to Upload Your Weekly Assignments to Brightspace Correct Weekly Assignments Folder?

You must Upload your Weekly Assignments and Homework to Brightspace correct Weekly Assignments Folder as following:

1) Copy the Python Code from Python IDLE to Word document:

1. Copy the **Python Code from** the **Python IDLE IDE** and **Paste it into** the **Microsoft Word Document.**
2. In IDLE IDE, press **Ctrl+A (select All)** to **select All** the **Python code**.
3. Press **Ctrl+C (Copy)** to **Copy** the **selected Python code** into **computer memory RAM.**

2) Paste the Python Code into Microsoft Word:

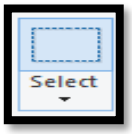
1. Get into **Microsoft Word document then** press **Ctrl+V (paste) to Paste** the **copied Python code** from **memory** into **Word document.**
2. Press **Ctrl+Home (go to the Top of Document)** and **type** your **Full Name** at the **top of document followed by** the **Python File Name** in **size 20 and bold.**

3) Print the Screen of the Output of Python Python IDLE:

1. Run the **Python project** and **make sure the program** is **running** with **correct output.**
2. Press **PrintScreen button** 

to **capture** the **output screen shot.**

4) Paste the Print Screen of Python output into Paint progra

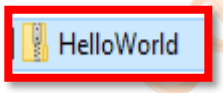
1. Get into **Paint program** and press **Ctrl+V (paste) to Paste** the **screen shot** in **Paint** program.
2. Inside the **Paint program,** Click **Select** 

icon and then **Select only** the **Output** of the **Python project.**
3. Press **Ctrl+C (Copy)** to **Copy** the **selected output image.**

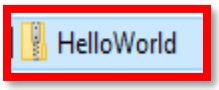
5) Get back into Microsoft Word program:

1. Go back to **Word Document,** press **Ctrl+End (End of Document)** to **go to End of document.**
2. In the **Word document,** press **Ctrl+V (Paste)** to **paste** the **Python output** there.
3. Save the **Word Document** as the **Name of the Python project** and in this **example (Save File as HelloWorld project)**

6) To Compress or Zip the Python project:

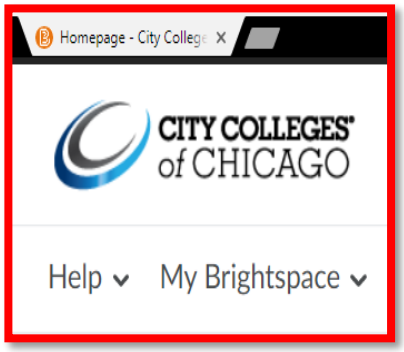


1. Right-Click on the **Python project (HelloWorld)** that is **saved on your computer**.
2. Point to **Send to,** then **click on Compressed (Zipped) Folder** and you will have **another File** which is Compressed or Zipped.



7) To Upload the 2 Files to Brightspace:

1. Log on Brightspace with your **User Name** and **Password**
2. Click on your course **CIS 103 Python** to **Select it**.
3. Click on **Assignments**
4. Click on **Assignments**
5. Click on **Week 03 Assignments Folder**
6. Click on **'Add a File'** button
7. Click on **'My Computer'**
8. Click on **Upload** button



9. Go to the **location** where you **saved the Python project 'HelloWorld'.**
10. Click on the **File** or **Folder (HelloWorld)**
11. Click on **Add** button and the **File** or **Folder** will be **added** to the **Week 01 Assignments Folder.**

Note: Always **Upload** to **Brightspace the Modified Python Project:**
1) The **Microsoft Word Document** of the **Python Project Code along with the Python Output Screen shots.**

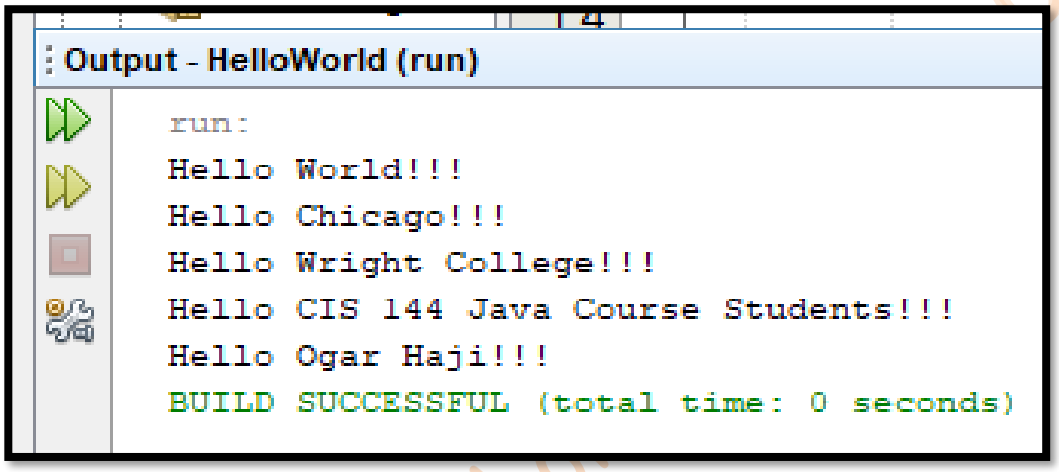
The Python project 'HelloWorld' code in Word Document along with the Output Screen Shots appear as following:

Ogar Haji (Your Full Name)

CIS 103 Python

Project Name: HelloWorld

(Always Place the Screen Output at the Top Before Python Code)



'''

Project Name: HelloWorld

This Python project will print the message 'Hello World' to screen

Programmer: Instructor + Ogar Haji (Type your Full Name)

Date: June 01, 2017

'''

This project will print the Literal String 'Hello World' to output screen

print ('Hello World!!!')
print ('Hello Chicago!!!')
print ('Hello Wright College!!!')
print ('Hello CIS 103 Python Course Students!!!')
print ('Hello Ogar Haji!!!')

print ('Title of program')

Printing Program Title and 'End of Program' Message

print ('End of program')

Lesson 103 +How to use print() function to print Title/End of Program messages?

print() built-in function is used to **display a string** on the **screen.** A **String** is any **Text enclosed between ' Single Quotation'** or **'' Double Quotation''**:

Keywords like **print** will be displayed in **Orange** Color.
Strings like **'**** Computer Books Store Program ***** '** will be displayed in **Green** Color.

```
print ('**** Computer Books Store Program ***** ')
print ('********   End of Program   ********')
```

```
******** Computer Books Store Program ********
Book Title: Introduction to Computer Programming
Book Price: 50.00
Number of Books Purchased: 10
Purchase Amount is : $ 500.00
********** End of Program **********
```

To print out the Purchase Amount:

```
print ('Purchase Amount is: $', purchase_amount)
```

print( )

print statement: print a Blank Line

print( )

Lesson 104 + How to use the print function to print a Blank Line ?

Use **print()** built-in function by itself with **no arguments** to print a Blank Line:

```
print ()  #print function with no Arguments will print a Blank Line
```

title =input ('Enter the title: ')

input() Function:

book_title = input ('Enter Book Title: ')

title =input ('Enter the title: ')

Lesson 105 + How to use the input( ) function to read (get) Strings or Text typed on the Keyboard ?

Use **input ()** built-in function to **prompt the User to Enter** the **Title of the Book** and also to **get** or **read the string or text or Numbers** the **user will input** or **type on the keyboard**:

Functions like **input()** will be displayed in **Purple** Color.

Use **input ()** function and **the prompt together**

```
book_title = input ('Enter Title of the Book:  ')
```

Python Language

book_title

price =input ('Enter price: ')

input() Function:

price = input ('Enter the Price of the Book: ')

price = float (price)

Lesson 106 + How to use the input function to read (get) String Numbers typed by User on the Keyboard?

Use **input ()** built-in function to prompt the user to Enter the Price of the Book and to **get** or **read the string float Number** the **user input** or **type** on **keyboard**:

Functions like **input()** will be displayed in **Purple** Color.

Use **input()** function and **the prompt together.**

21.88

price

Then use **float ()** **function** to **convert** the **string price** to **float**.

Use **int ()** **function** to **convert** the **string number_of_books** to **int**.

```
price = input ('Enter the Price of the Book: ')
price = float (price)
```

```
number_of_books = input ('Enter Number of Books Purchased')
number_of_books = int (number_of_books)
```

DISCOUNT_RATE = 0.25

Named Constants in Python: SALES_TAX_RATE = 0.09

SALES_TAX_RATE = 0.09

Lesson 107 + How to Declare Named Constant and Initialize its Value?

Named Constant Identifiers are **variables** that are **set** or **initialized to a Value** and **that Value can Not be Changed during the execution** of the **program**.

It is customary for Python programmers to write **Named Constants Variables in Upper Case and _ underscore to separate words**.

Example of **Named Constant Variables:**

DISCOUNT_RATE = 0.25

SALES_TAX_RATE = 0.09

PASSING_SCORE = 70

SALES_TAX_RATE = 0.09

To use the **Named Constant Variables in Calculations:**

```
discount_amount = purchase_amount * DISCOUNT_RATE
sales_tax = sale_amount * SALES_TAX_RATE
```

print('*'*50)

Using Multiplication Operator (*): To print letter 'Z' 25 Times

print('Z'*25)

Lesson 108 + How to use (*) Multiplication Operator to Repeat a String?

You can use the **Multiplication Operator (*)** to **repeat** a **String many times**:

```
>>> print ('*' * 50)      # will print the '*' 50 times
**************************************************
```

```
>>> print ('Z' * 25)     # will print the 'Z' 25 times
ZZZZZZZZZZZZZZZZZZZZZZZZZ
```

```
>>> print ('Python is Fun. ' * 7)  # will print the 'string' 7 times
Python is Fun. Python is Fun. Python is Fun. Python is Fun. Python is Fun. Python is Fun. Python is Fun.
```

```
>>> print ('-' * 51)              # will print the '_' 51 times
---------------------------------------------------
```

max (11, 44, 33)

max() function to find Maximum Number: max (11, 44, 33, 99, 77)

max (11, 44, 33)

Lesson 109 + How to use max() function to Find the Maximum Number in a Range of Numbers?

Use **max()** built-in function to **find** the **Largest** or **Maximum Number** in a Range of Numbers.

max (22, 11, 99, 77)

```
>>> max (11, 44, 33, 99, 77)
99
>>> max (20, 10, 5, 33, 41, 12, 7, 28)
41
```

min (11, 44, 33)

min() function to find Minimum Number: min (22, 14, 44, 33, 99, 77)

min (11, 44, 33)

Lesson 110 + How to use min() function to Find the Minimum Number in a Range of Numbers?

Use **min()** built-in function to **find the Lowest** or **Minimum Number** in a Range of Numbers.

min (22, 11, 99, 77)

```
>>> min (22, 14, 44, 33, 99, 77)
14
>>> min (20, 10, 5, 33, 41, 12, 7, 28)
5
```

max (test1, test2, test3)
min (test1, test2, test3)

Lesson 111 + How to use max(), min() functions with Variables?

You can also **use max()** and **min()** **functions on Variables that have been assigned values**.

```
>>> test1 = 89
>>> test2 = 93
>>> test3 = 75
>>> max (test1, test2, test3)
93
>>> min (test1, test2, test3)
75
```

max (test1, test2, test3)

min (test1, test2, test3)

help (input)

Using Python help() function to get help: help (input)

help (max)

Lesson 112 + How to use help command to get help on Python Functions?

Use **help()** built-in function in Interactive Python **to get help on the functions used in Python Language**.

```
>>> help (input)
Help on built-in function input in module __builtin__:

input(...)
    input([prompt]) -> string

    Read a string from standard input.  The trailing newline is stripped.
    If the user hits EOF (Unix: Ctl-D, Windows: Ctl-Z+Return), raise EOFError.
    On Unix, GNU readline is used if enabled.  The prompt string, if given,
    is printed without a trailing newline before reading.
>>> help (input)
Help on built-in function input in module __builtin__:

input(...)
    input([prompt]) -> value

    Equivalent to eval(input(prompt)).
>>> help (max)
Help on built-in function max in module __builtin__:

max(...)
    max(iterable[, key=func]) -> value
    max(a, b, c, ...[, key=func]) -> value

    With a single iterable argument, return its largest item.
    With two or more arguments, return the largest argument.
>>> help (min)
Help on built-in function min in module __builtin__:

min(...)
    min(iterable[, key=func]) -> value
    min(a, b, c, ...[, key=func]) -> value

    With a single iterable argument, return its smallest item.
    With two or more arguments, return the smallest argument.
```

help (input)
help (max)
help (min)

```
>>> help (input)
>>> help (max)
>>> help (min)
```

help (input)
help (max)
help (min)

```
Python 3.9 (64-bit)
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> help (input)
Help on built-in function input in module builtins:

input(prompt=None, /)
    Read a string from standard input.  The trailing newline is stripped.

    The prompt string, if given, is printed to standard output without a
    trailing newline before reading input.

    If the user hits EOF (*nix: Ctrl-D, Windows: Ctrl-Z+Return), raise EOFError.
    On *nix systems, readline is used if available.

>>> help (max)
Help on built-in function max in module builtins:

max(...)
    max(iterable, *[, default=obj, key=func]) -> value
    max(arg1, arg2, *args, *[, key=func]) -> value

    With a single iterable argument, return its biggest item. The
    default keyword-only argument specifies an object to return if
    the provided iterable is empty.
    With two or more arguments, return the largest argument.

>>> help (min)
Help on built-in function min in module builtins:

min(...)
    min(iterable, *[, default=obj, key=func]) -> value
    min(arg1, arg2, *args, *[, key=func]) -> value

    With a single iterable argument, return its smallest item. The
    default keyword-only argument specifies an object to return if
    the provided iterable is empty.
    With two or more arguments, return the smallest argument.

>>> _
```

help (input)
help (max)
help (min)

Python IDLE uses Colors Coding Scheme

Lesson 113 + How does Python IDLE display Python Code ?

IDLE displays the Python Code in different colors which is a great help to the programmer.

Important Colors Coding scheme used in IDLE environment:
 # Comments appear in **Red** color.
Key words: True appears in **Orange** color.
Strings 'Enter the User Full Name: ' appear in **Green** color.
Functions: print (), input () appear in **Purple** color.
Variables: full_name, hours, rate appear in **Black** color.
Output of the program will appear in Blue color.

Python Syntax Errors: Misspelled Python Functions or Misspelled Variable Names

Lesson 114 + How to Debug a Python language Program ?

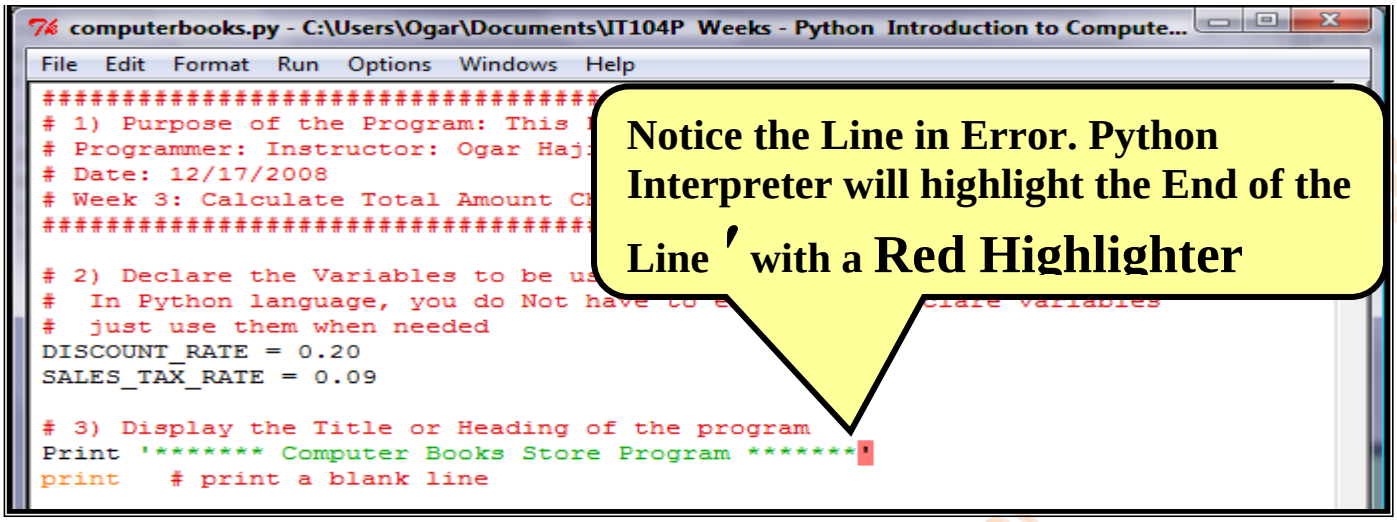
Debug the Program: **Debug** or **Correct any Syntax Errors until you have a clean interpreted (translated) program.**
Clean interpreted program means No Syntax Errors in program.
Always use lower case letters in Python language.
Examples: **print(), input(), book_title, price**.
In the following statement the programmer typed the Print statement in Upper Case letter. NO spaces Allowed in a variable name.

```
Print ('******* Computer Books Store Program *******')
print ()        # prints a blank line
```

So **Python Interpreter did Not recognize** it as a **Keyword 'print', so it was flagged as Invalid Syntax and displayed this Syntax Error dialog box.**

```
== RESTART: C:/Users/ogarh/AppData/Local/Programs/Python/Python310/Print syntax er
Traceback (most recent call last):
  File "C:/Users/ogarh/AppData/Local/Programs/Python/Python310/Print syntax errors
e 1, in <module>
    Print ('******* Computer Books Store Program *******')
NameError: name 'Print' is not defined. Did you mean: 'print'?
```

Correct the Error by typing the print statement in lower case letter.



Python Logical Errors: Wrong Calculations (using + instead of *)

Lesson 115 + How to Debug a Python Program for Logical Errors ?

With **Logical Errors** the **program will run but it will produce the wrong result**.

To calculate **discount_amount**, the programmer should **Multiply purchase _amount by DISCOUNT_RATE**. Instead, the programmer used **the – (Subtracted)**

discount_amount = purchase_amount - DISCOUNT_RATE

To correct the Logical Error, delete Minus Sign – and type *

discount_amount = purchase_amount * DISCOUNT_RATE

```
= RESTART: C:/Users/ogarh/App
ate.py
Today's Date is:  2021-09-06
>>>
```

Insert Computer Date and Time In Python Project

```
= RESTART: C:/Users/ogarh/App
ate.py
Today's Date is:  2021-09-06
>>>
```

+++(Do Lab Exercise 1) 100 Points+++

Do Lab Exercise 1

Lesson 116 Ex + How to Insert Computer Date and Time using Python?

Problem or Project: Design and Code in Python Language the project to Display Current Date and Time to the screen.

It is customary in programming to Display and Print Date and Time in Python language at the top of the output.

The **strftime()** (**String Format Time**) method **returns** a **string representing date** and **time** using **date**, **time** or **datetime objects**.

To Insert Computer Date and Time in a project,

```
= RESTART: C:/Users/ogarh/App
hon/Python311/PrintDateAndTim
Today's Date is:  2023-01-20
```

 # 1) Import the following **date classes**:

```
from datetime import date
```

 # 2) Create an instance **object** called '**today**' from the module **date.today()**

```
Today's Date is =  2021-09-07
Current Date and Time: = 2021-09-07 14:34:19.176979
Date and Time Formatted = 07/09/2021 14:34:19
```

```
today = date.today ()
```

 # 3) Print out **today** to output console, use **\'s**

```
print ('Today\'s Date is: ', today)
```

```
= RESTART: C:/Users/ogarh/App
hon/Python311/PrintDateAndTim
Today's Date is:  2023-01-20
```

This is an Explanation of What we will Do in this project:

 # 1) To Insert **Computer Date and Computer Time**, import the following **classes**:

from datetime import date

 # 2) Create an instance **object** called '**today**' from the class **date.today()**

today = date.today ()

```
Today's Date is =  2021-09-07
Current Date and Time: = 2021-09-07 14:34:19.176979
Date and Time Formatted = 07/09/2021 14:34:19
```

 # 3) Print out **today** to output console

print ('Today\'s Date is: ', today)

```
Today's Date is =  2021-09-07
Current Date and Time: = 2021-09-07 14:34:19.176979
Date and Time Formatted = 07/09/2021 14:34:19
```

 # 4) Import datetime from datetime

from datetime import datetime

 # 5) Create an instance object called '**now**' from the module **datetime.now()**

now = datetime.now ()

 # 6) print the Current Date and Time

print ('Current Date and Time: =', now)

```
Today's Date is =  2021-09-07
Current Date and Time: = 2021-09-07 14:34:19.176979
Date and Time Formatted = 07/09/2021 14:34:19
```

 # 7) Format the Date and time as **dd/mm/YY H:M:S**

date_and_time = now.strftime ('%d/%m/%Y %H:%M:%S')

 # 8) print the Formatted Date and Time

print ('Date and Time Formatted =', date_and_time)

+ Do Lab Exercise 1 + Do Lab Exercise 1 +

1) Type the following project in Python IDLE and

2) Save Python File as DateAndTime.py

Do Lab Exercise 1

```
'''
This project will print out the Date and Time on the screen
Date: 07/21/2018
```

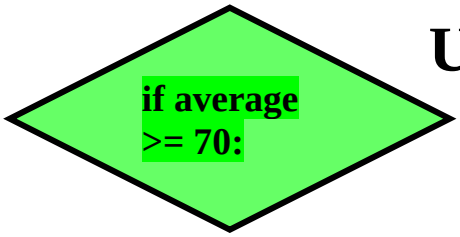
```
Programmer: Ogar Haji (Python Instructor)
'''
# 1) Import date from datetime class:
from datetime import date
# 2) Create an instance object called 'today' from the class date.today()
today = date.today()
# 3) Print out the date to out console
print ('Today\'s Date is: ', today)
# 4) Import datetime from datetime class
from datetime import datetime
# 5) Create an instance object called 'now' from the class datetime.now()
now = datetime.now()
# 6) print the Current Date and Time
print ('Current Date and Time: =', now)
# 7) Format the Date and time as m/d/Y H:M:S
date_and_time = now.strftime ('%m/%d/%Y %H:%M:%S')
# 8) print the Formatted Date and Time
print ('Date and Time Formatted =', date_and_time)
```

```
= RESTART: C:/Users/ogarh/App
Today's Date is:  2021-09-07
>>>
```

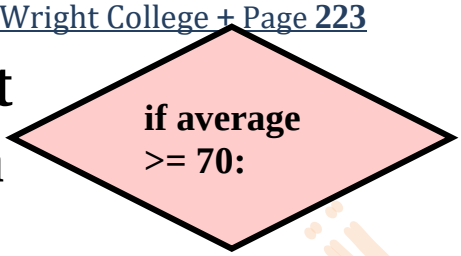
```
Today's Date is =  2021-09-07
Current Date and Time: = 2021-09-07 14:34:19.176979
Date and Time Formatted = 07/09/2021 14:34:19
```

The Output of this Python project will look like the following:


```
Today's Date is:  2023-01-09
Current Date and Time: = 2023-01-09 13:24:15.782068
Date and Time Formatted = 01/09/2023 13:24:15
```



Using Single if statement To Check if a Condition is True in Python



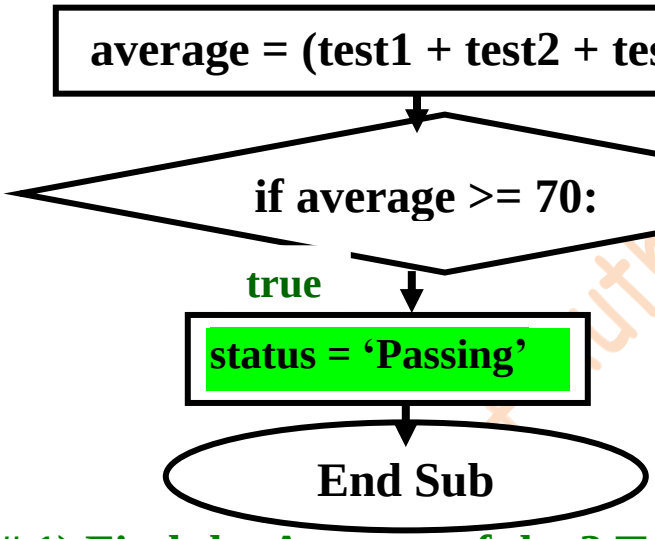
Lesson 117 + How to Use Single If statement to check if a Condition is True or False?

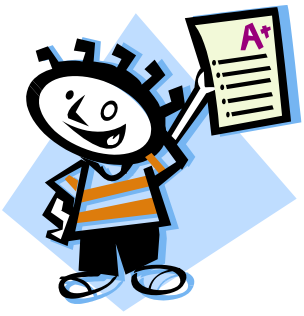
Using Single If statement to Check if a Condition is True:

You can use a **Single if statement to check for a condition if true**, you will **execute a statement or series of statement**, **otherwise you will do Nothing**.

It is **recommended to use Curly Braces { } to enclose the true statements** and the **false statements even if you have only 1 statement**.

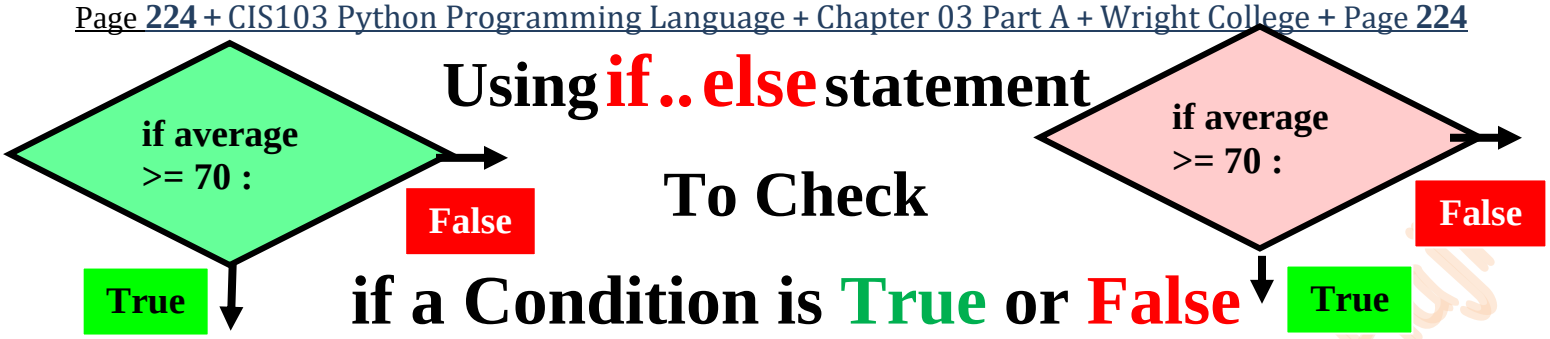
The following Flowchart is a portion of Calculate Average and then using if statement to check if average is >= 70 is true then only assign to status the String 'Passing', otherwise do nothing.






```
# 1) Find the Average of the 3 Tests
   average = (test1 + test2 + test3) / 3
# 2) Find if the student is Passing (If average is Greater or Equal to 70)
   if average >= 70:
       status = 'Passing'
# 3) Print out the Student Average and Status to output
print ('Student Average = %d and is %s' % (average, status) )
```

```
== RESTART: C:/Users/ogarh/AppData/L
Student Average = 95 and is Passing
```

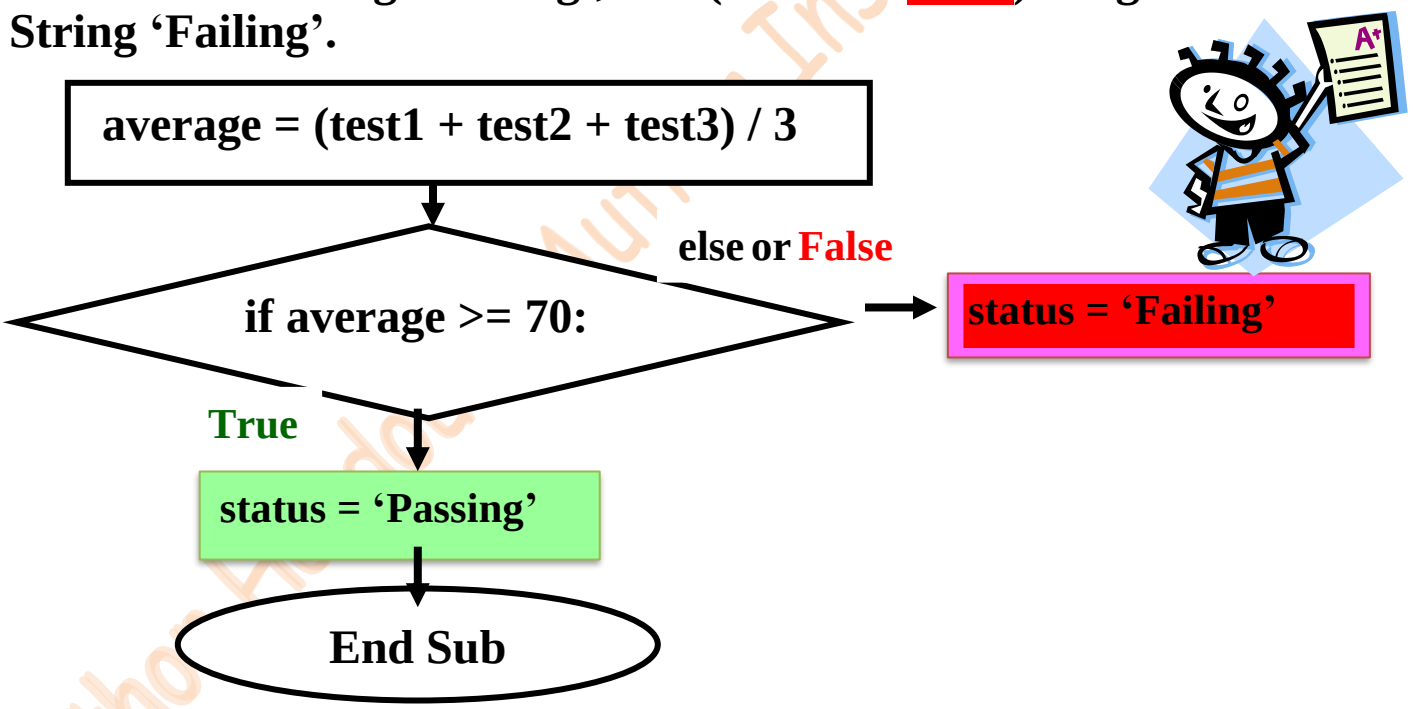


Lesson 118 + How to Use If - else statement to check if a Condition is True or False?

Using If - else statement to Check if a Condition is True or False:

You can use if statement to check whether a Condition is True or False.
For example, a student takes 3 Test, then I will calculate the Average and then use If statement to check if average is Greater or Equal to 70, then the student is Passing else the student is Failing.

The following Flowchart is a portion of Calculate Average and then using if - else statement to check if average >= 70: is True then assign to status the String 'Passing', else (means if False) assign to status the String 'Failing'.



This is also the Pseudocoe:

1- Calculate Average of the 3 Tests
2- Use if to check if average is Greater Equal to 70 if average >= 70: then student is Passing, else student is Failing.
3- Print using print() function the **student average** and the **status.**

The following is the Python Code:

```
# 1) Declare and initialize the variables used in the project
test1 = 100
test2 = 99
test3 = 98
# 2) Find the Average of the 3 Tests
average = (test1 + test2 + test3) / 3
# 3) Find out if the student is Passing or Failing
if average >= 70:
    status = 'Passing'
else:
    status = 'Failing'
# 4) print out the Student Average and Status to output
print ('Student Average = %d and status is %s' % (average, status) )
```

or

```
print ('Student Average = {0} and status is {1} % (average, status) )
```

```
Student Average = 99 and status is Passing
Student Average = 99.0 and status is Passing
```

```
>>> # To Find if student is Passing or Failing
>>> average = 95
>>> if average >= 70:
...     print ('The Student is Passing')
... else:
...     print ('The Student is Failing')
...
The Student is Passing
```

In Python Programming Language, make sure the statements within **if statement,** are **Indented with the same Number of Space Indentation**.

Please, Read, Study and Practice the Lessons in the Python Handout

Using if - else statement to Check if a Condition is True or False to Calculate Gross Pay using Overtime

Lesson 119 + How to Use If - else statement to check if a Condition is True or False?

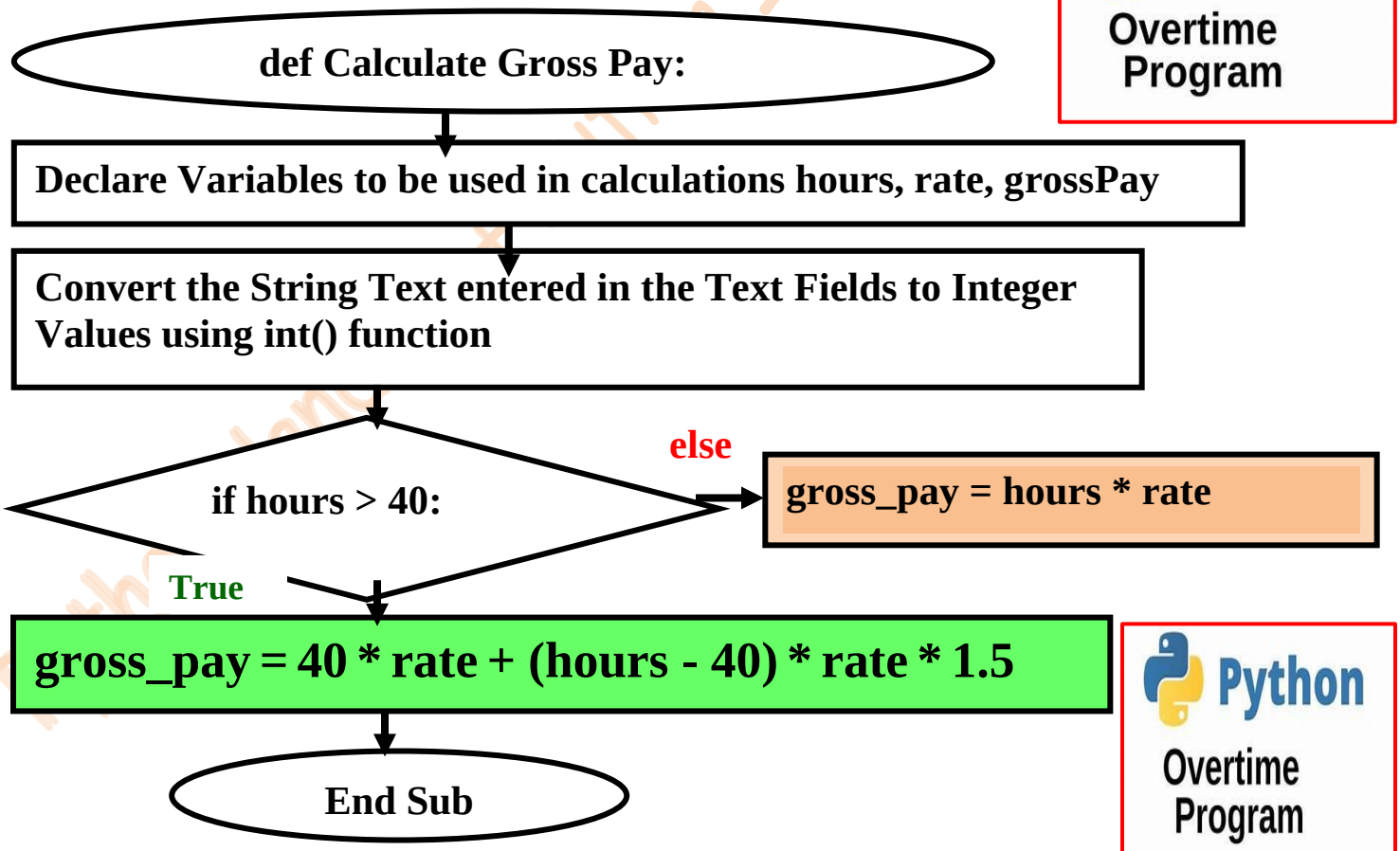
Using if - else statement to Check if a Condition is True or False:

You can use if statement to check whether a Condition is true or false.

For example, to check if an employee worked overtime (**hours > 40**), then calculate overtime gross pay and also display the message 'Overtime', else just calculate gross pay without overtime and display a blank message for overtime message.

The following Flowchart is a portion of Calculate Overtime Gross Pay and then using if - else statement to check if hours worked >40 is true then do overtime grosspay and display the message 'Overtime', else (means if false) calculate grosspay without overtime and display nothing for the overtime message.

3) Draw a Flowchart or print a Pseudo Code



To Calculate **overtime grosspay** and display message '**overtime**'

For example, if you **work 60 hours** and your **Rate = $10**, to **calculate Overtime Gross Pay:**

1) Calculate the **original pay** for **40 hours** = **40 * 10 = 400**

2) Calculate the **Hours Worked Overtime** = **60 – 40** = **20 Overtime hours**

3) To **Calculate** the **Extra Pay over 40 hours** = **20 * 10 * 1.5** = **300**

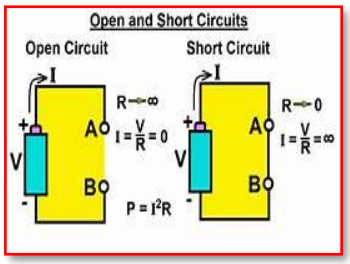
Note: **1.5 is Time and a Half** for **Overtime** work

4) Add Original Pay + Extra Pay over 40 hours = 400 + 300 = $700

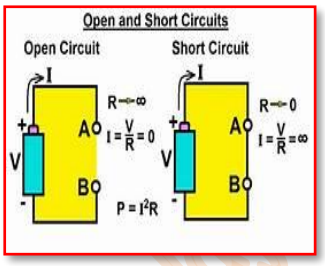
```
if hours > 40:
    gross_pay = 40 * rate + (hours - 40) * rate * 1.5
else:
    gross_pay = hours * rate
```

Use Python Interactive Mode to check simple and small program.

```
>>> hours = 60
>>> rate = 10
>>> if hours > 40:
...     gross_pay = 40 * rate + (hours - 40) * rate * 1.5
... else:
...     gross_pay = hours * rate
...
>>> print ('Gross Pay is = ', gross_pay)
Gross Pay is =  700.0
>>>
```



Using the Short-Circuit Conditional 'and' Operator In Python Language



Lesson 120 + How toUse Short-Circuit Conditional Operator 'and' to check if a Condition is True or False?

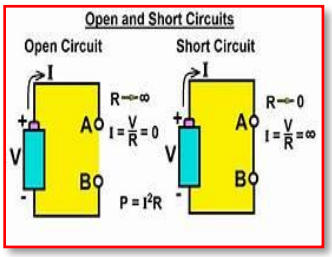
Using conditional Operator 'and' to Check if a Condition is True or False:

With Conditional Operator **and**, **Both Conditions must be True so** the **True block will be executed**; **otherwise**, the **False block will be executed**.

Use **and conditional Operator** when you want to check for a **Valid Range**. For Example, the **average** is in a **Range of 0 and 100.**

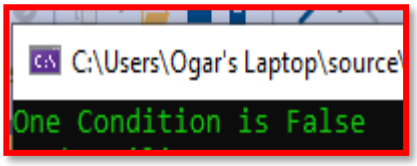
For Example:

```
average = 99
if average >= 0 and average <=100: #This is True for 99 is between 0 & 100
    print ('Average is Valid Number')
else:
    print ('Average is Invalid Number')
```



With Short-Circuit and Operator, **if 1 Condition is False**, then the **result is False** and the **Python compiler will NOT Check the rest of conditions**. For example,

```
a = 10
b = 20
c = 30
if  a == 20 and b == 20 and c == 30:     # a is Not == to 20. False
    print ('All Conditions are True')
```



```
else:
    print('One Condition is False')
```

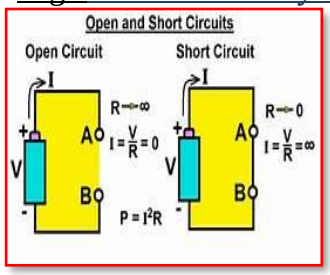
The Output is: **One Condition is False** as shown here:

The Short-Circuit 'and' Operator increases the Execution of the program because once compiler finds that 1 Condition is False, then it will ignore to Check the rest of conditions.

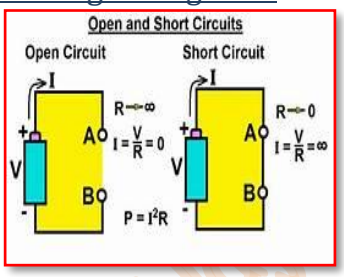
Truth table of 'and' Conditional operator:

a and b result
True and True result is True
True and False result is False
False and True result is False (Short-Circuit And)
False and False result is False (Short-Circuit And)

Notice: with 'and' logical Operator, the result is true only when Both conditions are True.



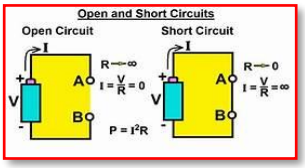
Using the Short-Circuit Conditional 'or' Operator In Python Language



Lesson 121 + How to Use Conditional Operator 'or' to Check if a Condition is True or False?

Using conditional 'or' Operator to Check if a Condition is True or False:

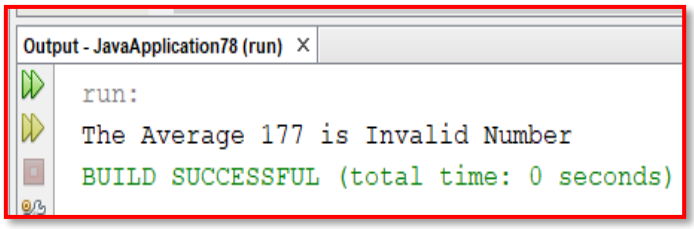
With Conditional Operator **or**, **Only 1 Condition may be True so** the **True block will be executed**; **otherwise**, the **False block will be executed**.



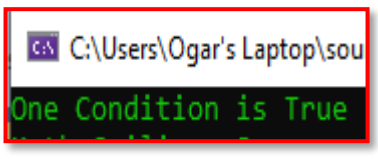
For Example:

```
average = 177
if average < 0 or average >100:      # This is True for 177 is > 100
    print ('The Average %d is Invalid Number' % (average) )
else:
    print ('Average %d is Valid Number' % (average) )
```

With Short-Circuit or Operator, **if 1 Condition is True**, then the **result is True** and the **Python compiler will NOT Check the rest of conditions**. For example,




```
a = 10
b = 20
c = 30
if a == 10 or b == 20 or  c == 30:   # a is == to 10. True (Short-Circuit)
    print ('One Conditions is True')
else:
    print ('One Condition is False')
```



The Output is: **One Condition is True** as shown here:

The Short-Circuit 'or ' Operator increases the Execution of the program because once compiler finds that 1 Condition is True, then it will ignore to Check the rest of conditions.

The Non Short-Circuit 'or' .

Truth table of or Conditional operator:

a or b result
True or True result is **True (Short-Circuit or)**
True or False result is **True (Short-Circuit or)**
False or True result is **true**
False or False result is **False**

Notice: with 'or' , the result is False only when Both conditions are False, otherwise **if only 1 condition is True**, then the result will be **True**.

Using the Conditional Operator: and Operator

Lesson 122+ How to use conditional Operator 'and' to check if a Condition is True or False?

Using conditional Operator 'and' to Check if a Condition is True or False:

With Conditional Operator **'and', Both Conditions must be True so** the **True block will be executed**; **otherwise**, the **False block will be executed**.

For Example:

```
average = 99
if average >= 0 and average <=100:    # Both conditions are met and True
        print ('Average {0} is Valid Number \n' , average  )
else:
        print ('Average {0} is Invalid Number \n' ,  average )
```

```
Output - CalculateSum3 (run)
run:
Average 99 is Valid Number
BUILD SUCCESSFUL (total time: 2 seconds)
```

Truth table of and Conditional operator:

a and b result

True and True result is **True**

True and False result is **False**

False and True result is **False** **(Short-Circuit And)**

False and False result is **False** **(Short-Circuit And)**

Notice: with 'and' operator, the result is True only when Both or All the conditions are True.

Using the Conditional Operator: or Operator

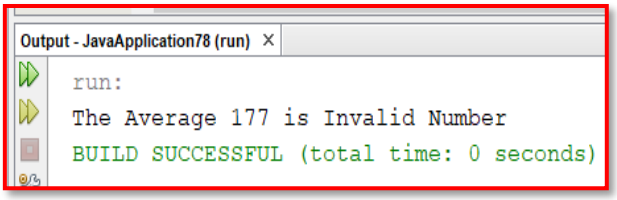
Lesson 123 + How to use conditional Operator 'or' to check if a Condition is True or False?

Using conditional or Operator to Check if a Condition is True or False:

With Conditional Operator **or, Only 1 Condition may be True so** the **True block will be executed**; **otherwise**, the **False block will be executed**.

For Example:

```
average = 177
if average < 0 or average >100:     # 1 condition is true (average is > 100)
    print ('The Average {0} is Invalid Number \n', average)
else :
    print ('The Average is {0} Valid Number \n', average)
```



Truth table of or Conditional operator:

a or b	result
True or True	result is True (Short-Circuit Or)
True or False	result is True (Short-Circuit Or)
False or True	result is True
False or False	result is False

Notice: with or, the result is False only when Both conditions are False.

‘Passing’ if average >=70 else ‘Failing’

Using the Ternary if …else Conditional Operator (if .. else)

‘Passing’ if average >=70 else ‘Failing’

Lesson 124 + How to conditional Ternary Operator (if ..else) to check if a Condition is True or False?

You can use Python if …else statements for Ternary Conditional Operator to check if a condition is True or False.

The format is as follows:

```
result = trueResult testExpression else falseResult
```

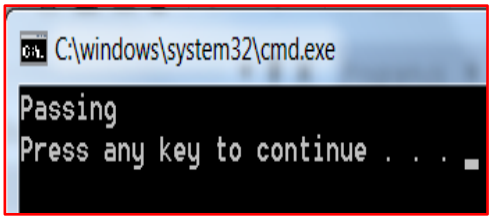
You test a condition, using if .. else statements for short one line condition testing.

For example:

average = 95

```
status = ‘Passing’ if average >= 70 else ‘Failing’
print (status)
```

The output written on output is: **Passing**

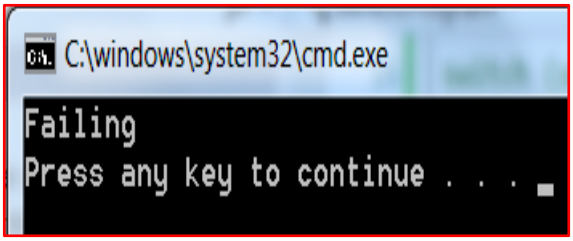


Another example:

average = 40

```
status = ‘Passing’ if average >= 70 else ‘Failing’
print (status)
```

The output written on output is: **Failing**



Not

Using the 'not' operator To Negate the Result (Opposite)

Not

Lesson 125 + How to use 'not' Operator to Negate the Result of comparison?

You can use the **not** **operator to change or Negate the result.**

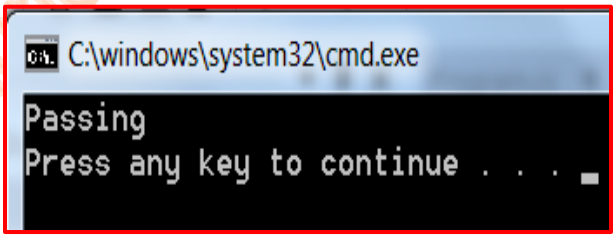
For example, **not (True)** will be **False**. **(Not True will be False)**

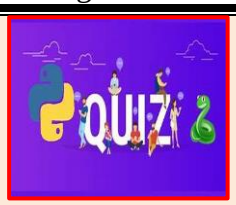
And **not (False)** will be **True**. **(Not False will be True)**

For example:

```
average = 95

if not (average >= 70):
    print ('Failing')
else:
    print ('Passing')
```

The output will be: **Passing**





Chapter 3 + Quiz 1 + Test Your Python Language Knowledge:

---

1. What is the output after executing the following Python code?

```
i = 0
while i < 1:                        # while 0 is < 1 is True
    print ('Hello', end=", ")       # print 'Hello, '
    i += 1                          # increment i by 1, so i = 1 and while False
else:
    print ("World")                 # this print 'World'
```

A. Hello B. "Hello", World C. Hello World D. Hello, World E. none

---

2. What is the output after executing the following Python code?

```
no_milk_left = "None"

if no_milk_left:            # This if statement is True
    print ("Buy Milk")
else:
    print ("You have Milk")
```

A. You have Milk B. Buy Milk C. error D. None

---

3. What is the output after executing the following Python code?

```
age = 21
print (age > 20 and age < 22)  # both conditions are True, so it prints True
```

A. 21 B. 22 C. True D. False E. None of these options

---

4. What is the output after executing the following Python code?

```
n = int (input ('Enter an integer: ') )      # suppose user enters 3
str = input ('Enter a string: ')             # suppose user enters 'Pie'
print ( n * str )
```

A. "PiePiePie" B. Pie Pie Pie C. PiePiePie D. 3 E. None of these options

---

---

5. What is the output after executing the following Python code?

```
name = input ('Enter your name: ')          # suppose user enters 'Mary'
age = int ( input ('Enter your age: ') )    # suppose user enters '21'
print (name, type (age) )
```

A. Mary 21 int B. Mary <class 'int'> C. Mary 21 D. Mary <class 'str'> E. error

---

6. What is the output after executing the following Python code?

```
name = ""           # name is assigned Empty String
while name:         # name is False, so else is executed
    print("Good Morning")
else:
    print("Good Night")
```

A. error B. Good Morning C. Good Night D. Good name="" E. nothing

---

7. What is the output after executing the following Python code?

```
average = 77
if average >= 90 and average <= 100:
    print("Grade 'A'")
elif average >= 80 and average <= 89:
    print("Grade 'B'")
elif average >=70 and average <= 79:  # with and both conditions must be True
    print("Grade 'C'")
elif average >=50 and average <= 69:
    print("Grade 'D'")
else:
    print("Grade 'F'")
```

A. Grade 'A' B. Grade 'B' C. Grade 'C' D. Grade 'D' E. Grade 'F'

---

8. What is the output after executing the following Python code?

```
k = (4 / 2) * 3 + 2   # do (4/2) first. 4/2=2.0 then 2.0*3=6.0, 6.0+2=8.0
print (k)
```

A. 8.0 B. 8 C. 1.2 D. 10 E. none of these options

Answers Are Found at The End of This Chapter 03.

Calculate the Average of 2 Tests and Assign Final Grade: 'A', 'B', 'C', 'D' or 'F'

+++(Do Lab Exercise 2) 100 Points +++

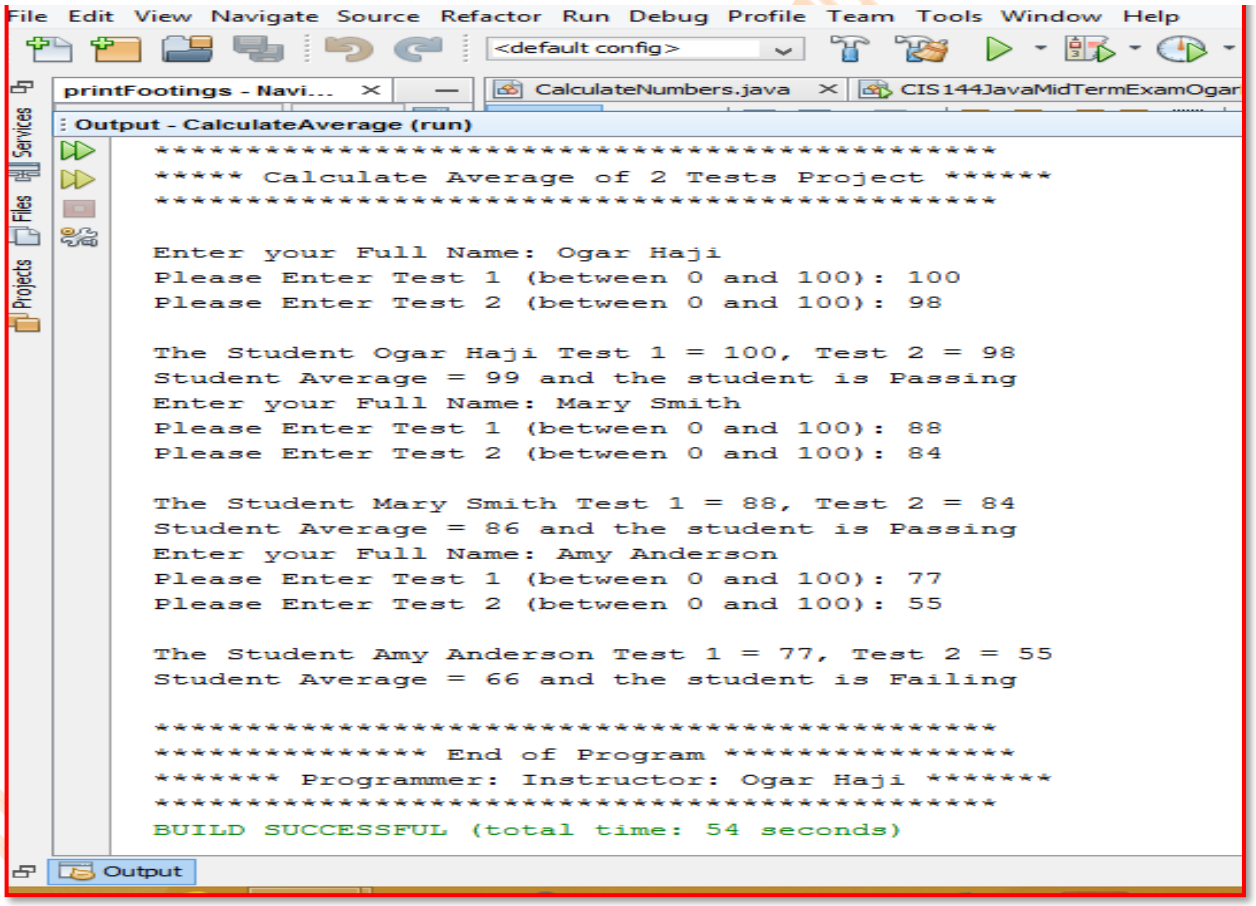
Do Lab Exercise 2

Lesson 126 Ex + How to Calculate the Average of 2 Tests and using If statement to check if the student is 'Passing' or 'Failing' and using Methods?

As you can See in doing this project, Calculate The Average o2 2 Tests" I start with the easiest Parts then I build an do the Advanced Project in Steps.

Do the 12 Must Steps to Design, Code and Solve a project using Python Language.

The Input and Output of the project will look like the following:



This is an Explanation of What we will Do in this project:

 # 1) Add Comments about the project on the top of the project.

1) Add Comments about the project on the top of the project.

```
'''
 * This program calculates the average of 2 Tests and
 *  prints out if the student passed or failed. Also Find the class average
 *  Programmer: Ogar Haji (Python Instructor)
 *  Date: Friday, November 03, 2017
 *  Project Name: CalculateFinalGrade
'''

# 2) Initialize the variables to be used in this project
total_average = 0
class_average = 0
students_counter = 0
a_counter = 0        # to count the 'a' student in the class
average = 0
status = ''          # Assign the empty string '' to status
PASSING_SCORE = 70

# 3) Call the methods to be used in this project from the main() method
print ('*****************************************')
print ('*** Calculate Average of 2 Tests Project *****')
print ('*****************************************')
print ()    # print a Blank Line

# 4) Prompt the user to enter Full Name
full_name = input ('Enter your Full Name: ')

# 5) Prompt the user to enter Test 1 between 0 and 100
test1 = input ('Please Enter Test 1 (between 0 and 100): ')

# 6) Convert the string test1 to int using int() function
test1 = int (test1)

# 7) Prompt the user to enter Test 2 between 0 and 100
test2 = input ('Please Enter Test 2 (between 0 and 100): ')

# 8) Convert the string test2 to int using int() function
test2 = int (test2)
```

```
Enter your Full Name: Ogar Haji
Please Enter Test 1 (between 0 and 100): 100
Please Enter Test 2 (between 0 and 100): 98
```


```
Enter your Full Name: Ogar Haji
Please Enter Test 1 (between 0 and 100): 100
Please Enter Test 2 (between 0 and 100): 98
```

# 9) Find average of the 2 tests using integer // divisor
average = (test1 + test2) // 2

10) Add the average to the total average
total_average += average

11) Find out if the student is Passing or Failing
```
if average >= PASSING_SCORE:
        status = 'Passing'
else:
        status = 'Failing'
```

```
Enter your Full Name: Ogar Haji
Please Enter Test 1 (between 0 and 100): 100
Please Enter Test 2 (between 0 and 100): 98
The Final Grade is:  A

The Student Ogar Haji Test 1 = 100, Test 2 = 98
```

12) Print out the data about the student
```
print ()                    # Print a Blank Line
print ('The Student %s Test 1 = %d, Test 2 = %d \n' % (full_name, test1, test2) )
print ('Student Average = %d and the student is %s \n' % (average, status) )
```

13) Print End of Program message
```
print ()
print ('*********************************************')
print ('********** End of Program ****************')
print ('***** Programmer: Instructor: Ogar Haji *******')
print ('*********************************************')
```

+ Do Lab Exercise 2 + Do Lab Exercise 2 +

3) Save the project as **CalculateFinalGrade.py** **Do Lab Exercise 2**

```
# 1) Add Comments about the project on the top of the project.
'''
 * This program calculates the average of 2 Tests and
 *  prints out if the student passed or failed. Also Find the class average
 *  Programmer: Ogar Haji (Python Instructor)
```

```
 *  Date: Friday, November 03, 2017
 *  Project Name: CalculateFinalGrade
'''
```

Do Lab Exercise 2

```
# 2) Initialize the Local variables to be used in this project
total_average = 0
class_average = 0
students_counter = 0
a_counter = 0        # to count the 'a' student in the class
average = 0
status = ''          # Assign the empty string '' to status

# 3) Print the Headings of the project
print ('*****************************************')
print ('*** Calculate Average of 2 Tests Project *****')
print ('*****************************************')
print ()    # print a Blank Line

# 4) Prompt the user to enter Full Name
full_name = input ('Enter your Full Name: ')
```

```
Enter your Full Name: Ogar Haji
Please Enter Test 1 (between 0 and 100): 100
Please Enter Test 2 (between 0 and 100): 98
```

```
# 5) Prompt the user to enter Test 1 between 0 and 100
test1 = input ('Please Enter Test 1 (between 0 and 100): ')

# 6) Convert the string test1 to int using int() function
test1 = int (test1)

# 7) Prompt the user to enter Test 2 between 0 and 100
test2 = input ('Please Enter Test 2 (between 0 and 100): ')

# 8) Convert the string test2 to int using int() function
test2 = int (test2)

# 9) Find the Average of the 2 Tests
average = (test1 + test2 ) // 2

# 10) Add the average to the total average
total_average += average
```

```
Enter your Full Name: Ogar Haji
Please Enter Test 1 (between 0 and 100): 100
Please Enter Test 2 (between 0 and 100): 98
```

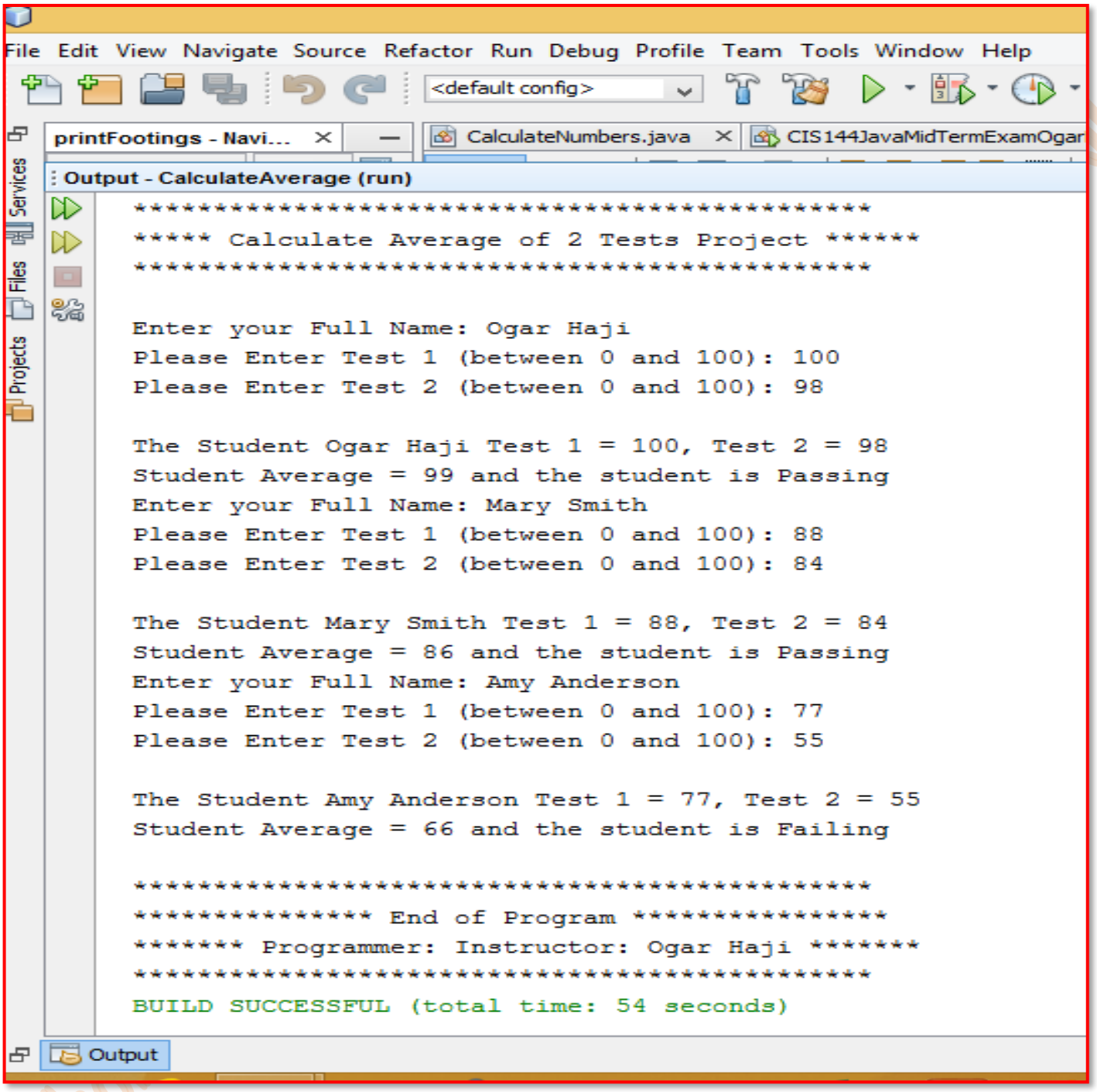
```
# 11) Find out if the student is Passing or Failing
if average >= 70:
        status = 'Passing'
else:
        status = 'Failing'
# 12) Print out the data about the student
print ()                    # Print a Blank Line
print ('The Student %s Test 1 = %d, Test 2 = %d \n' % (full_name,
test1, test2) )
print ('Student Average = %d and the student is %s \n' % (average,
status) )
# 13) Print End of Program message
print ()
print ('*************************************************')
print ('************ End of Program ******************')
print ('***** Programmer: Instructor: Ogar Haji *******')
print ('*************************************************')
```

```
Enter your Full Name: Ogar Haji
Please Enter Test 1 (between 0 and 100): 100
Please Enter Test 2 (between 0 and 100): 98
The Final Grade is:  A

The Student Ogar Haji Test 1 = 100, Test 2 = 98
```

The Input/Output of the Project Calculate Final Grade

```
****************************************
*** Calculate Average of 2 Tests Project *****
****************************************

Enter your Full Name: Ogar Haji
Please Enter Test 1 (between 0 and 100): 100
Please Enter Test 2 (between 0 and 100): 98
The Final Grade is:  A

The Student Ogar Haji Test 1 = 100, Test 2 = 98

Student Average = 99 and the student is Passing with Grade of "A"

*************************************************
*************** End of Program ********************
******** Programmer: Instructor: Ogar Haji ***********
*************************************************
```

The Input and Output of the project will look like the following:



Modify This Project to Do the Following Modifications: 30%

Lesson 127 Ex + Modify the Calculate Final Grade Project to Check for Invalid Tests Data.

1) Modify the project to Check if test1 is Invalid Test. Invalid Test is: Invalid Test is test1 Less than 0 or test1 Greater than 100.

while (test1 < 0 or test1 > 100): # Check for Invalid Test1

Valid Test is test1 Greater Equal than 0 or test1 Less than or Equal to 100.

while (test1 >= 0 and test1 <= 100): (Valid Test1) Check for the Range.

5) Prompt the user to enter Test 1 (between 0 and 100)
test1 = input ('Please Enter Test 1 (between 0 and 100): ')

6) Convert the string test1 to int using int() function
test1 = int (test1)

Modify Project

7) Check for **Invalid Data (test1 < 0 or test1 > 100)**

while (test1 < 0 or test1 > 100):

    print (' Invalid test ++ Test 1 should be between 0 and 100 **')**

    # 8) Prompt the user to enter Test 1 between 0 and 100
    test1 = input ('Please Enter Test 1 (between 0 and 100): ')

    # 9) Convert the string test1 to int using int() function
    test1 = int (test1)

10) Prompt the user to enter Test 2 between 0 and 100
test2 = input ('Please Enter Test 2 (between 0 and 100): ')

11) Convert the string test2 to int using int() function
test2 = int (test2)

Modify Project

12) Check for Invalid Data (test2 < 0 or test2 > 100)

while (test2 < 0 or test2 > 100):

    print (' Invalid test ++ Test 2 should be between 0 and 100 **')**

    # 13) Prompt the user to enter Test 2 between 0 and 100
    test2 = input ('Please Enter Test 2 (between 0 and 100): ')

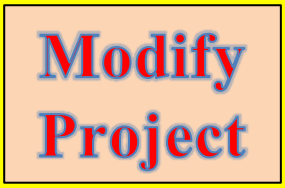
    # 14) Convert the string test2 to int using int() function
    test2 = int (test2)

# 15) Find the Average of the 2 Tests
average = (test1 + test2) // 2

Modify This Project to Do the Following Modifications: 30%

Modified Version of the project (Use While to check for Invalid Data)

```
'''
# 1) Add Comments about the project on the top of the project.
 * This program calculates the average of 2 Tests and prints out
 * if the student passed or failed.  Also Find the class average
 * Programmer: Ogar Haji (Python Instructor)
 * Date: Friday, November 03, 2017
 * Project Name: CalculateFinalGrade
'''

# 2) Initialize the Local variables to be used in this project
total_average = 0
class_average = 0
students_counter = 0
a_counter = 0          # to count the ‘a’ student in the class
average = 0
status = ''         # Assign the empty string ‘’ to status

# 3) Print the Headings of the project
print ('*****************************************')
print ('*** Calculate Average of 2 Tests Project *****')
print ('*****************************************')
print ()    # print a Blank Line

# 4) Prompt the user to enter Full Name
full_name = input ('Enter your Full Name: ')

# 5) Prompt the user to enter Test 1 between 0 and 100
test1 = input ('Please Enter Test 1 (between 0 and 100): ')

# 6) Convert the string test1 to int using int() function
test1 = int (test1)

# 7) Check for Invalid Data (test1 < 0 or test1 > 100)
```



```
while (test1 < 0 or test1 > 100):
    print ('** Invalid test ++ Test 1 should be between 0 and 100 **')

    # 8) Prompt the user to enter Test 1 between 0 and 100
    test1 = input ('Please Enter Test 1 (between 0 and 100): ')

    # 9) Convert the string test1 to int using int() function
    test1 = int (test1)

# 10) Prompt the user to enter Test 2 between 0 and 100
test2 = input ('Please Enter Test 2 (between 0 and 100): ')

# 11) Convert the string test2 to int using int() function
test2 = int (test2)

# 12) Check for Invalid Data (test2 < 0 or test2 > 100)
while (test2 < 0 or test2 > 100):
    print ('** Invalid test ++ Test 2 should be between 0 and 100 **')

    # 13) Prompt the user to enter Test 2 between 0 and 100
    test2 = input ('Please Enter Test 2 (between 0 and 100): ')

    # 14) Convert the string test2 to int using int() function
    test2 = int (test2)

# 15) Find the Average of the 2 Tests
average = (test1 + test2 ) // 2

# 16) Add the average to the total average
total_average += average

# 17) Find out if the student is Passing or Failing
if average >= 70:
    status = 'Passing'
else:
    status = 'Failing'

# 18) Print out the data about the student
print ()            # Print a Blank Line
```

Modify Project

```
print ('The Student %s Test 1 = %d, Test 2 = %d \n' % (full_name, test1, test2) )

print ('Student Average = %d and the student is %s \n' % (average, status) )

# 19) Print End of Program message
print ()
print ('*********************************************')
print ('************ End of Program ****************')
print ('***** Programmer: Instructor: Ogar Haji *******')
print ('*********************************************')
```

Checking for Invalid Data: **while (test1 < 0 or test1 >100):**

```
Enter your Full Name: Ogar Haji
Please Enter Test 1 (between 0 and 100): -22
** Invalid test ++ Test 1 should be between 0 and 100 **
Please Enter Test 1 (between 0 and 100): 133
** Invalid test ++ Test 1 should be between 0 and 100 **
Please Enter Test 1 (between 0 and 100): 90
Please Enter Test 2 (between 0 and 100): -11
** Invalid test ++ Test 2 should be between 0 and 100 **
Please Enter Test 2 (between 0 and 100): 144
** Invalid test ++ Test 2 should be between 0 and 100 **
Please Enter Test 2 (between 0 and 100): 95

The Student Ogar Haji Test 1 = 90, Test 2 = 95

Student Average = 92 and the student is Passing
```

Modify This Project to Do the Following Modifications: 30%

Lesson 128 Ex + Modify the Calculate Final Grade Project to Add another Test3 And Assign the Final Grades 'A', 'B', 'C', 'D', 'F'.

1) Modify the project to Add another test which will be test3 and then Calculate the Average of 3 Tests.

2) Modify project to **Assign the Final Grade** according to the following **Average** Criteria:

90 - 100 Grade A
80 - 89 Grade B
70 - 79 Grade C
50 - 69 Grade D
0 - 49 Grade F

Modify Lab Exercise 1

```
if average >= 90:
        final_grade = 'A'
        a_count +=1

elif average >= 80:
        final_grade = 'B'
        b_count +=1

elif average >= 70:
        final_grade = 'C'

elif average >= 50:
        final_grade = 'D'

elif average >= 0:
       final_grade = 'F'

else:
      final_grade = 'U'
```

Modify This Project to Do the Following Modifications: 30%

Lesson 129 Ex + Modify the Calculate Final Grade Project to Add Named Constants to the Project.

3) Modify the **program** to **use Named Constants in case of Numbers like 90, 80, 70 as the following:**

Modify Lab Exercise 2

A_GRADE = 90

B_GRADE = 80 and so on for the rest

if average >= A_GRADE:

```
****************************************
*** Calculate Average of 2 Tests Project *****
****************************************

Enter your Full Name: Ogar Haji
Please Enter Test 1 (between 0 and 100): 100
Please Enter Test 2 (between 0 and 100): 98
The Final Grade is:  A

The Student Ogar Haji Test 1 = 100, Test 2 = 98

Student Average = 99 and the student is Passing with Grade of "A"


****************************************
********** End of Program ***************
***** Programmer: Instructor: Ogar Haji *******
****************************************
```

```
****************************************
*** Calculate Average of 2 Tests Project *****
****************************************

Enter your Full Name: Mary Smith
Please Enter Test 1 (between 0 and 100): 88
Please Enter Test 2 (between 0 and 100): 84
The Final Grade is:  B

The Student Mary Smith Test 1 = 88, Test 2 = 84

Student Average = 86 and the student is Passing with Grade of "B"


****************************************
********** End of Program ***************
***** Programmer: Instructor: Ogar Haji *******
****************************************
```

Modified and Not Complete project to Calculate Final Grade.

```
# 1) Add Comments about the project on the top of the project.
'''
 * This program calculates the average of 2 Tests and
 *  prints out if the student passed or failed. Also do the modifications
 *  Programmer: Ogar Haji (Python Instructor)
 *  Date: Friday, November 03, 2017
 *  Project Name: CalculateFinalGrade          Modify Lab Exercise 2
'''
# 2) Define and Initialize the variables to be used in this project
total_average = 0
class_average = 0
students_counter = 1
aCounter = 0          # to count the 'a' student in the class
average = 0
status = ''

# Define and initialize Named Constants
A_GRADE = 90
B_GRADE = 80                                  Modify Lab Exercise 2

# 3) Define and initialize the counter to 0
a_count = 0
b_count = 0

# 4) Print the Heading of the project
print ('*****************************************')
print ('*** Calculate Average of 2 Tests Project *****')
print ('*****************************************')
print ()    # print a Blank Line
```

```
# 5) Prompt the user to enter Full Name
full_name = input ('Enter your Full Name: ')

# 6) Prompt the user to enter Test 1 between 0 and 100
test1 = input ('Please Enter Test 1 (between 0 and 100): ')

# 7) Convert the string test1 to int using int() function
test1 = int (test1)
```

```
Enter your Full Name: Ogar Haji
Please Enter Test 1 (between 0 and 100): 100
Please Enter Test 2 (between 0 and 100): 98
```

```
# 8) Prompt the user to enter Test 2 between 0 and 100
test2 = input ('Please Enter Test 2 (between 0 and 100): ')

# 9) Convert the string test2 to int using int() function
test2 = int (test2)

# 10) Calculate the Average of the 2 Tests
average = (test1 + test2 ) // 2

# 11) Add the average to total average
total_average += average

# 12) Find out if the student is Passing or Failing
if average >= 70:
   status = 'Passing'

else:
   status = 'Failing'

# 13) Add the modified code here to find the final grade ‘A’, ‘B’ and so
# Use if....elif (else if) to Find the final grade
if average >= A_GRADE:
    final_grade = ‘A’
    a_count += 1     # increment the a_count by 1
```

```
The Final Grade is:  A

The Student Ogar Haji Test 1 = 100, Test 2 = 98
```

```
elif average >= B_GRADE:
    final_grade= 'B'
    b_count += 1
elif average >= 70:
    final_grade = 'C'     # You must increment c_count by 1
elif average >= 50:
    final_grade = 'D'
elif average >= 0:
    final_grade = 'F'
else:
    final_grade = 'U'
```

```
Student Average = 99 and the student is Passing with Grade of "A"

*********************************************************
*************** End of Program **********************
******** Programmer: Instructor: Ogar Haji ***********
*********************************************************
```

```
# 14) print out the Final Grade
print ('The Final Grade is: ', final_grade)
# 15) Print out the data about the student
print ()       # Print a Blank Line
print ('The Student %s Test 1 = %d, Test 2 = %d \n' % (full_name, test1, test2) )
print ('Student Average = %d and the student is %s with Grade of '%s' \n' % (average, status, final_grade) )
```

```
Student Average = 99 and the student is Passing with Grade of "A"

*********************************************************
*************** End of Program **********************
******** Programmer: Instructor: Ogar Haji ***********
*********************************************************
```

```
# 16) Print End of Program message
print ('*' * 55)    # repeat the '*' 55 times and print it
print ('*************** End of Program **********************')
print ('******** Programmer: Instructor: Ogar Haji **********')
print ('*' * 55)
```

The Input/Output of the Project:

```
****************************************
*** Calculate Average of 2 Tests Project *****
****************************************

Enter your Full Name: Ogar Haji
Please Enter Test 1 (between 0 and 100): 100
Please Enter Test 2 (between 0 and 100): 98
The Final Grade is:  A

The Student Ogar Haji Test 1 = 100, Test 2 = 98

Student Average = 99 and the student is Passing with Grade of "A"

****************************************************
*************** End of Program ***********************
******* Programmer: Instructor: Ogar Haji ***********
****************************************************
```

Note: Always **Upload** to **Brightspace the Modified Python Project:**
1) The **Microsoft Word Document** of the **Python Project Code along with the Python Output Screen shots.**

Chapter 3 + Quiz 2 + Test Your Python Language Knowledge:

---

1. What is the output after executing the following Python code?

```
number = 22
if number or True:      # number has a value 22 which means True
    number = 33         # assign 33 to number and replace old number
print (number)
```

A. 22 B. 55 C. 33 D. none E. error

---

2. What is the output after executing the following Python code?

```
price = input ('Enter the price of the Book: ')  # Assume user enters 21.88
print ( type (price) )      # user enters '21.88' and it is string
```

A. <class 'int'> B. <class 'double'> C. <class 'str'> D. <class 'double'> E. error

---

3. What is the output after executing the following Python code?

```
number = input ('Enter number of students: ')  # Assume user enters  7
print ( type (number) )      # user enters '7' and it is string
```

A. <class 'int'> B. <class 'double'> C. <class 'str'> D. <class 'double'> E. error

---

4. To Convert a String Number entered by user to float, use function: .

A. int () B. str () C. float () D. bool() E. None of these options

---

5. What is the output after executing the following Python code?

```
print (max ( 22, 11, 44, 77 ) )
```

A. 22 B. 11 C. 44 D. 77 E. None of these options

---

6. What is the output after executing the following Python code?
print (min (22, 11, 44, 77))

A. 22 B. 11 C. 44 D. 77 E. None of these options

---

7. Match the following Symbols used in Python language:

A. **#** B. **%** C. **/** D. **//** E. ****

A. Line comment **B. Modulus operator** **C. Integer division**

D. Float Division **E. Line continuation**

---

8. What is the output after executing the following Python code?

```
average = 97
if average >= 70:                # if 97 is >= 70 which is True
    print ('Passing')            # then prints 'Passing'
else:
    print ('Failing')
```

A. nothing B. Passing C. Failing D. 97E. None of these options

---

9. What is the output after executing the following Python code?

```
i = 1
while i < 10:          # while 1 < 10: which is True
    i += 3             # increment i by 3, i = 1 + 3 = 4
    print (i)          # prints value of i which is 4, then repeats while for 7, 10
```

A. 4 7 10 B. 1 4 7 C. 2 6 10 D. Syntax Error

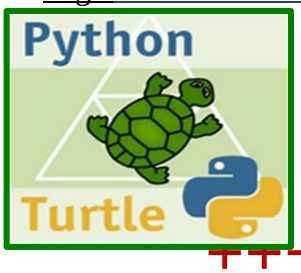
---

10. What is the output after executing the following Python code?

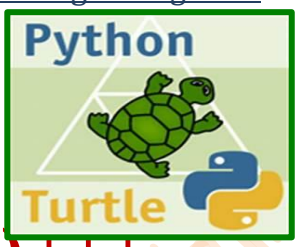
```
print (7 * 7 / 7 * 7)     # * and / have same precedence. From left to right
                          # 7 * 7 = 49,  49 / 7 = 7.0, 7.0 * 7 = 49.0
```

A. 0 B. 1 C. 1.0 D. 49 E. 49.0 F. error

Answers Are Found at The End of This Chapter 03.



Python Language
Introduction to Python Turtle
To Draw Shapes



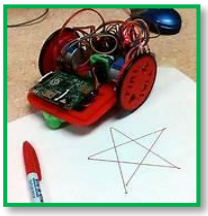
+++ (Read and Study This Lesson) +++

Lesson 130 + How to Draw Shapes using Python Turtle program?

Turtle is a **Python Module** or **Library** which is used to **Create** and **Design Shapes, graphics, pictures, games** and **Animations**.

Python Turtle was developed by **Wally Feurzeig, Seymour Parpet** and **Cynthina Slolomon** in **1967**.

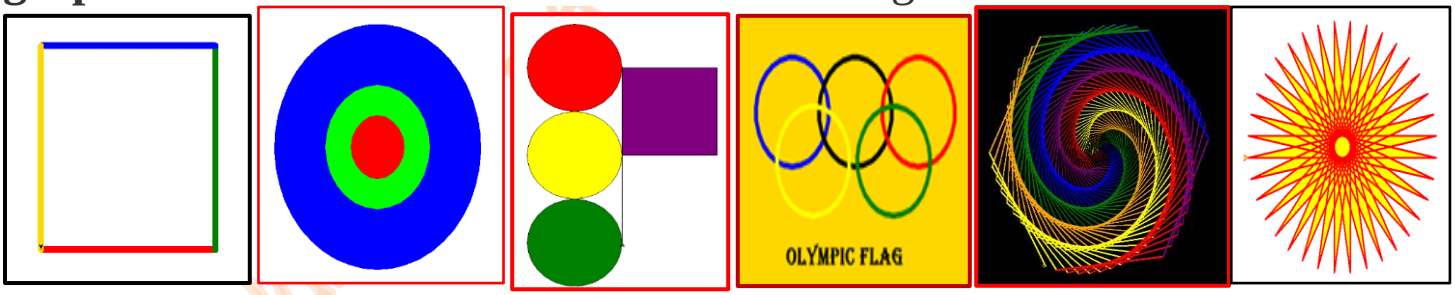
It was a part of the **original Logo programming language**.



In the **late 1960s**, **MIT professor Seymour Papert** used a **Robotic 'turtle'** to **teach programming to students**.

'turtle' is a **Python module** or **feature** like a **drawing board**, which lets us **command** a **turtle** to **draw all over the paper**.

By using the **commands** in **Python turtle**, you can **draw nice shapes**, **graphics** and **do animations** like the following:

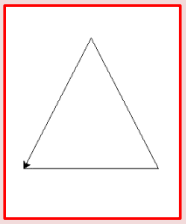


- **Python turtle** is a **Module** or **Library** which is used to **create shapes, patterns,** and **Games** on the Screen.
- **Python turtle** is initially **positioned** in the **Center** of the **Graphic Window** and the **turtle** looks like an **Arrow Head ▶**.
- **Python turtle** default heading is **0 degrees** and facing **East**.

```
import turtle
turtle.forward(100)
```

Some of Python turtle() functions:

- **Turtle** is a **python feature** like a **drawing board**, which **allows** you **to command** a **turtle** to **draw Lines** and **make Shapes** and **Games**.

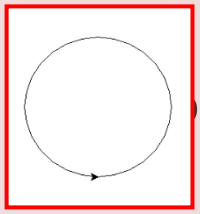


- **1)** To **use turtle**, we must **import turtle** module first.

```
import turtle
```

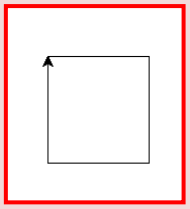
- **2)** Use **turtle.forward (100)** function to **move forward** and **draw** a **line** of **length 100 pixels**. **(96 Pixels are equal to 1 inch)**
- **3)** Use **turtle.left (90)** function to **turn the turtle left** by **90 degrees (left is Up the window screen)**.
- **4)** Use **turtle.right (90)** function to **turn the turtle Right** by **90 degrees (right is Down the window screen)**.
- **5)** Use **turtle.setheading (90)** function to **set the turtle heading** to **90 degrees**.

- **6)** Use **turtle.heading ()** function to **get and display the turtle current heading**.
- **7)** Use **turtle.penup ()** function to **raise** the **turtle pen up** so **No Drawing will take effect**.
- **8)** Use **turtle.pendown ()** function to **lower the turtle pen down and will start Drawing**.



- **9)** Use **turtle.circle (25)** function to **Draw a Circle** of **radius 25 pixels**.
- **10)** Use **turtle.pensize (5)** function to **set** the **turtle pensize** to **size 5 pixels**. **(a little bit thicker than 1 pixel the default pensize)**
- **11)** Use **turtle.dot ()** function to **draw a Dot (.)** on window screen.

12) Use **turtle.pencolor ('red')** function to change the **pen color** to **red color**. (**Black** color is the **default pencolor**)



13) Use **turtle.bgcolor ('yellow')** function to change the **background color** of the window screen to **yellow color**.

14) Use **turtle.setup (640,480)** function to **set the size** of the **window screen** to **Width 640 pixels** and **Height** to **480 pixels**.

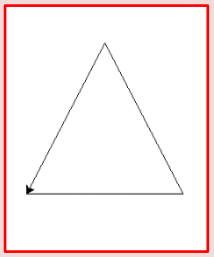
15) Use **turtle.goto (0, 100)** function to **move** the **turtle pen** to **location 0, 100** which **is x and y coordination's**.

16) Use **turtle.xcor ()** function to **return** the **x coordination** of the **current position** of **turtle pen**.

17) Use **turtle.ycor ()** function to **return** the **y coordination** of the **current position** of **turtle pen**.

18) Use **turtle.speed (0)** function to change turtle **Animation speed** of the **turtle pen drawing** to **0**. (**0** is the **Fastest speed with No Animation**). Turtle **Speed settings** can be set between **1 (Slowest)** to **10 (Fastest)**.

turtle.speed (0) is the Super Fastest and No Animation.
turtle.speed (10) is the Fast speed.
turtle.speed (6) is the Normal speed.
turtle.speed (3) is Slow speed.
turtle.speed (1) is the Slowest speed.



19) Use **turtle.hide ()** function to **hide** the **turtle arrow head**.

20) Use function **turtle.showturtle ()** to **show turtle arrow head**.

21) Use **turtle.write ('Ogar Haji')** function to **write the literal string 'Ogar Haji'** on the Window screen.

22) Use **turtle.begin_fill ()** function to **begin** to **fill** a **shape** with **color**

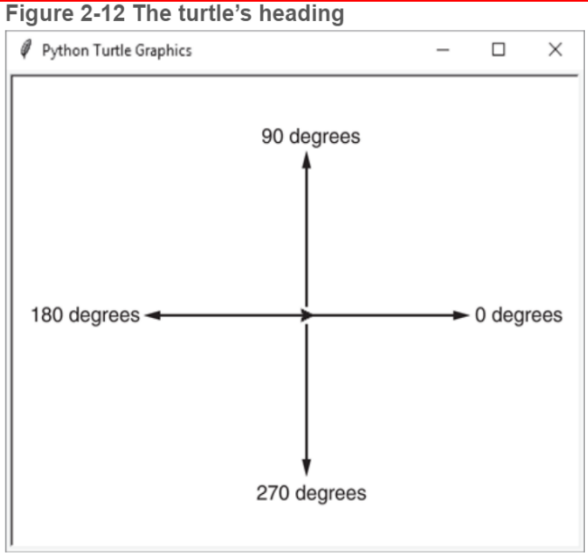
- **23)** Use **turtle.fillcolor ('green')** function to **fill the shape** with **color green**.
- **24)** Use **turtle.end_fill ()** function to **end** to **fill** a **shape** with **color**
- **25)** Use **turtle.shape ('turtle')** to **display** the **turtle shape instead** of **Arrow shape.**
- **26)** Use **turtle.home()** function to **move** the **turtle** to the **origin coordinates (0,0)**.
- **27)** Use **turtle.done ()** function to Inform turtle that **we are done** and **leave the window open on screen**.

turtle.done()

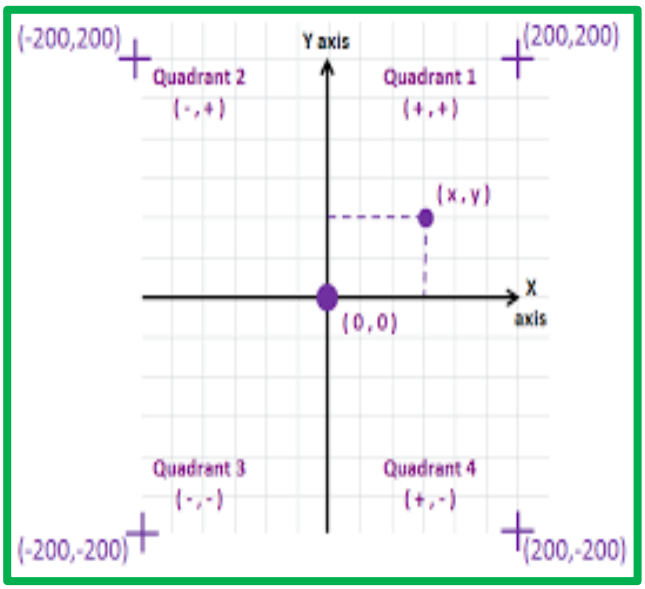
- The **python turtle Module or Library contains** all the **methods** and **functions** that we need to **create Shapes and Animation on screen**.
- This shape is the **Turtle Heading** ▶ which is **Facing toward East**. →

Turtle Left and Right Degrees

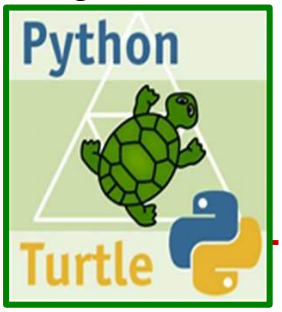
Figure 2-12 The turtle's heading



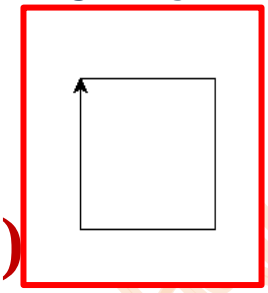
Turtle X and Y Coordinates



```
import turtle
turtle.forward(100)
```



Python Language
Introduction to Python Turtle
To Draw a Square Shape
+(Read and Study This Lesson)

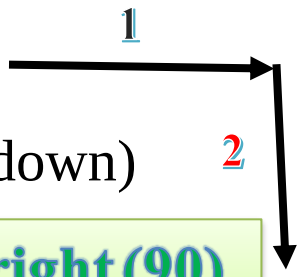


Lesson 101 + How to Draw a Square using Python Turtle program?

You can **Draw a Square** using Python **turtle.forward (100)** and **turtle.right (90)** or **turtle.left (90)** functions of the turtle library as shown below:

turtle.forward (100)

```
import turtle
turtle.forward (100)
turtle.right (90)        # will turn turtle right (down)
turtle.forward (100)
```



turtle.right (90)

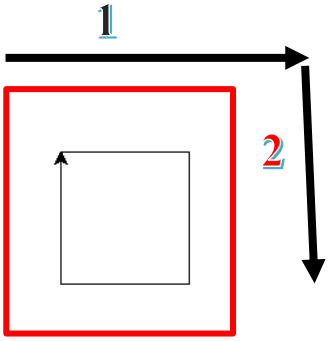
And repeat the above 4 times to draw All 4 Sides of a Square.

But you have to import the Python turtle library first.

```
import turtle
```

This is an Explanation of What we will Do in this project:

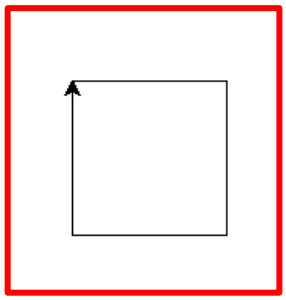
```
# To Draw a Square on screen using turtle with length and width of 100 pixels
# Date: 9/19/2021
# Programmer: Ogar Haji
# 1) import the turtle class Library
import turtle
# 2) Move the turtle forward by 100 pixels
turtle.forward (100)
```



3) Turn the turtle right by 90 degrees (**Down**)
turtle.right (90)

4) Move the turtle forward by 100 pixels
turtle.forward (100)

5) Turn the turtle right by 90 degrees
turtle.right (90)



6) Move the turtle forward by 100 pixels
turtle.forward (100)

7) Turn the turtle right by 90 degrees
turtle.right (90)

8) Move the turtle forward by 100 pixels
turtle.forward (100)

1) Start a New File so you will type your program: Click **File, New File**

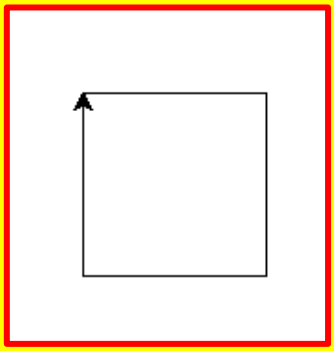
2) Type the following DrawSquare.py Python Code:

3) Save Python File as **'TurtleDrawSquareUsingNamedConstants.py'**

To Draw a Square on screen using turtle with length and width of 100 pixels
Date: 9/19/2021
Programmer: Ogar Haji

1) import the turtle class Library
import turtle

2) Move the turtle forward by 100 pixels
turtle.forward (100)



3) Turn the turtle right (down) by 90 degrees
turtle.right (90)

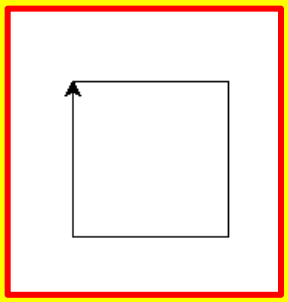
4) Move the turtle forward by 100 pixels
turtle.forward (100)

5) Turn the turtle right (down) by 90 degrees
turtle.right (90)

6) Move the turtle forward by 100 pixels
turtle.forward (100)

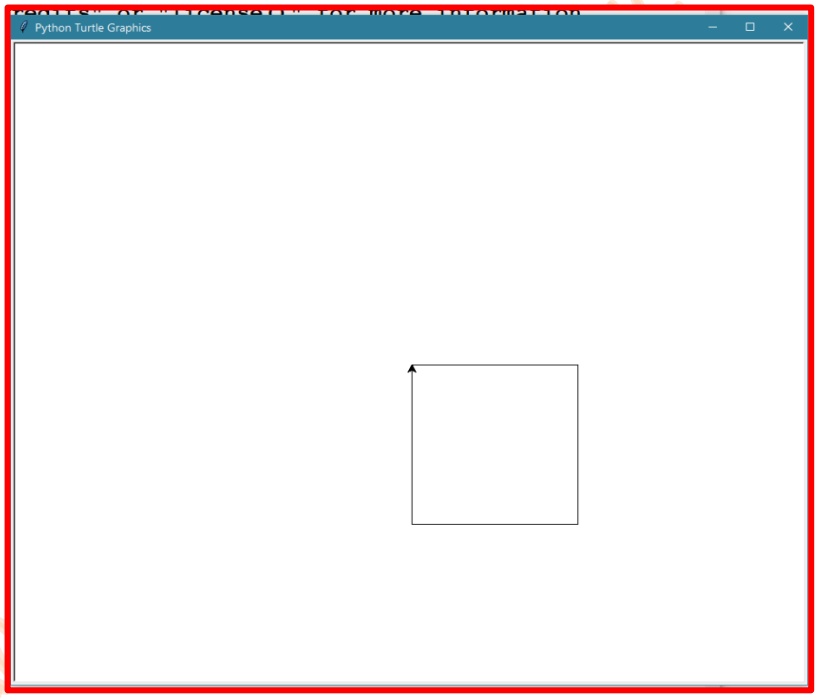
7) Turn the turtle right (down) by 90 degrees
turtle.right (90)

8) Move the turtle forward by 100 pixels
turtle.forward (100)



9) Inform turtle that we are done and leave the window open on screen
turtle.done()

The Output of the Python turtle project to Draw a Square



Modify This Project to Do the Following Modifications: 30%

A **Named Constant** is a **name that represents a Constant Value that does not change during the program's execution**.

The following is an example of how we will **declare named constants and initialize Named Constant** in our code:

PIXELS = 200

Note: It is **customary** to **Declare** a **Named Constant** in **ALL UPPER CASE** in **Python language**.

 # 1.1) Declare a Named Constant 'PIXELS' and initialize it to 100

```
PIXELS = 100
```

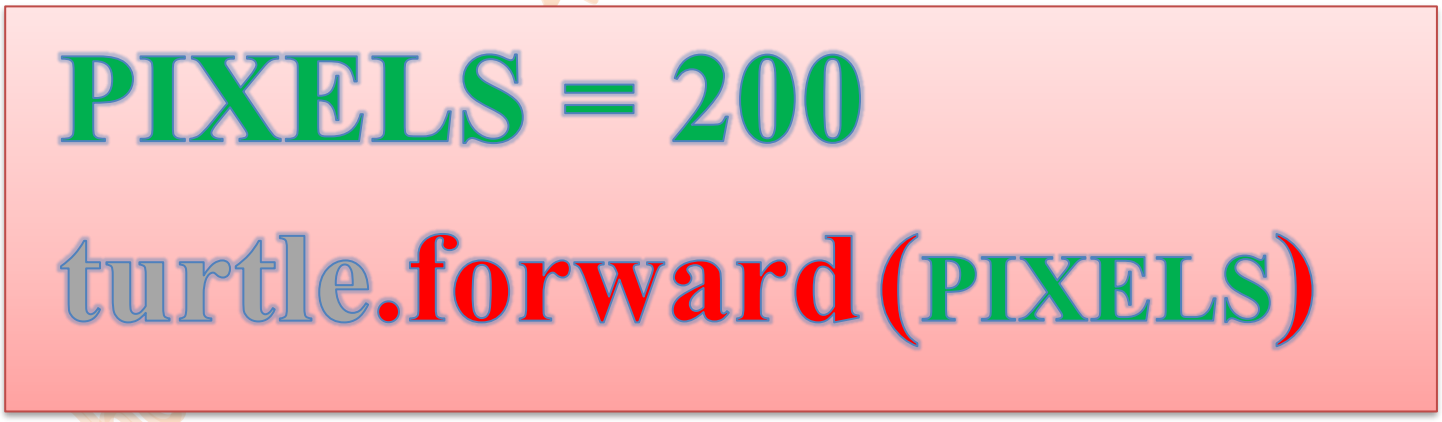
 # 2) Then add PIXELS to all turtle.forward (PIXELS) functions as shown below:

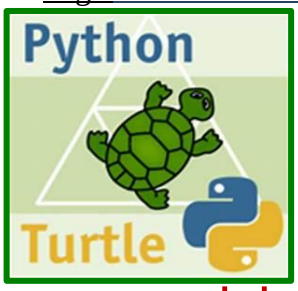
```
turtle.forward (PIXELS)
```

To do the following modifications, just assign these length 200 to the Named Constant 'PIXELS':

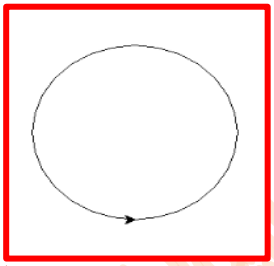
```
PIXELS = 200
```

1) **Use Turtle to Draw a Square of Length 200 pixels.**
2) **Use Turtle to Draw a Square of Length 300 pixels.**
3) **Use Turtle to Draw a Square of Length 350 pixels.**
4) **Upload the Last Modified project to Brightspace.**





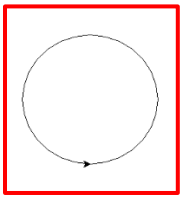
Python Language
Introduction to Python Turtle
To Draw a Circle Shape



+++ (Read and Study This Lesson) +++

Lesson 131 Ex + How to Draw a Circle Shape using Python Turtle program?

You can **Draw a Circle** using Python **turtle.circle (100)** function of the turtle library will draw a **Circle of radius 100 pixels**.



1) First, **import** the **Python turtle library**.

import turtle

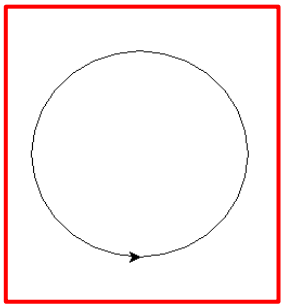
import turtle

2) Draw a Circle of Radius 100 pixels by typing the following statement:

turtle.circle (100)

turtle.circle (100)

This is an Explanation of What we will Do in this project:



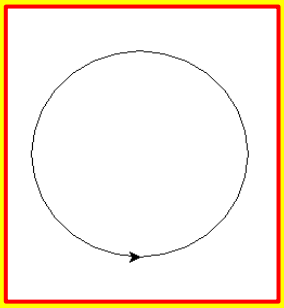
```
# To draw a Circle
# Date: 10/21/2019
# Programmer: Ogar Haji
# 1) import the turtle library
import turtle
# 2) Draw a circle of radius 100 pixels
turtle.circle (100)
# 3) Inform turtle that we are done and leave the window open on screen
turtle.done ()
```

1) Start a New File so you will type your program: Click **File, New File**

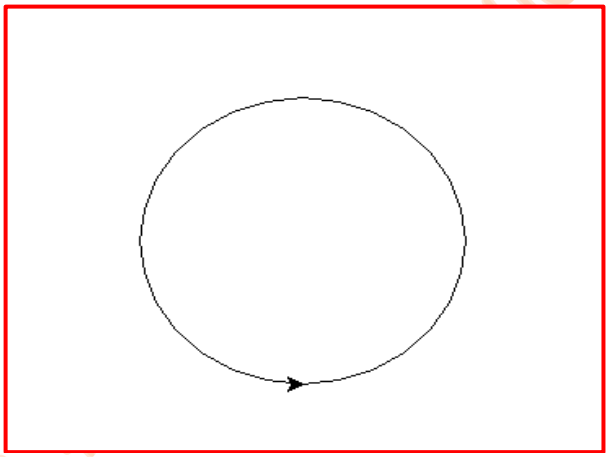
2) Type the following DrawSquare.py Python Code:

3) Save Python File as **'TurtleDrawCircleUsingNamedConstants.py'**

```
# To draw a Circle
# Date: 10/21/2019
# Programmer: Ogar Haji
# 1) import the turtle module or library
import turtle
# 2) Draw a circle of radius 100 pixels
turtle.circle (100)
# 3) Inform turtle that we are done and leave the window open on screen
turtle.done ()
```



The Output of the Python turtle project to Draw a Circle



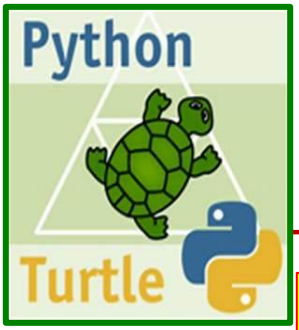
Modify This Project to Do the Following Modifications: 30%

Use Named Constants to Make the Following Changes:

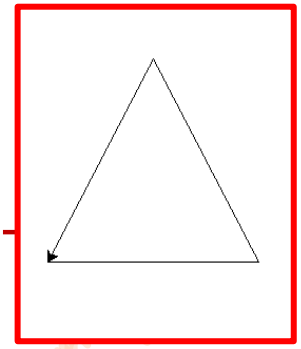
```
RADIUS = 100
turtle.circle (RADIUS)
```

```
RADIUS = 100
turtle.circle (RADIUS)
```

1) **Use Turtle to Draw a Circle of Radius 200 pixels.**
2) **Use Turtle to Draw a Circle of Radius 250 pixels.**
3) **Use Turtle to Draw a Circle of Radius 300 pixels.**
4) **Upload the Last Modified project to Brightspace.**



Python Language
Introduction to Python Turtle
To Draw a Triangle Shape
++(Do Lab Exercise 3) 100 Points ++



Do Lab Exercise 3

Lesson 132 Ex + How to Draw a Triangle using Python Turtle program?

An **Equilateral Triangle** is a **triangle** that has **all equal sides and all equal angles**.

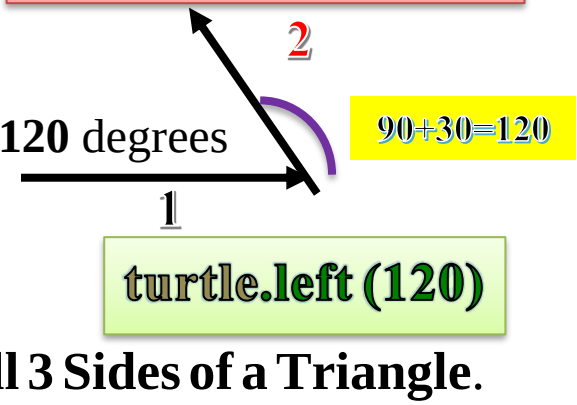
Because a **triangle's interior angles** always add up to **180°** and **180 / 3 = 60** , an **equilateral triangle** will always have **three 60° angles**.

You can **Draw a Triangle** using Python **turtle.forward (200)** and **turtle.left (120)** functions of the turtle library as shown below:

turtle.forward (200)

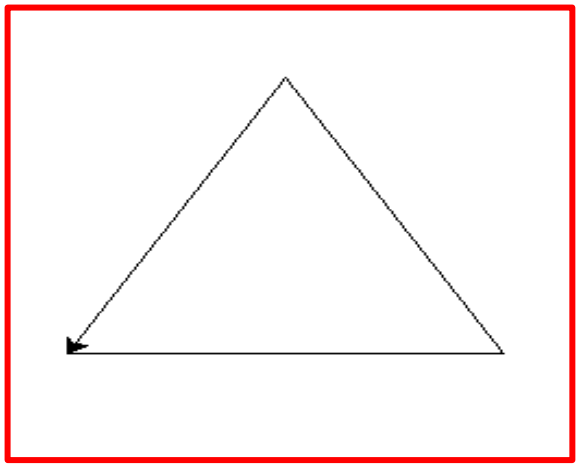
turtle.left (120) # Turn turtle left (up) by **120** degrees
 # (90 + 30 = 120)

turtle.forward (200)



Repeat the above 2 more times to draw All 3 Sides of a Triangle.

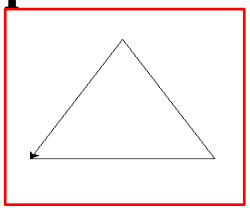
turtle.forward (200)



turtle.left (120)

This is an Explanation of What we will Do in this project:

Draw an **Equilateral Triangle** that has **all Equal Sides and all**
Equal Angles 60 Degrees.
Date: 10/21/2019
Programmer: Ogar Haji



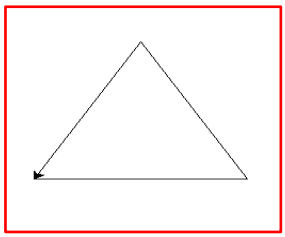
```
import turtle
turtle.forward (200)
turtle.left (90+30)
```

1) Import the turtle module or library
import turtle

2) Move the turtle forward by 200 pixels
turtle.forward (200)

3) Turn the turtle left (up) by **120** degrees **(90 + 30 = 120)**
turtle.left (90+30) **# or turtle.left (120)**

4) Move the turtle forward by 200 pixels
turtle.forward (200)



5) Turn the turtle left by 120 degrees
turtle.left (90+30) # or **turtle.left (120)**

6) Move the turtle forward by 200 pixels
turtle.forward (200)

7) Inform turtle that we are done and leave the window open on screen
turtle.done ()

+ Do Lab Exercise 3 + Do Lab Exercise 3 +

1) Start a New File so you will type your program:
2) Click **File, New File**

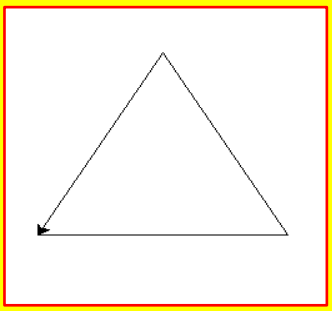
Do Lab Exercise 3

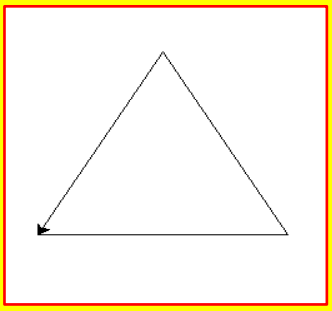
3) Type the **Python code & save File** as **'TurtleDrawTriangle.py'**

```
# Draw a Triangle
# Date: 10/21/2020
```

Do Lab Exercise 3

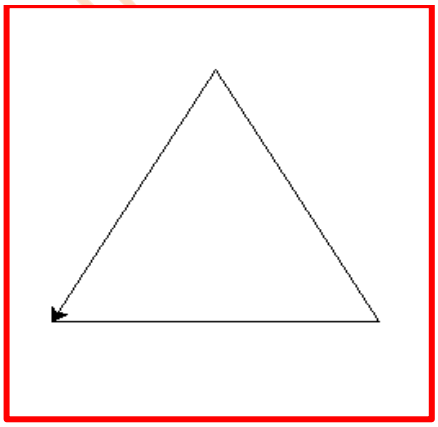
```
# Programmer: Ogar Haji
# 1) Import the turtle module or library
import turtle
# 2) Move the turtle forward by 200 pixels
turtle.forward (200)
# 3) Turn the turtle left by 120 degrees (90 + 30 = 120)
turtle.left (90+30)          # or turtle.left (120)
# 4) Move the turtle forward by 200 pixels
turtle.forward (200)
# 5) Turn the turtle left by 120 degrees
turtle.left (90+30)          # or turtle.left (120)
# 6) Move the turtle forward by 200 pixels
turtle.forward (200)
# 7) Inform turtle that we are done and leave the window open on screen
turtle.done ()
```





The Output of the project 'Draw Triangle' appears Below:

turtle.forward (300)



turtle.left (120)

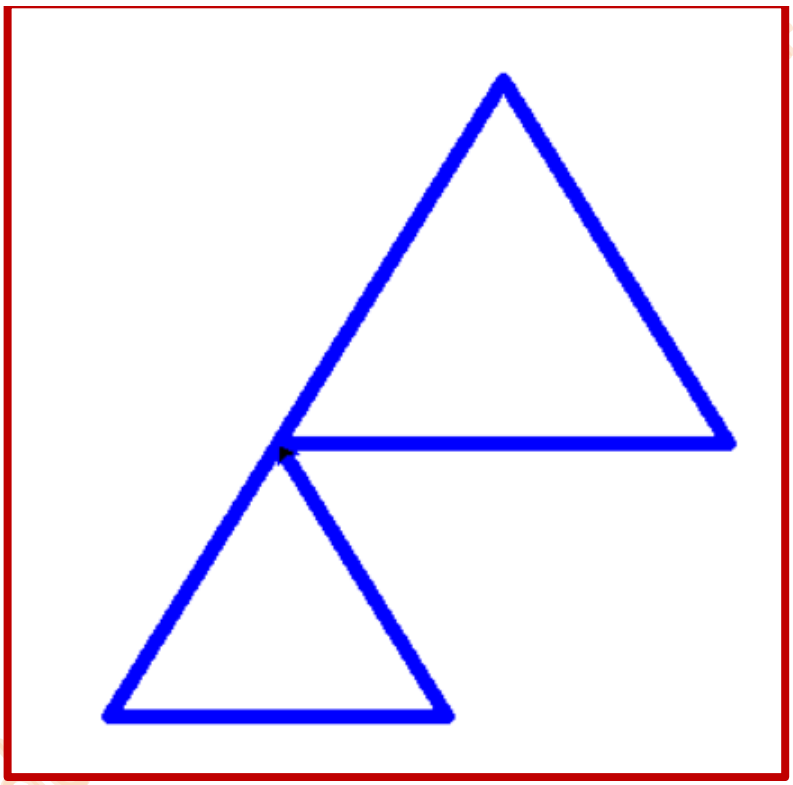
Modify This Project to Do the Following Modifications: 30%

1) **Use Turtle and Named Constants to Draw a Triangle of Length 300 pixels.**
2) **Use Turtle and Named Constants to Draw a Triangle of Length 350 pixels.**

3) **Use Turtle and Named Constants to Draw a Triangle of Length 400 pixels.**

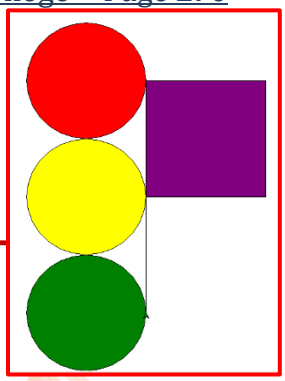
Modify This Project to Do the Following Modifications: 30%

4) **Change the turtle pensize to size 7**
5) **Change the turtle pencolor to blue color**
6) **Modify the turtle project to draw another Smaller Triangle below the previous Triangle as shown Below:**



7) **Upload the Last Modified turtle project to Brightspace Assignments Folder.**

Python Language
Introduction to Python Turtle
To Draw & Fill Shapes with Colors
+(Do Lab Exercise 4)100 Points+



Do Lab Exercise 4

Lesson 133 Ex + How to Draw Shapes and Color Shapes using Python Turtle?

To Draw and Fill Shapes with Color using Python turtle methods.

1) Use **turtle.begin_fill()** method to **begin fill Color** of a **shape**.

2) Use **turtle.fillcolor('red')** method to Fill the **shape** with **red color**.

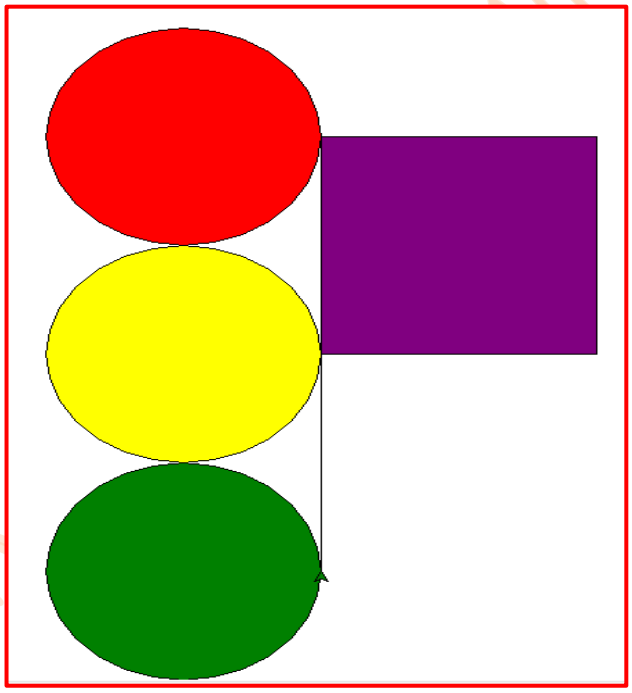
3) Use **turtle.end_fill()** method to **end** the **fill color**.

turtle.begin_fill()

turtle.fillcolor ('red')

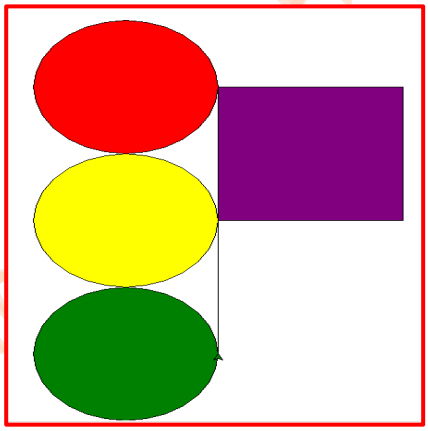
turtle.end_fill()

```
turtle.begin_fill()
turtle.fillcolor ('red')
turtle.end_fill()
```



This is an Explanation of What we will Do in this project:

Do Lab Exercise 4



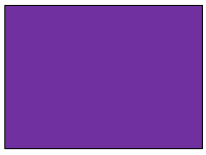
To Draw a Square and 3 Circles to simulate Traffic Lights and fill the circles with colors (Red, Yellow and Green) on screen using turtle
Date: 9/19/2021
Programmer: Ogar Haji

1) import the turtle class Library

import turtle

2) To draw the Rectangle and Fill it will purple color
Begin turtle begin_fill() function to begin fill color

turtle.begin_fill ()

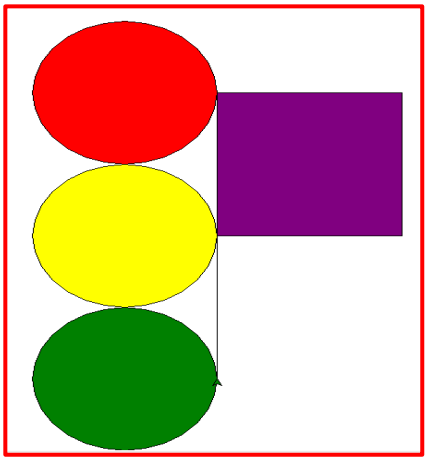


turtle.color ('black')

turtle.fillcolor ('purple')

3) Move the turtle forward by 200 pixels

turtle.forward (200)



4) Turn the turtle right by 90 degrees

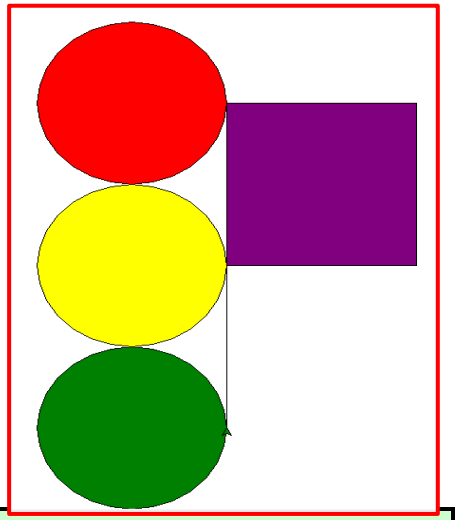
turtle.right (90)

5) Move the turtle forward by 200 pixels

turtle.forward (200)

6) Turn the turtle right by 90 degrees

turtle.right (90)



7) Move the turtle forward by 200 pixels

turtle.forward (200)

8) Turn the turtle right by 90 degrees

turtle.right (90)

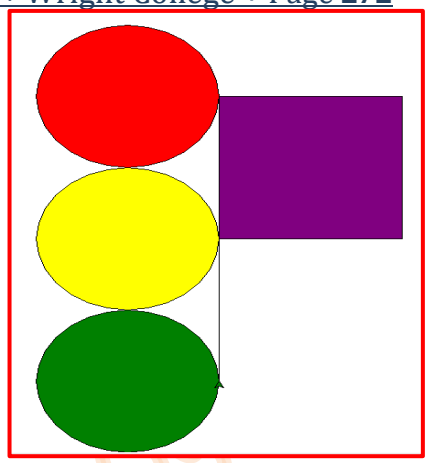
9) Move the turtle forward by 200 pixels

turtle.forward (200)

10) End the turtle end_fill color

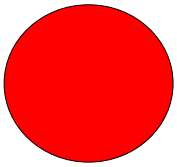
turtle.end_fill ()

11) Draw the **First Circle** and Fill the Color **Red**
to Draw a Circle of radius 100 pixels



turtle.begin_fill()

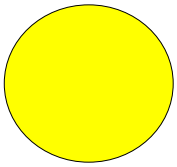
turtle.circle (100)



turtle.color ('black')
turtle.fillcolor ('red')
turtle.end_fill ()

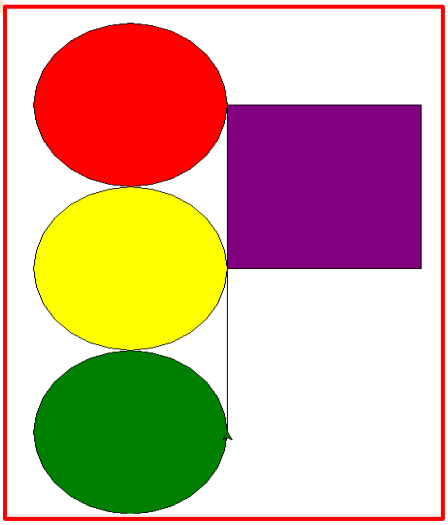
12) Draw the **Second Circle** and Fill the Color **Yellow**

turtle.begin_fill ()
turtle.backward (200)
turtle.circle (100)

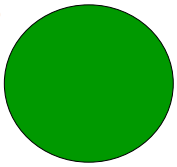


turtle.color ('black')
turtle.fillcolor ('yellow')
turtle.end_fill ()

13) Draw the **Third Circle** and Fill the Color **Green**



turtle.begin_fill ()
turtle.backward (200)
turtle.circle (100)



turtle.fillcolor ('green')
turtle.end_fill ()

14) Use **turtle.done()** function to inform the turtle you are done and
leave the window open

turtle.done ()

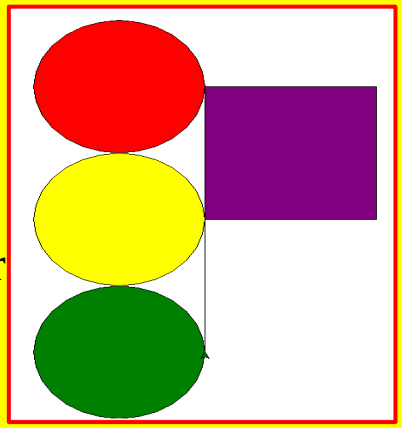
+ Do Lab Exercise 4 + Do Lab Exercise 4 +

1) Get into **Python IDLE** and **Do Lab Exercise 4**

2)Save Python turtle project as **TurtleDrawSquareAndTrafficLights.py**

To Draw a Square and 3 Circles to simulate Traffic Lights and fill the
circles with colors (red, Yellow and Green) on screen using turtle
Date: 9/19/2021
Programmer: Ogar Haji

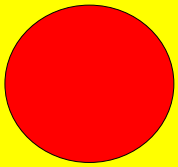
Do Lab Exercise 4



1) import the turtle module or class Library
import turtle
2) To draw the Rectangle and Fill it will purple color
Begin turtle begin_fill() function to begin fill color
turtle.begin_fill () # begin_fill to fill shape with color
turtle.color ('black')
turtle.fillcolor ('purple')

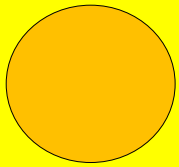
3) Move the turtle forward by 200 pixels
turtle.forward (200)
4) Turn the turtle right by 90 degrees
turtle.right (90)
5) Move the turtle forward by 200 pixels
turtle.forward (200)
6) Turn the turtle right by 90 degrees
turtle.right (90)
7) Move the turtle forward by 200 pixels
turtle.forward (200)
8) Turn the turtle right by 90 degrees
turtle.right (90)
9) Move the turtle forward by 200 pixels
turtle.forward (200)
10) End the turtle end_fill color
turtle.end_fill () # end_fill to end fill shape with color
11) Draw the **First Circle** and Fill the Color **Red**
to Draw a Red Circle of radius 100 pixels
turtle.begin_fill ()

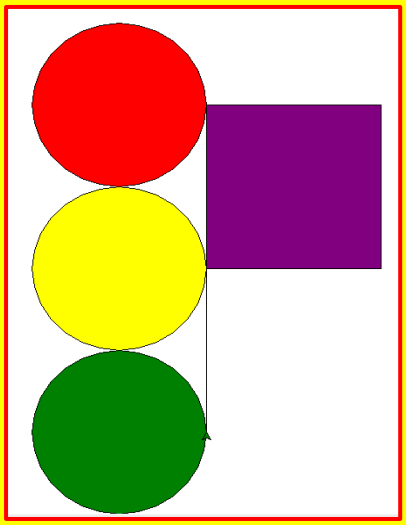
```
turtle.circle (100)
turtle.color ('black')
turtle.fillcolor ('red')
turtle.end_fill ()
```



12) Draw the **Second Yellow Circle** and Fill the Color **Yellow**

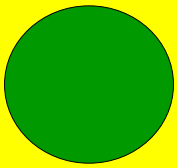
```
turtle.begin_fill ()
turtle.backward (200)
turtle.circle (100)
turtle.color ('black')
turtle.fillcolor ('yellow')
turtle.end_fill ()
```





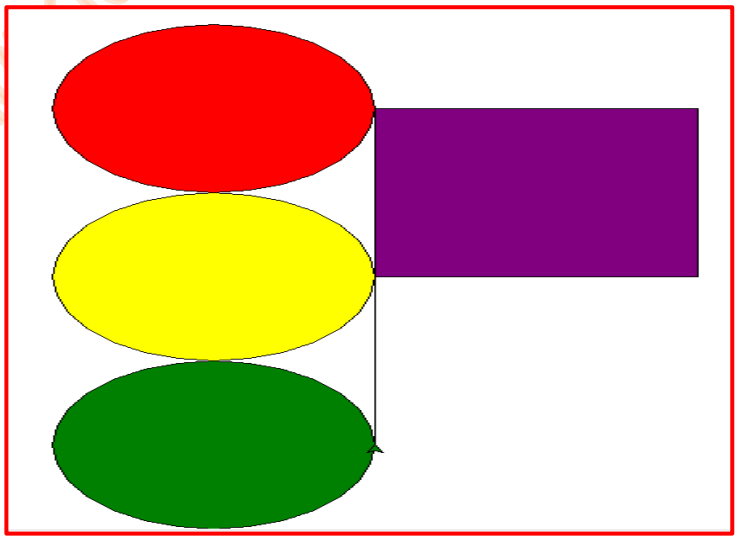
13) Draw **Third Green Circle** and Fill the Color **Green**

```
turtle.begin_fill ()
turtle.backward (200)
turtle.circle (100)
turtle.fillcolor ('green')
turtle.end_fill ()
```



```
# 14) Use turtle.done() function to inform the turtle you are done and
# leave the window open
turtle.done ()
```

The Output of this Python Turtle project to Draw Colored Shapes



Chapter 3 + Python Homework #3 Part A (Due This Sunday) 100 Points

Name:_____ CIS 103 Python Programming Language + Wright College

Answer the Following Chapter 3 Part A Python Homework: *Instructor: Ogar Haji*

1) It is always a **good programming technique** to print a **Nice Header** at the **Top** of the **project describing what the Project is about**. _____ (True/False)

2) It is always a **good programming technique** to print a **Nice Footer** at the **End** of the **project describing the End of the Project and include Python Programmer: and your Full Name**. _____ (True/False)

3) Write the Python code to **print a Blank Line**.

4) Use _____ **function** to **convert** the **string price '7.21'** to a **float number like 7.21**.

5) Use _____ **function** to **convert** the **string number_of_books '8'** to **integer number** like **8**.

6) What is a **Named Constant in Python**, explain briefly?

7) Write the Python code to **Define** a **Named Constant** called **'Sales Tax Rate'** and **assign** the **float number 0.10** to **it.**

8) You can use the **Multiplication Operator (*)** to **repeat** a **String many times** in **Python language**. _____ (True/False)

9) Write the Python code to **print the following '*' 51 Times**.

****

10) Write the Python code to **print the following 'ZZ' 41 Times**.

ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

11) What does the Python **max() function** do, explain briefly?

12) What does the Python **min() function** do, explain briefly?

13) Write the Python code to **get help** on **max()** function**.**

14) Write the Python code to **get help** on **min()** function**.**

15) Find the **result** of the following Python code. _____

>>> min (22, 14, 44, 12, 99, 77)

16) Find the **result** of the following Python code. _____

>>> max (22, 14, 44, 12, 88, 77)

17) What does **Debug the Program** in Python Language mean, explain briefly?

18) What is **Error if any** in the **following Python statement** and **Correct** it?

Print ('Hello, my name is Ogar Haji')

19) What is a **Logical Error** in a Python program, explain briefly?

20) What is **Wrong if any** in the **following Python statement** and **Correct** it?

grossPay = hours_Worked + hourly_rate

21) Write the Python code to **Declare** a **variable** called '**Student average'** and **assign** the value **98** to the variable. Then **use if …. Else** statement to **Check** if the **student average** is **Greater or equal to 70**, then **print** the message '**The Student is Passing**', **else print** the message '**The student is Failing'**.

22) What is a **Short-Circuit** in Python language, explain briefly?

23) With **Short-Circuit (and Operator)**, if **1 Condition** is **False**, then the **result** is **False** and **Python interpreter** will **Not Check** the **rest of conditions**. _____ (True/False)

24) **The Short-Circuit 'and'** Operator **increases** the **Execution** of the **program** because **once compiler finds** that **1 Condition** is **False,** then it will **ignore** to **Check** the **rest of conditions.** _____ (true/false)

25) With Conditional Operator **or , only 1 Condition** may be **True** so the **true block** will be **executed**; **otherwise**, the **false block** will be **executed**. _____ (True/False)

26) With **Short-Circuit 'or' Operator**, if **1 Condition** is **True**, then the **result** is **True** and the **Python compiler** will **Not Check** the **rest of conditions**. _____ (True/False)

27) Does the **Short-Circuit apply** to this following exercise, and why, explain briefly?

```
average = 99
if  average >= 0 and average <=100:
        print ('Average %d is Valid Number %n', average)
else:
        print ('Average %d is Invalid Number %n', average)
```

28) What is the **result** or the **print out** of the **previous example** in **question**?

29) Does the **Short-Circuit apply** to this following exercise, and why, explain briefly?

```
a = 10
b = 20
c = 30
if  a == 20 and b == 20 and c == 30:
        print  ('All Conditions are True')
else:
        print ('One Condition is False')
```

30) What is the **result** or the **print** of the previous question?

31) Does the **Short-Circuit apply** to this following exercise, and why, explain briefly?

```
average = 177
if average < 0 or average >100:
    print ('Average %d is Invalid Number', average)
else:
    print ('Average %d is Valid Number', average)
```

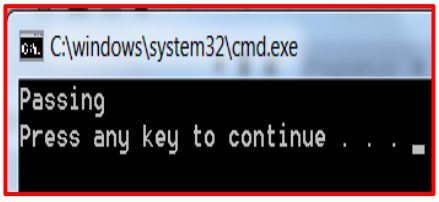
32) What is the **result** or the **print** of the previous example in previous **question**?

33) Does the **Short-Circuit apply** to this following exercise, and why, explain briefly?

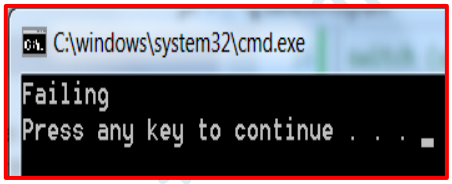
```
a = 10
b = 20
c = 30
if  a == 10 or b == 25 or c == 35:
    print ('One Conditions is True')
else:
    print ('One Condition is False')
```

34) What is the **result** or the **print** of the previous **question**?

35) Write the Python code to use an **if statement to check if a condition is true**?
Declare a variable called **average** and **assign** the **value 100** to it and then **print out** the message '**Passing**' to the **output**?



36) Write the Python code to use **if…else statement to check if a condition is true or false**? **Declare a variable** called **average** and **assign** the **value 55** to it and then writing the message '**Passing**' to the **output if the average is Greater or equal to 70** and **'Failing' if false**?



37) Write the Python code to use **if…else statement with multiple else if statements to assign the final grade a letter 'A', 'B', 'C', 'D', or 'F. according to the following criteria:**

Assign the Final Grade according to the following **Average** Criteria:

90 - 100 Grade A
80 - 89 Grade B
70 - 79 Grade C
50 - 69 Grade D
0 - 49 Grade F

38) Write the Python code to **calculate Overtime Gross Pay if** the **employee works more than 40 Hours**.

39) Write Python code to use the Conditional **And** operator to **check if average** is **between 0 and 100 inclusive**, then **print** the **message** '**Average is Valid Number**', **else print** the **message** '**Average is Invalid Number**'.

40) Write the Python code to use the Conditional **Or** operator to **check if the average is less than 0** or **average Greater than 100**, then **print** to the **output** the **message** '**Average is Invalid Number**', **else print** to the **output** the **message** '**Average is Valid Number**'.

41) Write the **and Truth Table** for the following:

True and True result is _____

True and False result is _____

False and True result is _____

False and False result is _____

42) Write the **or Truth Table** for the following:

True or True result is _____

True or False result is _____

False or True result is _____

False or False result is _____

43) What is the **not Operator**?

44) Write the Python code to use the **Negator operator (not)** to **check** the **average** of a student and **whether** the **student** is **Passing or Failing**.

45) What is the **Ternary Operator** in Python, explain?

46) What is the **strftime()** method in Python, explain?

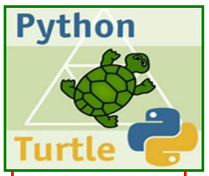
47) Write the Python code to **Import** the **date classes** in Python languages.

48) Write the Python code to Create an **instance object** called '**today**' from the module **date.today()** in Python languages.

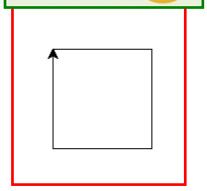
49) Write the Python code to print out the instance object **today** to output console.

```
= RESTART: C:/Users/ogarh/App
ate.py
Today's Date is:  2021-09-06
>>>
```

50) Write the Python turtle command to **include or import turtle module** so you can work with turtle commands in Python.

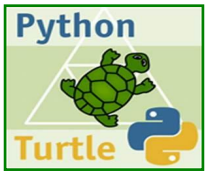


51) Write the Python turtle command to **Move** the **turtle forward by 120 pixels**.

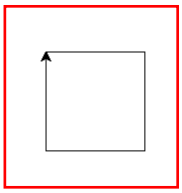


52) Write the Python turtle command to **Turn** the **turtle Left by 90 degrees**.

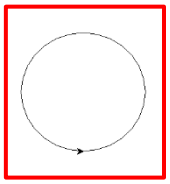
53) Write the Python turtle command to **Turn** the **turtle Right by 150 degrees**.



54) Write the Python turtle code to **Draw** a **Square** of **length 400 pixels**.

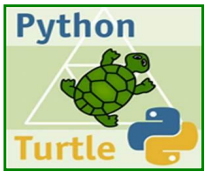


55) Write the Python turtle code to **Draw** a **Circle** of **Radius 500 pixels**.



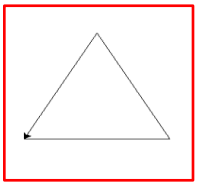
56) Write the Python turtle code to **Change** the **turtle pen color** to **'red' color**.

57) Write the Python turtle code to **Change** the **turtle pen size** to **size 7**.

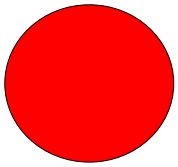


58) Write the Python turtle code to **inform turtle** that you **are done and leave the shape drawn on the screen**.

59) Write the Python turtle code to Draw an **Equilateral Triangle** that has **all Equal Sides** and **all Equal Angles of 60 Degrees.**



60) Write the Python turtle code to **Draw** a **Circle** with a **Radius** of **100 pixels** and **fill** the **circle** with **Red color**.



CIS 103 Python Inatructor: Ogar Haji

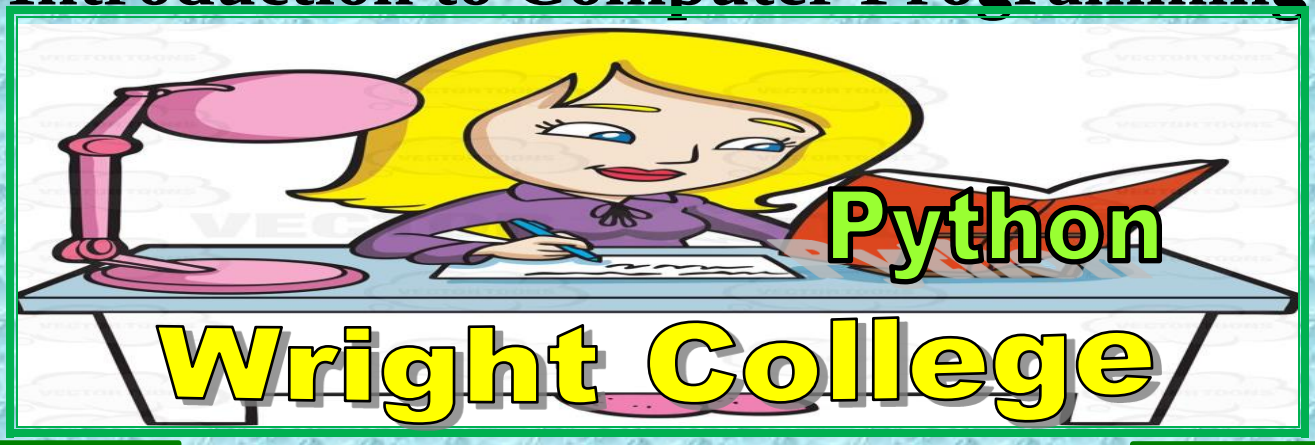
Wright College +

Chapter 3 Part B

Conditional Statements: Using if...else and Logical Operators: ('and', 'or' & 'not') and Python Turtle Module

CIS103 Python Programming Language

Introduction to Computer Programming



'Hands-On' Mastering Computer Logic, Design and Programming Using Python Language

Written By:

OGAR HAJI

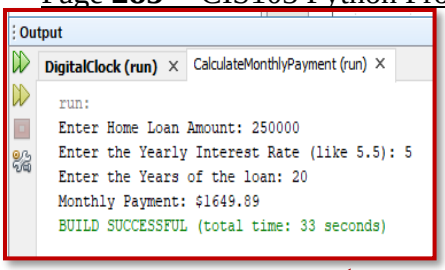
Master's Degree in Computer Science
DePaul University ✚ Chicago, Illinois, USA

Date Published: January 29, 2023

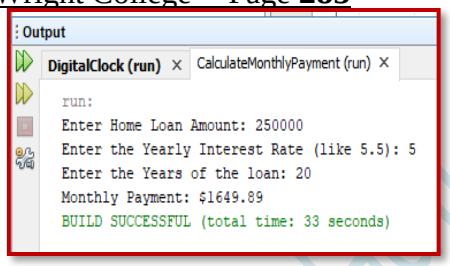
Chapter 03+Part B

You will learn the following in Chapter 3 Part B:

- Named Constants in Python: SALES_TAX_RATE = 0.09
- Using Multiplication Operator (*) to print letter 'Z' 25 Time
- Using max() and min() functions to find the Maximum Number and the Minimum Number.
- Using Single **if** statement to Check if a Condition is **true.**
- Using Single **if…else** statements to Check if a Condition is **true** or **false** in Python
- Using the Short-Circuit Conditional 'and' and 'or' operators in Python Language.
- Calculate the Average of 2 Tests and Assign Final Grade. Lab Exercise
- Insert Computer Date and Time in Python + Lab Exercise.
- Calculate Monthly Payment of a Loan Python Project
- Python Language to Draw a Square with Colored Lines (red, green, blue, gold)
- Chapter 3 + Python Homework #3 (Due Next Week)
- Do Python Lab Assignments 3A + American Cars Rental



Calculate Monthly Payment of a Loan Python Project



+++(Do Lab Exercise 5) 100 Points+++

Do Lab Exercise 5

Lesson 134 Ex + How to Calculate Loan Monthly Payment Python Project ?

Problem or Project: Design and Code in Python Language the project to Calculate the Monthly Payment of a Loan.

To print the **Monthly Payment formatted with Comma ,** and **2 Decimal Places use {:,.2f}.format(monthly_payment) as shown below:**

print ('Monthly Payment: $ {:,.2f}'.format (monthly_payment))

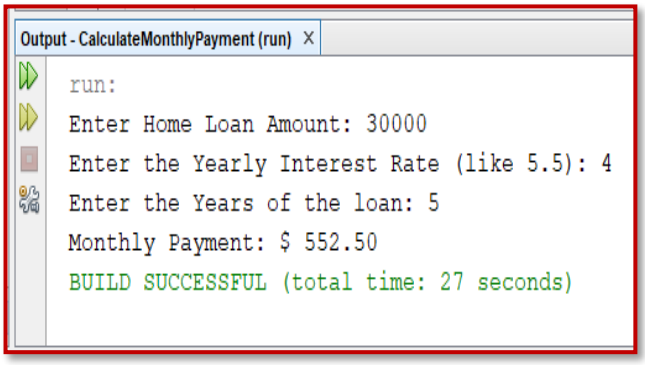
Use the **Math Power math.pow () method** to **Calculate Monthly Payment:**

monthlyPayment = (loanAmount * interestRate) /

(1 - math.pow (1 + interestRate, -months))

Do the 12 Must Steps to Design, Code and Solve a project using Python.

1) State the Purpose of the project.
2) State how the Input will look.
3) State what Calculations to be Done.
4) State how the Output will Look
5) Either Draw a Flowchart or
6) Write a Pseudocode



7) Code the project in Python Language and start Python IDLE.
8) Run the project and make sure the Calculations are correct.

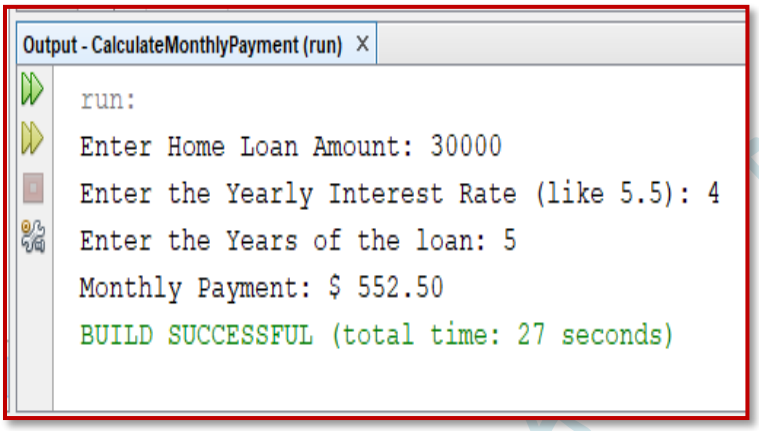
This Python project 'Calculate Loan Amount' will Calculate the Monthly Payment for a Loan:

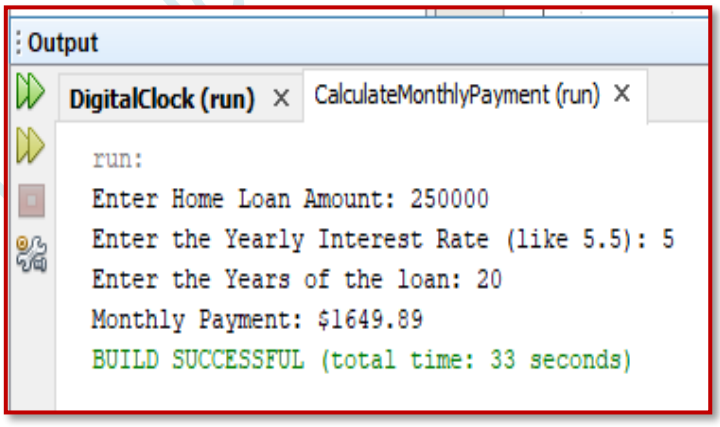
1) It will **prompt** the **User** to **Enter** the **Laon Amount: 250000**
2) It will **prompt** the **user** to **Enter** the **Yearly Interest Rate: 5**
3) It will **prompt** the **user** to **Enter** the **Years of the loan: 20**

Then the **Python project** will **calculate** the **Compound Monthly Payment**.
Use the Math Power **math.pow ()** **method** to **Calculate Monthly Payment:**

monthlyPayment = (loanAmount * interestRate) /
(1 - math.pow (1 + interestRate, -months))

The Python project will Display the following to Calculate Monthly Payment





This is an Explanation of What we will Do in this project:

'''
This project will Calculate the Monthly Loan Payment for a Loan.
The project will prompt the user to Enter the Loan Amount, the Interest Rate and the Years of the loan.
Date: 06/01/2021
Programmer: Ogar Haji (Python Instructor)
'''

```
Enter Home Loan Amount: 30000
Enter the Yearly Interest Rate (like 5.5): 4
Enter the Years of the loan: 5
Monthly Payment: $ 552.50
```

1) import the math library function to be used in calculation of loan
import math

2) Prompt the user to enter the loan_amount, read it and store it
loan_amount = input ('Enter Home Loan Amount: ')

3) Convert loan amount to float using **float()** function
loan_amount = float (loan_amount)

4) Prompt the user to enter the Yearly Interest Rate, read it and store it
yearly_interest_rate= input ('Enter the Yearly Interest Rate (like 5.5): ')

5) Convert yearly interest rate to float using **float()** function
yearly_interest_rate = float (yearly_interest_rate)

6) Calculate Monthly intertest rate by dividing
yearly interest rate by 100 then divide the result by 12 months
monthly_interest_rate = (yearly_interest_rate / 100) / 12

7) Prompt the user to enter the Years, read it and store it
years = input ('Enter the Years of the loan: ')

```
Enter Home Loan Amount: 30000
Enter the Yearly Interest Rate (like 5.5): 4
Enter the Years of the loan: 5
Monthly Payment: $ 552.50
```

8) Convert years to int using int() function
years = int (years)

9) Convert years to months by multiplying years * 12
months = years * 12

10) Calculate the Compound Monthly Payment
using the Math Power **Math.pow()** method
monthly_payment = (loan_amount * monthly_interest_rate) / (1 - math.pow(1 + monthly_interest_rate, -months))

11) Print out the Monthly Payment formatted
print ('Monthly Payment: $ {:,.2f}'.format (monthly_payment))

+ Do Lab Exercise 5 + Do Lab Exercise 5 +

Do Lab Exercise 5

1) Launch **Python IDLE IDE** program
2) Type **Python lab exercise**, **Save** as **CalculateLoanMonthlyPayment.py**

```
'''
This project will Calculate the Monthly Loan Payment for a Loan.
The project will prompt the user to Enter the Loan Amount, the Interest Rate and the Years of the loan.
Date: 06/01/2021
Programmer: Ogar Haji (Python Instructor)
'''
```

Do Lab Exercise 5

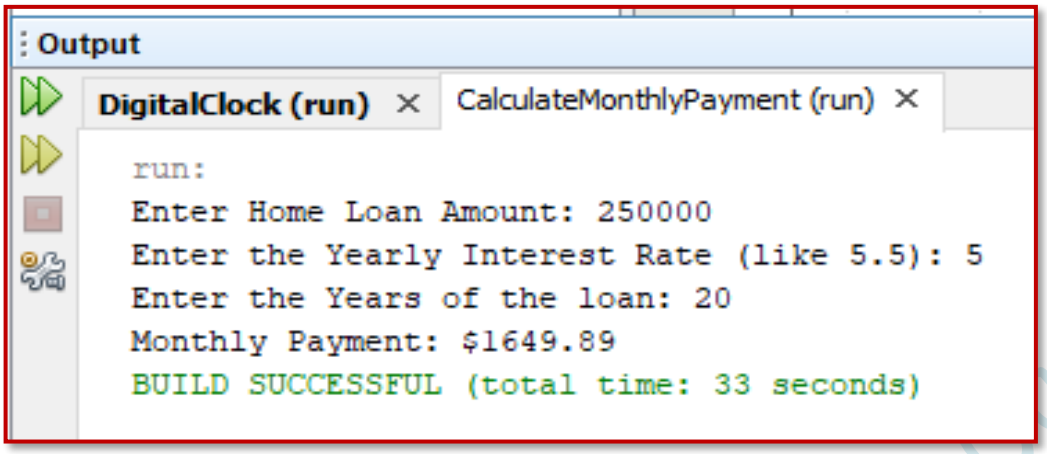
```
# 1) import the math library function to be used in calculation of loan
import math
# 2) Prompt the user to enter the loan_amount, read it and store it
loan_amount = input ('Enter Home Loan Amount: ')
# 3) Convert loan_amount to float using float() function
loan_amount = float (loan_amount)
# 4) Prompt the user to enter the Yearly Interest Rate, read it and store it
yearly_interest_rate = input ('Enter the Yearly Interest Rate (like 5.5): ')
# 5) Convert yearly interest rate to float using float() function
yearly_interest_rate = float(yearly_interest_rate)
# 6) Calculate Monthly intertest rate by dividing
# yearly interest rate by 100 then divide the result by 12 months
monthly_interest_rate = (yearly_interest_rate / 100)  / 12
# 7) Prompt the user to enter the Years, read it and store it
years = input ('Enter the Years of the loan: ')
# 8) Convert years to int using int() function
years = int (years)
# 9) Convert years to months by multiplying years * 12
months = years * 12
# 10) Calculate the Compound Monthly Payment
#  using the Math Power math.pow() method
monthly_payment = (loan_amount * monthly_interest_rate) / (1 -
math.pow (1 + monthly_interest_rate, -months))
# 11) Print out the Monthly Payment formatted
print ('Monthly Payment: $ {:,.2f}'.format(monthly_payment))
```

```
Enter Home Loan Amount: 30000
Enter the Yearly Interest Rate (like 5.5): 4
Enter the Years of the loan: 5
Monthly Payment: $ 552.50
```


```
Enter Home Loan Amount: 30000
Enter the Yearly Interest Rate (like 5.5): 4
Enter the Years of the loan: 5
Monthly Payment: $ 552.50
```


The Input/Output of Calculate Monthly Payment Project:

```
Output - CalculateMonthlyPayment (run)
run:
Enter Home Loan Amount: 30000
Enter the Yearly Interest Rate (like 5.5): 4
Enter the Years of the loan: 5
Monthly Payment: $ 552.50
BUILD SUCCESSFUL (total time: 27 seconds)
```



Modify This Project to Do the Following Modifications: 30%

1) **Modify the project by Printing Nice Headers and Footers.**
2) **Insert Todays' Date and Time in the Headers at top of the output.**
3) **Add the following at the end of program before printing the Footers.**
4) **Prompt the user to 'Enter the First Name: ' and read and store it.**
5) **Prompt the user to 'Enter the Last Name: ' and read and store it.**
6) **Prompt user to 'Enter the Yearly Income: ' and read and store it.**
7) **Prompt user to 'Enter the Years on the Job: ' and read and store it.**
8) **Check if the Yearly Income is >= 25,000 and Years on the Job >=5, then the Applicant is Qualified for the Loan, otherwise the Applicant is Not Qualified for the Loan.**

```
if  yearly_income >= 25000 and years_on_job >=5:
        print ('The Applicant is Qualified for the Loan. ')
else:
        print ('The Applicant is NOT Qualified for the Loan. ')
```

```
Enter the Year as a Number: 2021
Enter the Month as a Number: 10

    October 2021
Mo Tu We Th Fr Sa Su
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

Display Yearly Calendar On Screen Project

(Do Lab Exercise 6) 100 Points

Do Lab Exercise 6


```
Enter the Year as a Number: 2021
Enter the Month as a Number: 10

    October 2021
Mo Tu We Th Fr Sa Su
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```


Lesson 135 Ex + Display Yearly Calendar on Screen in Python language.

Problem or Project: Design and Code in Python Language the project to Display and Print the Calendar on Screen.

To Print a Yearly Calendar with a **specific Year** like **2022**, you call the **calendar.calendar (2022)** function and pass the **parameter year (2022)**.

1) You must first import the calendar module

import calendar

2) Declare a variable called 'year' and initialize it to 2022

year = 2022

2) Print out the calendar and pass to calendar(year) the year parameter

print (calendar.calendar (year))

Do the Following 12 Steps to Design, Create and Code a Project in Python:

Do Steps 1 thru 7 in your Note Book or on Paper.

The Input/Output of the Calendar is the following:

```
                                  2022

      January                   February                  March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
                1  2         1  2  3  4  5  6         1  2  3  4  5  6
 3  4  5  6  7  8  9       7  8  9 10 11 12 13       7  8  9 10 11 12 13
10 11 12 13 14 15 16      14 15 16 17 18 19 20      14 15 16 17 18 19 20
17 18 19 20 21 22 23      21 22 23 24 25 26 27      21 22 23 24 25 26 27
24 25 26 27 28 29 30      28                        28 29 30 31
31

      April                     May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
             1  2  3                        1                 1  2  3  4  5
 4  5  6  7  8  9 10       2  3  4  5  6  7  8       6  7  8  9 10 11 12
11 12 13 14 15 16 17       9 10 11 12 13 14 15      13 14 15 16 17 18 19
18 19 20 21 22 23 24      16 17 18 19 20 21 22      20 21 22 23 24 25 26
25 26 27 28 29 30         23 24 25 26 27 28 29      27 28 29 30
                          30 31
```

```
      July                      August                    September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
                1  2  3       1  2  3  4  5  6  7                 1  2  3  4
 4  5  6  7  8  9 10       8  9 10 11 12 13 14       5  6  7  8  9 10 11
11 12 13 14 15 16 17      15 16 17 18 19 20 21      12 13 14 15 16 17 18
18 19 20 21 22 23 24      22 23 24 25 26 27 28      19 20 21 22 23 24 25
25 26 27 28 29 30 31      29 30 31                  26 27 28 29 30

      October                   November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
                1  2         1  2  3  4  5  6                 1  2  3  4
 3  4  5  6  7  8  9       7  8  9 10 11 12 13       5  6  7  8  9 10 11
10 11 12 13 14 15 16      14 15 16 17 18 19 20      12 13 14 15 16 17 18
17 18 19 20 21 22 23      21 22 23 24 25 26 27      19 20 21 22 23 24 25
24 25 26 27 28 29 30      28 29 30                  26 27 28 29 30 31
31
```

This is an Explanation of What we will Do in this project:

This project will display a Yearly Calendar on Screen
Date: 10/12/2021
Programmer: (Python Instructor) Ogar Haji

1) Import the calendar module or library
import calendar

2) Declare a variable called year and assign a value to it
year = 2022

3) Call the calendar.calendar(year) method and pass the year to it
print (calendar.calendar (year))

+ Do Lab Exercise 6 + Do Lab Exercise 6 +

1) Ex- Type the following Python Code in Python IDLE IDE and
2) Save it as **PrintYearlyCalendar.py**

Do Lab Exercise 6

```
# This project will display a Yearly Calendar on Screen
# Date: 10/12/2017
# Programmer: (Python Instructor) Ogar Haji

# 1) Import the calendar module or library
import calendar

# 2) Declare a variable called year and assign a value to it 2022
year = 2022

# 3) Call the calendar.calendar(year) function and pass the year to it
print (calendar.calendar (year) )
```

The Input/Output of the Calendar is the following:

```
                                  2022

      January                   February                  March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
                1  2       1  2  3  4  5  6       1  2  3  4  5  6
 3  4  5  6  7  8  9       7  8  9 10 11 12 13       7  8  9 10 11 12 13
10 11 12 13 14 15 16      14 15 16 17 18 19 20      14 15 16 17 18 19 20
17 18 19 20 21 22 23      21 22 23 24 25 26 27      21 22 23 24 25 26 27
24 25 26 27 28 29 30      28                        28 29 30 31
31

      April                     May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
             1  2  3                         1                 1  2  3  4  5
 4  5  6  7  8  9 10       2  3  4  5  6  7  8       6  7  8  9 10 11 12
11 12 13 14 15 16 17       9 10 11 12 13 14 15      13 14 15 16 17 18 19
18 19 20 21 22 23 24      16 17 18 19 20 21 22      20 21 22 23 24 25 26
25 26 27 28 29 30         23 24 25 26 27 28 29      27 28 29 30
                          30 31

       July                    August                  September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
                1  2  3       1  2  3  4  5  6  7                 1  2  3  4
 4  5  6  7  8  9 10       8  9 10 11 12 13 14       5  6  7  8  9 10 11
11 12 13 14 15 16 17      15 16 17 18 19 20 21      12 13 14 15 16 17 18
18 19 20 21 22 23 24      22 23 24 25 26 27 28      19 20 21 22 23 24 25
25 26 27 28 29 30 31      29 30 31                  26 27 28 29 30

      October                   November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
                1  2       1  2  3  4  5  6                 1  2  3  4
 3  4  5  6  7  8  9       7  8  9 10 11 12 13       5  6  7  8  9 10 11
10 11 12 13 14 15 16      14 15 16 17 18 19 20      12 13 14 15 16 17 18
17 18 19 20 21 22 23      21 22 23 24 25 26 27      19 20 21 22 23 24 25
24 25 26 27 28 29 30      28 29 30                  26 27 28 29 30 31
31
```

Modify This Project to Do the Following Modifications: 30%

Modify the Python Project to do the following:

1) **Prompt the user to Enter the Year for the Calendar.**
2) **Check for Invalid Year which is Less than 2000 or Greater than 3000, then print a Message of '**** Invalid Year ****'**

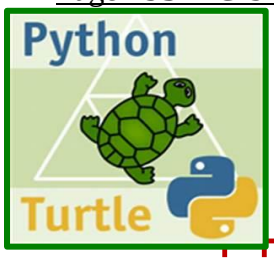
2) Declare a variable called year and prompt user for input

```
year = input ('Enter the Year of the Calendar: ')
year = int (year)
```

3) Check for Invalid Year using **or** operator

```
if year < 2000 or year > 3000:
    print ('****** Invalid Year *******')
else:
    # 4) Call the calendar.calendar(year) method and pass year to it
    print (calendar.calendar (year) )
```

```
= RESTART: C:\Users\ogarh\AppData\Loc
Enter the Year of the Calendar: 1988
****** Invalid Year *******

= RESTART: C:\Users\ogarh\AppData\Loc
Enter the Year of the Calendar: 3333
****** Invalid Year *******
```



Python Language
Introduction to Python Turtle
To Draw Shapes

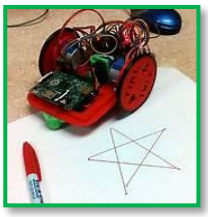
+++(Read and Study This Lesson)+++

Lesson 136 + How to Draw Shapes using Python Turtle program?

Turtle is a **Python Module** or **Library** which is used to **Create** and **Design Shapes, graphics, pictures, games** and **Animations**.

Python Turtle was developed by **Wally Feurzeig, Seymour Parpet** and **Cynthina Slolomon** in **1967**.

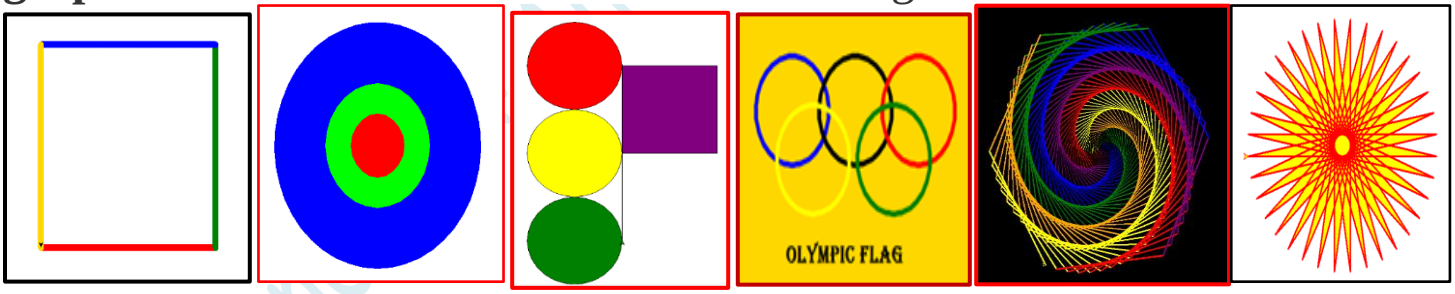
It was a part of the **original Logo programming language**.



In the **late 1960s**, **MIT professor Seymour Papert** used a **Robotic 'turtle'** to **teach programming to students**.

'turtle' is a **Python module** or **feature** like a **drawing board**, which lets us **command** a **turtle** to **draw all over the paper**.

By using the **commands** in **Python turtle**, you can **draw nice shapes**, **graphics** and **do animations** like the following:

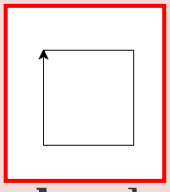


- **Python turtle** is a **Module** or **Library** which is used to **create shapes, patterns,** and **Games** on the Screen.
- **Python turtle** is initially **positioned** in the **Center** of the **Graphic Window** and the **turtle** looks like an **Arrow Head ▶**.
- **Python turtle** default heading is **0 degrees** and facing **East**.

```
import turtle
turtle.forward(100)
```

Some of Python turtle() functions:

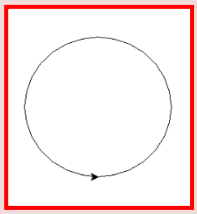
- **Turtle** is a **python feature** like a **drawing board**, which **allows** you **to command** a **turtle** to **draw Lines** and **make Shapes** and **Games**.
- **1)** To **use turtle**, we must **import turtle** module first.



```
import turtle
```

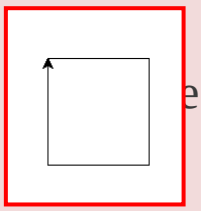
- **2)** Use **turtle.forward (100)** function to **move forward** and **draw** a **line** of **length 100 pixels**. **(96 Pixels are equal to 1 inch)**
- **3)** Use **turtle.left (90)** function to **turn the turtle left** by **90 degrees (left is Up the window screen)**.
- **4)** Use **turtle.right (90)** function to **turn the turtle Right** by **90 degrees (right is Down the window screen)**.
- **5)** Use **turtle.setheading (90)** function to **set the turtle heading** to **90 degrees**.

- **6)** Use **turtle.heading ()** function to **get and display the turtle current heading**.
- **7)** Use **turtle.penup ()** function to **raise** the **turtle pen up** so **No Drawing will take effect**.
- **8)** Use **turtle.pendown ()** function to **lower the turtle pen down and will start Drawing**.



- **9)** Use **turtle.circle (25)** function to **Draw a Circle** of **radius 25 pixels**.
- **10)** Use **turtle.pensize (5)** function to **set** the **turtle pensize** to **size 5 pixels**. **(a little bit thicker than 1 pixel the default pensize)**
- **11)** Use **turtle.dot ()** function to **draw a Dot (.)** on window screen.

12) Use **turtle.pencolor ('red')** function to change the **pen color** to **red color**. (**Black** color is the **default pencolor**)



13) Use **turtle.bgcolor ('yellow')** function to change the **background color** of the window screen to **yellow color**.

14) Use **turtle.setup (640,480)** function to **set the size** of the **window screen** to **Width 640 pixels** and **Height** to **480 pixels**.

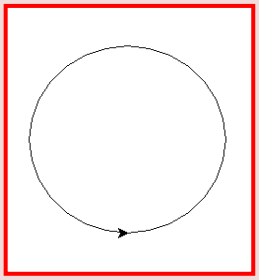
15) Use **turtle.goto (0, 100)** function to **move** the **turtle pen** to **location 0, 100** which **is x and y coordination's**.

16) Use **turtle.xcor ()** function to **return** the **x coordination** of the **current position** of **turtle pen**.

17) Use **turtle.ycor ()** function to **return** the **y coordination** of the **current position** of **turtle pen**.

18) Use **turtle.speed (0)** function to change turtle **Animation speed** of the **turtle pen drawing** to **0**. (**0** is the **Fastest speed with No Animation**). Turtle **Speed settings** can be set between **1 (Slowest)** to **10 (Fastest)**.

turtle.speed (0) is the Super Fastest and No Animation.
turtle.speed (10) is the Fast speed.
turtle.speed (6) is the Normal speed.
turtle.speed (3) is Slow speed.
turtle.speed (1) is the Slowest speed.



19) Use **turtle.hide ()** function to **hide** the **turtle arrow head**.

20) Use function **turtle.showturtle ()** to **show turtle arrow head**.

21) Use **turtle.write ('Ogar Haji')** function to **write the literal string 'Ogar Haji'** on the Window screen.

22) Use **turtle.begin_fill ()** function to **begin** to **fill** a **shape** with **color**

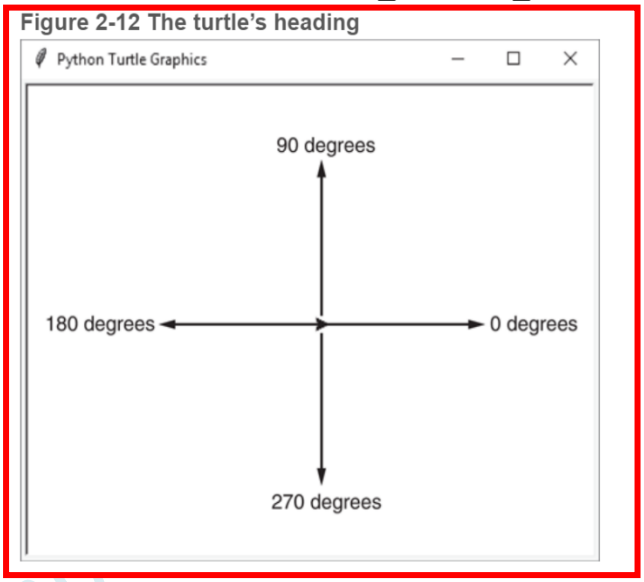
- **23)** Use **turtle.fillcolor ('green')** function to **fill the shape** with **color green**.
- **24)** Use **turtle.end_fill ()** function to **end** to **fill** a **shape** with **color**
- **25)** Use **turtle.shape ('turtle')** to **display** the **turtle shape instead** of **Arrow shape.**
- **26)** Use **turtle.home()** function to **move** the **turtle** to the **origin coordinates (0,0)**.
- **27)** Use **turtle.done ()** function to Inform turtle that **we are done** and **leave the window open on screen**.

turtle.done()

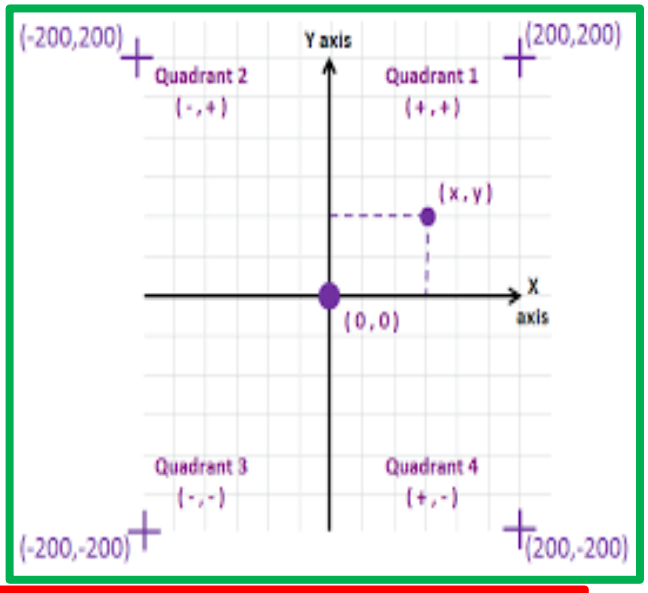
- The **python turtle Module or Library contains** all the **methods** and **functions** that we need to **create Shapes and Animation on screen**.
- This shape is the **Turtle Heading** ▶ which is **Facing toward East**.

Turtle Left and Right Degrees

Figure 2-12 The turtle's heading

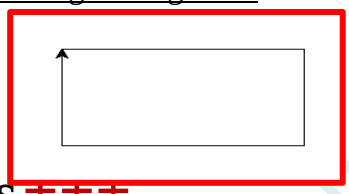


Turtle X and Y Coordinates



```
import turtle
turtle.forward(100)
```

Python Turtle Language
To Draw a Rectangle Shape



+++(Do Lab Exercise 7) 100 Points +++

Do Lab Exercise 7

Lesson 137 Ex + How to Draw a Rectangle using Python Turtle program ?

You can **Draw a Rectangle shape** of **length = 200** pixels and **Width** of **100** pixels by using **turtle.forward (200)** and **turtle.right (90)** and **turtle.forward (100)** functions of turtle library as shown below:

turtle.forward (200)
turtle.right (90)
turtle.forward (100)

And repeat the above 4 times to draw All 4 Sides of a Rectangle.

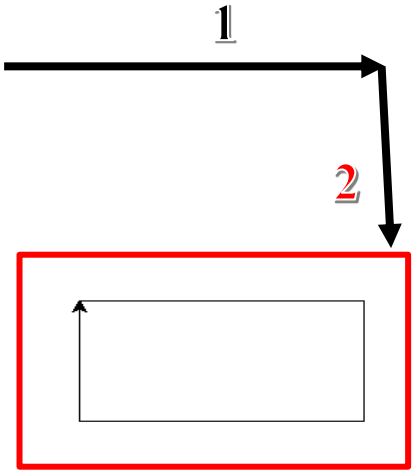
But you have to import the Python turtle library first.

import turtle

This is an Explanation of What we will Do in this project:

 # We will do the following to Draw a Rectangle in Python Turtle

To Draw a Square on screen using turtle with **length of 200** pixels and **width of 100** pixels.



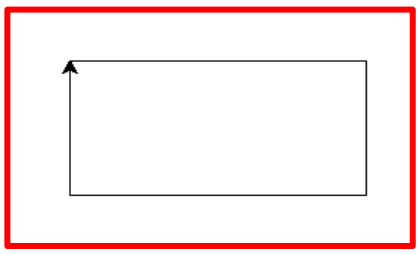
Date: 9/19/2021
Programmer: Ogar Haji
1) import the turtle class Library

import turtle

2) Move the turtle forward by 200 pixels

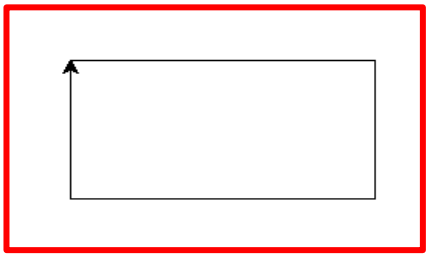
turtle.forward (200)

3) Turn the turtle right or down by 90 degrees
turtle.right (90)



4) Move the turtle forward by 100 pixels
turtle.forward (100)

5) Turn the turtle right by 90 degrees
turtle.right (90)



6) Move the turtle forward by 100 pixels
turtle.forward (200)

7) Turn the turtle right by 90 degrees
turtle.right (90)

8) Move the turtle forward by 100 pixels
turtle.forward (100)

Do Lab Exercise 7

+ Do Lab Exercise 7 + Do Lab Exercise 7 +

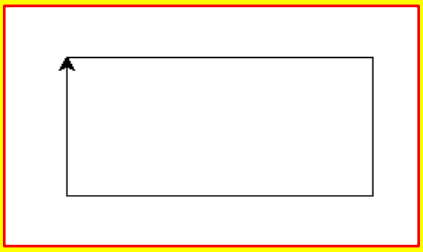
4) Start a New File so you will type your program: Click **File, New File**

5) Type the following DrawRectangle.py Python Code:

6) Save the **Python File** as **'TurtleDrawRectangle.py'**

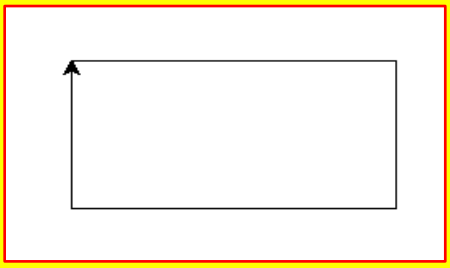
```
# To Draw a Rectangle on screen using turtle with length of 200 pixels and
# width of 100 pixels
# Date: 9/19/2021
# Programmer: Ogar Haji
# 1) import the turtle module or class Library
import turtle
# 2) Move the turtle forward by 100 pixels
turtle.forward (200)
# 3) Turn the turtle right or down by 90 degrees
turtle.right (90)
# 4) Move the turtle forward by 100 pixels
turtle.forward (100)
```

Do Lab Exercise 7



5) Turn the turtle right or down by 90 degrees
turtle.right (90)

6) Move the turtle forward by 100 pixels
turtle.forward (200)

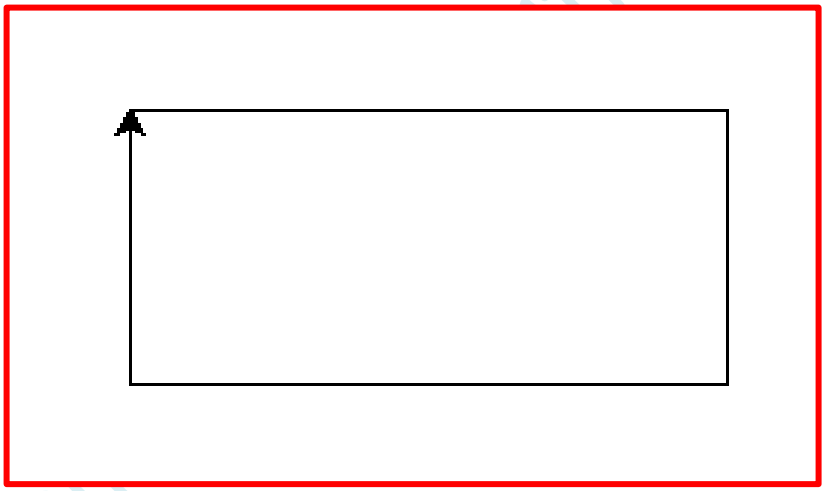


7) Turn the turtle right by 90 degrees
turtle.right (90)

8) Move the turtle forward by 100 pixels
turtle.forward (100)

9) Inform turtle that we are done and leave the window open on screen
turtle.done ()

The Output of the Python turtle project to Draw a Rectangle



Modify This Project to Do the Following Modifications: 30%

Note: Use Named Constants: LENGTH and WIDTH

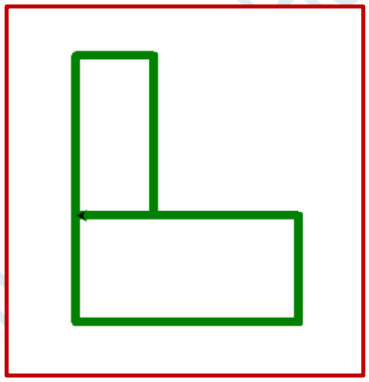
1) Draw a Rectangle of Length 300 pixels and Width of 200 pixels.

2) Draw a Rectangle of Length 400 pixels and Width of 300 pixels.

3) Draw a Rectangle of Length 450 pixels and Width of 350 pixels.

Modify This Project to Do the Following Modifications: 30%

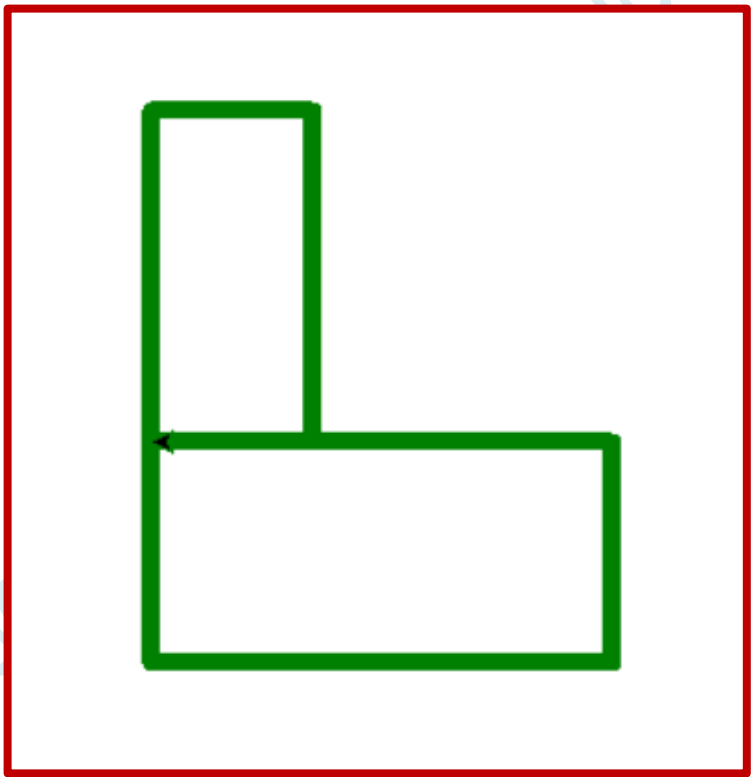
Modify the Turtle Project to Draw another Smaller Rectangle on Top of the previous Rectangle as shown here.



4) **Change the turtle pensize to size 8.**

5) **Change the turtle pencolor to green color.**

6) **Draw another Smaller Rectangle on the top of the previous Rectangle like the following.**



7) **Upload the Last Modified turtle project to Brightspace Assignments Folder.**

Chapter 3 + Quiz 3 + Test Your Python Language Knowledge:

---

1. What is the output after executing the following Python code?

```
x = 10;  y = 20;  z = 30
if x >= 10 and y >= 20 and z > 30:  # with 'and' All conditions must be True
                                    # the first 2 if conditions are True but Z > 30 is False
    print ('True')
else:
    print ('False')
```

A. True B. False C. 10 D. none E. error

---

2. What is the output after executing the following Python code?

```
name1 = name2 = name3 = 'John'
print (Name3)      # notice Name3 and N is in capital
```

A. John B. Name3 C. name1 D. name2 E. NameError

---

3. What is the output after executing the following Python code?

```
a = 0;     b = 1
while a < 1:                    # while 0 < 1: which is True
    while b < 2:                # while 1 < 2: which is True
        print (a, b)            # prints 0 1
        a += 1; b += 1          # increment a by 1, 0+1=1, b=1+1=2
```

A. 1 0 B. 0 1 C. 1 1 D. 1 2 E. error

---

4. What is the output after executing the following Python code?

```
a = False;  b = True
if a or b:      # remember with 'or' only 1 condition may be True
   print ('True')
else:
   print ('False')
```

A. True B. False C. a D. b E. error

---

5. What is the output after executing the following Python code?

```
print ( 13 // ( 2 * 2)  )        # Do first: (2*2) = 4, Then 13 // 4 = 3
```

A. 3 B. 3.5 C. 4 D. 12 E. None of these options

---

6. What is the output after executing the following Python code?

```
if 2 == 2.0:                    # if 2 is == to 2.0, which is True
    print ('same values')       # then print 'same values'
else:
    print ('different values')
```

A. same values B. different values C. 2 D. 2.0 E. None of these options

---

7. What is the output after executing the following Python code?

```
average = 99
if average >= 0 and average <= 100:    #average is >= 0 and average <= 100(True)
    print ('Valid Average')
else:
    print ('Invalid Average')
```

A. Valid Average B. Invalid Average C. 99 D. error E. None of these options

---

8. What is the output after executing the following Python code?

```
average = 122
if average < 0 or average > 100:     # average is < 0 or average > 100 (True)
    print ('Invalid Average')
else:
    print ('Valid Average')
```

A. Valid Average B. Invalid Average C. 122 D. error E. none of these options

---

9. Which of the following is a **Valid Python Variable Name** ?

A. if B. while C. break D. 1number E. if

---

10. Which of the following Python statement will **print the value 7** ?

A. `print(7 % 7)` # remainder **0** # **%** returns **remainder** of **division**

B. `print(7 % 4)` # remainder **3**

C. `print(7 % 5)` # remainder **2**

D. `print(7 % 8)` # remainder **7**

Answers Are Found at The End of This Chapter 03.

Chapter 3 + Python Homework #3 Part B (Due This Sunday) 100 Points

Name:_____ **CIS 103 Python** Programming Language + Wright College

Do the Following Chapter 3B Python Homework: *Instructor: Ogar Haji*

1) What is the output after executing the following Python code?

print (max (22, 11, 44, 77))

A. 22 B. 11 C. 44 D. 77 E. None of these options

2) What is the output after executing the following Python code?

print (min (22, 11, 44, 77))

A. 22 B. 11 C. 44 D. 77 E. None of these options

3) What are the following Symbols used in Python language:

A. # B. **%** C. / D. // E. \

4) Write the Python code to **print** the **Calendar** of the **year 2023**.

5) Write the Python code to **declare** a **variable** called '**year'** and **assign** the year **2023** to it.

6) Write the Python code to **print** the **Calendar** and **pass** the **parameter year** to it.

7) Write the Python code to **prompt** the user to **input** the **year** of the **calendar.**

8) Write the Python code to **Check** for **Invalid Year** which is **less than 2000** and **more than 2500**.

9) What is the output after executing the following Python code?

```
age = 21
print (age > 17 and age < 25)      # both conditions are True
```

A. 21 B. 22 C. True D. False E. None of these options

10) Write the turtle Python code to **Draw** a **Rectangle** in Turtle module.

11) Write the turtle Python code to **change** the **turtle pen size** to **size 5**.

12) Write the turtle Python code to **change** the **turtle pen color** to **blue color**.

13) With Conditional Operator **and , All Conditions** must be **True** so the **true block** will be **executed**; **otherwise**, the **false block** will be **executed**. _____ (True/False)

14) What is the output after executing the following Python code?

```
price = input ('Enter the price of the Book: ')  # Assume user enters 21.88
print ( type (price) )   # user enters '21.88' and it is string
```

A. <class 'int'> B. <class 'double'> C. <class 'str'> D. <class 'double'> E. error

15) What is the output after executing the following Python code?

```
number = input ('Enter number of students: ')  # Assume user enters  7
print ( type (number) )        # user enters '7' and it is string
```

A.<class 'int'> B.<class 'double'> C. <class 'str'> D.<class 'double'> E.error

16) To Convert a String Number entered by user to float, use function: .

A. int () B. str () C. float () D. bool() E. None of these options

17) What is the output after executing the following Python code?

```
i = 1
while i < 10:          # while 1 < 10: which is True
    i += 3             # increment i by 3, i = 1 + 3 = 4
    print (i)          # prints value of i which is 4, then repeats while for 7, 10
```

A. 4 7 10 B. 1 4 7 C. 2 6 10 D. Syntax Error

18) What is the output after executing the following Python code?

```
print (7 * 7 / 7 * 7)     # * and / have same precedence. From left to right
                          # 7 * 7 = 49,  49 / 7 = 7.0, 7.0 * 7 = 49.0
```

A. 0 B. 1 C. 1.0 D. 49 E. 49.0 F. error

19) With Conditional Operator **or , only 1 Condition** may be **True** so the **true block** will be **executed**; **otherwise**, the **false block** will be **executed**. _____ (True/False)

20) With **Short-Circuit 'and' operator**, if **1 Condition** is **False**, then the **result** is **False** and **Python compiler** will **Not Check** the **rest of conditions**. _____(True/False)

21) With **Short-Circuit 'or' Operator**, if **1 Condition** is **True**, then the **result** is **True** and the **Python compiler** will **Not Check** the **rest of conditions**. _____ (True/False)

22) What is the output after executing the following Python code?

```
n = int (input ('Enter an integer: ') )      # suppose user enters 3
str = input ('Enter a string: ')             # suppose user enters 'Car'
print ( n * str )
```

A."CarCarCar" B. Car Car Car C. CarCarCar D. 3 E. None of these options

23) What is the output after executing the following Python code?

```
name1 = name2 = name3 = 'Mary'
print (Name3)      # notice Name3 and N is in capital
```

A. Mary B. Name3 C. name1 D. name2 E. NameError

24) What is the output after executing the following Python code?

```
a = False
b = True
if a or b:      # remember with 'or' only 1 condition may be True
    print ('True')
else:
    print ('False')
```

A. True B. False C. a D. b E. error

25) What is the output after executing the following Python code?

```
x = 10;  y = 20;  z = 30
if x >= 10 and y >= 20 and z > 30:  # with 'and' All conditions must be True
                                    # the first 2 if conditions are True but Z > 30 is False
    print ('True')
else:
    print ('False')
```

A. True B. False C. 10 D. none E. error

26) What is the output after executing the following Python code?

```
print ( 13 // ( 2 * 2)  )          # Do first: (2*2) = 4, Then 13 // 4
```

A. 3 B. 3.5 C. 4 D. 12 E. None of these options

27) What is the output after executing the following Python code?

```
if 2 == 2.0:                    # if 2 is == to 2.0, which is True
    print ('same values')
else:
    print ('different values')
```

A. same values B. different values C. 2 D. 2.0 E. None of these options

28) What is the output after executing the following Python code?

```
average = 99
if average >= 0 and average <= 100:  #average is >= 0 and average <= 100 (True)
    print ('Valid Average')
else:
    print ('Invalid Average')
```

A. Valid Average B. Invalid Average C. 99 D. error E. None of these options

29) What is the output after executing the following Python code?

```
average = 122
if average < 0 or average > 100:      #  average is < 0 or average > 100 (True)
   print ('Invalid Average')
else:
print ('Valid Average')
```

A. Valid Average B. Invalid Average C. 122 D. error E. none of these options

30) Which of the following is a **Valid Python Variable Name** ?

A. if B. while C. break D. 1number E. _if

31) What are **the results** of the following **print()** Python function ?

A. **print (7 % 7)** # remainder # % returns remainder of division
B. **print (7 % 4)** # remainder
C. **print (7 % 5)** # remainder
D. **print (7 % 8)** # remainder

32) What is the output after executing the following Python code?

```
name = input ('Enter your name: ')          # suppose user enters 'Mary'
age = int ( input ('Enter your age: ') )    # suppose user enters '21'
print (name, type (age) )
```

A. Mary 21 int B. Mary <class 'int'> C. Mary 21 D. Mary <class 'str'> E. error

33) What is the output after executing the following Python code?

```
name = ""               # name is assigned Empty String
while name:             # name is False, so else is executed
    print ("Good Morning")
else:
    print ("Good Night")
```

A. error B. Good Morning C. Good Night D. Good name="" E. nothing

Chapter 3 + Python Lab Assignment #3A (Due This Sunday) 100 Points

Name:_____ CIS 103 Python language + Wright College

```
********** American Cars Rental *********
*****
******** Calculate Total Amount Due Program
Enter Customer Full Name: Mary Smith
Enter Number of Days Rented: 5
Enter Ending Miles: 800
Enter Beginning Miles: 700
****************************************
```

Python Console Assignment #3A: American Cars Rental Project:

+(Do Lab Assignment + Lab 3A)+-

Do Lab Assignment + Lab 3A

Problem or Project: Design and Code in Python Language the project to Calculate the Charges for American Cars Rental Company.

Do the 12 Must Steps to Design, Code & Solve a Project in Python Language

Do Steps 1 thru 7 in your Note Book or on Paper.

Note: When Modifying a Project, Do One Modification at a time.

1) Save Console project as **AmericanCarsRentalLab3A**

2) Note: Company Charges $40 Dollars per **Day** and **0.30 Cents** per **Mile**.

3) The Calculations are as follows:

Charges By Days = Days Rented * 40

Charges By Miles = (Ending Miles – Beginning Miles) * 0.30

Total Charges = Charges By Days + Charges By Miles

Discount = Total Charges * 0.20

Amount Due = Total Charges – Discount

4) Run the project, and display Amount in Currency Formatting $.

5) Type the following Data: Your Name 5 800 700

6) Check for Invalid Input Data:

7) Check for Invalid Days: Less than Zero and Greater than 100

8) Check for Invalid starting Miles: Less than Zero

9) Check for Invalid Ending Miles: Less than Zero or less than Starting Miles

The Input of the program will look like the following:

********** American Cars Rental **********
***** ********
********* Calculate Total Amount Due Program *********
Enter Customer Full Name: Mary Smith
Enter Number of Days Rented: 5
Enter Ending Miles: 800
Enter Beginning Miles: 700
****

The calculated Output of the **project looks like the following:**

Charges By Days = $200.00
Charges By Miles = $30.00
Total Charges = $230.00
Discount (20%) = $46.00
Total Amount Due = $184.00

****
********** End of Program **********
****** Programmer: Instructor: Ogar Haji *****

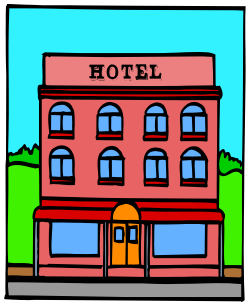
Modify This Project to Do the Following Modifications: 30%

1) Modify the Project, **Add** a **Named Constant SALES_TAX** at **0.08** and **Calculate Sales Tax** and also **Total Amount Due**.
2) Before printing **Footers, Declare 2 integer variables** called **'number1'** and **'number2'** and **initialize number1** to **55** and **number2** to **33**.
3) Use max() and min() functions to **find** the **Maximum number** and the **Minimum number** of the **2 numbers** and print them out.
maxNumber = max(number1, number2)

Note: When Modifying a Project, Do One Modification at a time.

Chapter 3 + Python Lab Assignment #3B (Due This Sunday) 100 Points

Name:_____ CIS 103 Python language + Wright College



Python Programming Assignment #3B: Hotel Room Charges Calculator Project

+++(Lab Assignment + Lab 3B)+++

Lab Assignment Lab 3 B

Do the 12 Must Steps to Design, Code and Solve a Project in Python Language.

Do Steps 1 thru 7 in your Note Book or on Paper.

Note: When Modifying a Program, Do One Modification at a time.

1) Design and **Code** a **Console Python App and Save as**

HotelRoomChargesLab3B.

Do Lab Assignment 3B

2) Prompt the User to Enter Number of Nights stayed at the

Hotel and

3) Prompt the user to Enter the Nightly Charges

4) Then continue the Rest of Prompt and Calculations.

Purpose of the Project:

- **Calculate Room Charges based on a per night Rate**

- **Room Service Charges**
- **Telephone Charges**
- **Miscellaneous Charges**
- **Also Calculate Taxes then**
- **Calculate Total Charges**
- **Give a Discount of 10% if Number of Nights is over 7 Nights**
- **Give a Gift Certificate of $50 and if Total Amount Charges is over $500.**
- **Insert Computer Current Date and Time at the Top.**

Note: Always **Upload** to **Brightspace** the **Modified Python Project**
1) The **Microsoft Word Document** of the **Python Project Code along with the Python Output Screen shots.**

Chapter 3 + Quiz 1 + Answers to Python Language Knowledge:

---

1. What is the output after executing the following Python code?

```
i = 0
while i < 1:                    # while 0 is < 1 is True
    print ('Hello', end=", ")    # print 'Hello, '
    i += 1                      # increment i by 1, so i = 1 and while is False
else:
    print ("World")             # this print 'World'
```

A. Hello B. "Hello", World C. Hello World **D. Hello, World** E. none

---

2. What is the output after executing the following Python code?

```
no_milk_left = "None"

if no_milk_left:            # This if statement is True
    print ("Buy Milk")
else:
    print ("You have Milk")
```

A. You have Milk **B. Buy Milk** C. error D. None

---

3. What is the output after executing the following Python code?

```
age = 21
print(age > 20 and age < 22)  # both conditions are True, so it prints True
```

A. 21 B. 22 **C. True** D. False E. None of these options

---

4. What is the output after executing the following Python code?

```
n = int (input ('Enter an integer: ') )      # suppose user enters 3
str = input ('Enter a string: ')             # suppose user enters 'Pie'
print ( n * str )
```

A. "PiePiePie" B. Pie Pie Pie **C. PiePiePie** D. 3 E. None of these options

---

---

5. What is the output after executing the following Python code?

```
name = input ('Enter your name: ')          # suppose user enters 'Mary'
age = int ( input ('Enter your age: ') )    # suppose user enters '21'
print (name, type (age) )
```

A. Mary 21 int **B. Mary <class 'int'>** C. Mary 21 D. Mary <class 'str'> E. error

---

6. What is the output after executing the following Python code?

```
name = ""           # name is assigned Empty String
while name:         # name is False, so else is executed
    print("Good Morning")
else:
    print("Good Night")
```

A. error B. Good Morning **C. Good Night** D. Good name="" E. nothing

---

7. What is the output after executing the following Python code?

```
average = 77
if average >= 90 and average <= 100:
    print("Grade 'A'")
elif average >= 80 and average <= 89:
    print("Grade 'B'")
elif average >=70 and average <= 79:  # with and both conditions must be True
    print("Grade 'C'")
elif average >=50 and average <= 69:
    print("Grade 'D'")
else:
    print("Grade 'F'")
```

A. Grade 'A' B. Grade 'B' **C. Grade 'C'** D. Grade 'D' E. Grade 'F'

---

8. What is the output after executing the following Python code?

```
k = (4 / 2) * 3 + 2   # do (4/2) first. 4/2 = 2.0 then 2.0 * 3 = 6.0, 6.0 + 2 = 8.0
print (k)
```

A. **8.0** B. 8 C. 1.2 D. 10 E. none of these options

Answers:

Chapter 3 + Quiz 2 + Answers to Python Language Knowledge:

---

1. What is the output after executing the following Python code?

```
number = 22
if number or True:      # number has a value 22 which means True
    number = 33         # assign 33 to number and replace old number
print (number)
```

A. 22 B. 55 **C. 33** D. none E. error

---

2. What is the output after executing the following Python code?

```
price = input ('Enter the price of the Book: ')  # Assume user enters 21.88
print ( type (price) )      # user enters '21.88' and it is string
```

A. <class 'int'> B. <class 'double'> **C. <class 'str'>** D. <class 'double'> E. error

---

3. What is the output after executing the following Python code?

```
number = input ('Enter number of students: ')  # Assume user enters  7
print ( type (number) )      # user enters '7' and it is string
```

A. <class 'int'> B. <class 'double'> **C. <class 'str'>** D. <class 'double'> E. error

---

4. To Convert a String Number entered by user to float, use function: .

A. int () B. str () **C. float ()** D. bool() E. None of these options

---

5. What is the output after executing the following Python code?

```
print (max ( 22, 11, 44, 77 ) )
```

A. 22 B. 11 C. 44 **D. 77** E. None of these options

---

6. What is the output after executing the following Python code?

print (min (22, 11, 44, 77))

A. 22 **B. 11** C. 44 D. 77 E. None of these options

---

7. Match the following Symbols used in Python language:

A. **#** B. **%** C. **/** D. **//** E. ****

A. Line comment **B. Modulus operator** **C. Integer division**

D. Float Division **E. Line continuation**

---

8. What is the output after executing the following Python code?

```
average = 97
if average >= 70:                # if 97 is >= 70 which is True
    print ('Passing')            # then prints 'Passing'
else:
    print ('Failing')
```

A. nothing **B. Passing** C. Failing D. 97 E. None of these options

---

9. What is the output after executing the following Python code?

```
i = 1
while i < 10:          # while 1 < 10: which is True
i += 3                 # increment i by 3, i = 1 + 3 = 4
print (i)              # prints value of i which is 4, then repeats while for 7, 10
```

A. 4 7 10 B. 1 4 7 C. 2 6 10 D. Syntax Error

---

10. What is the output after executing the following Python code?

```
print (7 * 7 / 7 * 7)      # * and / have same precedence. From left to right
                           # 7 * 7 = 49,  49 / 7 = 7.0, 7.0 * 7 = 49.0
```

A. 0 B. 1 C. 1.0 D. 49 **E. 49.0** F. error

Answers

Chapter 3 + Quiz 2 + Answers to Python Language Knowledge:

1. What is the output after executing the following Python code?

```
x = 10;  y = 20;  z = 30
if x >= 10 and y >= 20 and z > 30:  # with 'and' All conditions must be True
                                    # the first 2 if conditions are True but Z > 30 is False
    print ('True')
else:
    print ('False')
```

A. True **B. False** C. 10 D. none E. error

---

2. What is the output after executing the following Python code?

```
name1 = name2 = name3 = 'John'
print (Name3)      # notice Name3 and N is in capital
```

A. John B. Name3 C. name1 D. name2 **E. NameError**

---

3. What is the output after executing the following Python code?

```
a = 0;      b = 1
while a < 1:                    # while 0 < 1: which is True
      while b < 2:              # while 1 < 2: which is True
            print (a, b)        # prints 0 1
            a += 1; b += 1      # increment a, b by 1, 0 + 1 =1, b = 1 + 1 = 2
```

A. 1 0 **B. 0 1** C. 1 1 D. 1 2 E. error

---

4. What is the output after executing the following Python code?

```
a = False;  b = True
if a or b:      # remember with 'or' only 1 condition may be True
   print ('True')
else:
   print ('False')
```

A. True B. False C. a D. b E. error

---

5. What is the output after executing the following Python code?

```
print ( 13 // ( 2 * 2)  )       # Do first: (2*2) = 4, Then 13 // 4 = 3
```

A. 3 B. 3.5 C. 4 D. 12 E. None of these options

---

6. What is the output after executing the following Python code?

```
if 2 == 2.0:                    # if 2 is == to 2.0, which is True
    print ('same values')       # then print 'same values'
else:
    print ('different values')
```

A. same values B. different values C. 2 D. 2.0 E. None of these options

---

7. What is the output after executing the following Python code?

```
average = 99
if average >= 0 and average <= 100:
    print ('Valid Average')
else:
    print ('Invalid Average')
```

A. Valid Average B. Invalid Average C. 99 D. error E. None of these options

---

8. What is the output after executing the following Python code?

```
average = 122
if average < 0 or average > 100:
    print ('Invalid Average')
else:
    print ('Valid Average')
```

A. Valid Average **B. Invalid Average** C. 122 D. error E. none of these options

---

9. Which of the following is a **Valid Python Variable Name** ?

A. if B. while C. break D. 1number **E. _if**

---

10. Which of the following Python statement will **print the value 7** ?

A. print(7 % 7) # remainder **0** **# % returns remainder of division**

B. print(7 % 4) # remainder **3**

C. print(7 % 5) # remainder **2**

D. print(7 % 8) **# remainder 7** returns the less number **7** than % by **8**

Answers:

Wright College + Chapter 3

Conditional Statements: Using if...else and Logical Operators: ('and', 'or' & 'not') and Python Turtle Module

CIS 103 Python Programming Language

Introduction to Computer Programming

'Hands-On' Mastering Computer Logic, Design and Programming Using Python Language

Written By:

OGAR HAJI

Master's Degree in Computer Science
DePaul University ✚ Chicago, Illinois, USA

Date Published: January 29, 2023