CB2240: Introduction to Business Programming in Python

Wang Le, Albert Chung, David Xu

Instruction team

Instructors

- Wang Le (Wed morning)
  - Le.Wang@cityu.edu.hk
- Albert Chung (Thu morning)
  - ykchung@cityu.edu.hk
- David Xu (Thu noon)
  - davidxu@cityu.edu.hk

Teaching Assistants

Qin Yaxue:
yaxueqin2-c@my.cityu.edu.hk
Song Bingqing:
bingqing.song@my.cityu.edu.hk
Wang Tianteng (Tom):
Tom.Wang@my.cityu.edu.hk
Zheng Ruoshu (Judy):
ruoszheng3-c@my.cityu.edu.hk

Office Hours:
Monday 5:00-7:00 pm (Tom)
Thursday 3:00-5:00 pm (Song)
Friday 2:00-4:00 pm (Judy)

Email mail them in advance so they can better coordinate the meetings with students.

Key topics of CB2240

- Basics:
  - Numbers, strings, list, tuples.
  - if statements, for and while loops, functions, etc.

- Applications
  - Data handling, data visualization, etc.

- Textbook: David I. Schneider, An Introduction to Programming Using Python, 1st edition, Pearson, 2016.

Our goals

- Understand the basics of Python.
- Learn some business applications of Python.
- Learn how to think like a programmer.

Course overview

- Python is a popular open source programming language widely used by business professionals to develop applications in big data, artificial intelligence, financial technology, and so on.
- This course provides an introduction to the Python language and helps students master its fundamentals and apply them in different contexts to solve business problems.

Course overview

- It is easy to learn and fun to use Python.
- Students without programming background are perfectly fine.
- Students' skill set upon course completion should include implementing decision-making structure, repetition structure, function procedures, data handling and visualization, testing, among others.

Course topics (tentative)

	Module	Topic	Reading
1	Introduction	Overview	Chapter 1
2	Core objects	Numbers	2.1
3		Strings	2.2
4		Lists and tuples	2.4
5	Conditionals and loops	Conditions and if statements I	3.1, 3.2
6		Conditions and if statements II	3.1, 3.2
7		while loops	3.3
8		for loops	3.4
9	Functions and program design	Functions I	4.1
10		Functions II, recursion	4.2, 4.3, 6.4
11		Object-oriented programming	Chapter 7
12	Data handling	Data processing	5.1, 5.2
13		Data visualization	

Weekly routine

- Lecture (1 hour)
  - Quiz 5 minutes.
- Lab (2 hours)
  - Python examples
  - Classwork
- Homework

Assessment (mostly graded by TAs)

- Participation: 7% (e.g., help answer questions in class or on Canvas discussion board)
- Class Exercises: 13% (1% per week)
- Weekly Homework: 20%
- Weekly Quizzes: 20%
- Final Exam: 40%

Weekly classwork & homework

- Deadlines for classwork
    - Within 1 hour after the class section
- Deadlines for homework
    - Sunday evening for the Wed section
    - Monday evening for the Thu sections
- Classwork and Homework Submission format
    - Name the file with week number and your student ID, e.g.: 51234567week1.py
    - Include your name in both the code and screenshot (e.g., jpg file) showing the code output. Do not zip the files.
- Late submission will result in point deductions.
    - For each hour after the deadline, the submission will lose 10% of the maximum possible score.

Communication policy

- If you have questions about class logistics, feel free to email the instructors and TAs.
- Python questions should be posted on Canvas' discussions so all students can learn.

- What the instructor & TAs won't do: debug programs for you.

A note on academic (dis)honesty

- As a general principle, all work submitted must be **your own** work
- Focus:
  - your **understanding** of the concepts learned
  - your **ability to apply** them
- Copying and pasting does not demonstrate understanding or ability to apply
- Cheating, plagiarism, and collusion are serious offenses resulting in an F grade and disciplinary action
- Code of Student Conduct:
  - http://www.cityu.edu.hk/provost/academic_honesty/index.htm

Some ground rules

- Actively participate
- Be prepared
- Be punctual & don’t miss the quiz
- Mute your mic during the online class
- Be professional
    - Emails include “CB2240” in subject line
    - Include greeting and signature
    - Include session number

Quiz 1

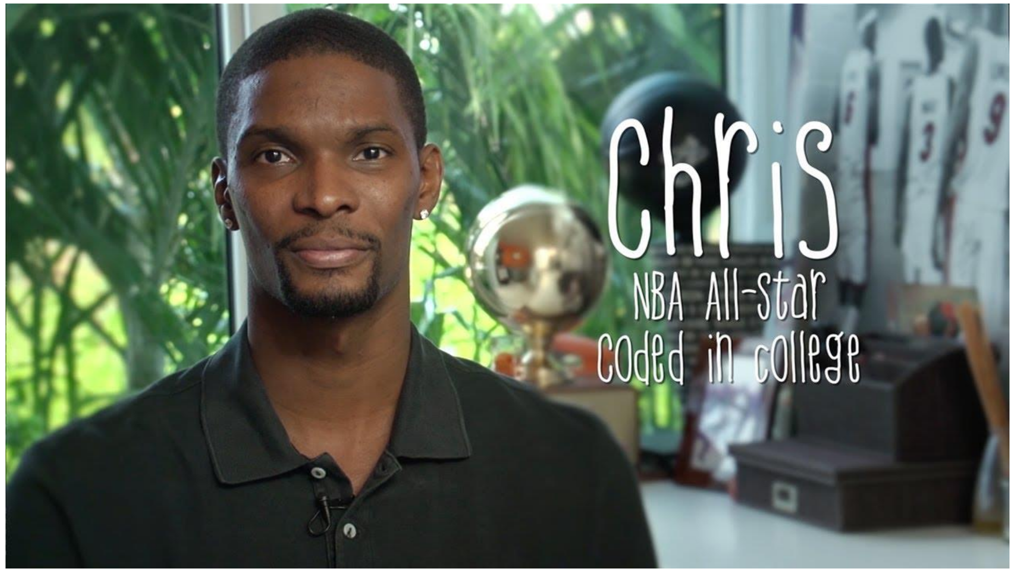
Lecture 1.
An Introduction to Computing and Problem Solving

Why programming?

- Programming allows us to
  - Communicate with computers and machines.
  - Harness computing powers; automate tasks.
  - Tap into machine learning (ML) and artificial intelligence (AI).

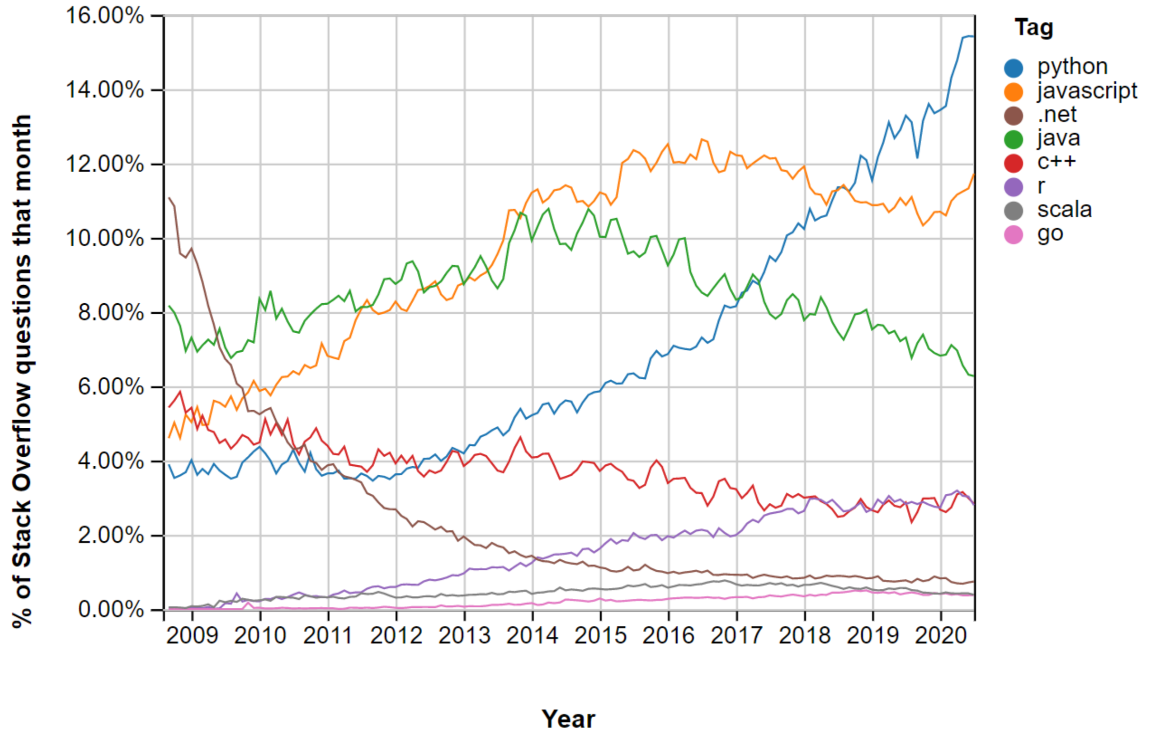
Why programming?

- Today, almost every profession needs programming to some extent.
  - https://www.youtube.com/watch?v=Dv7gLpW91DM



Why Python?

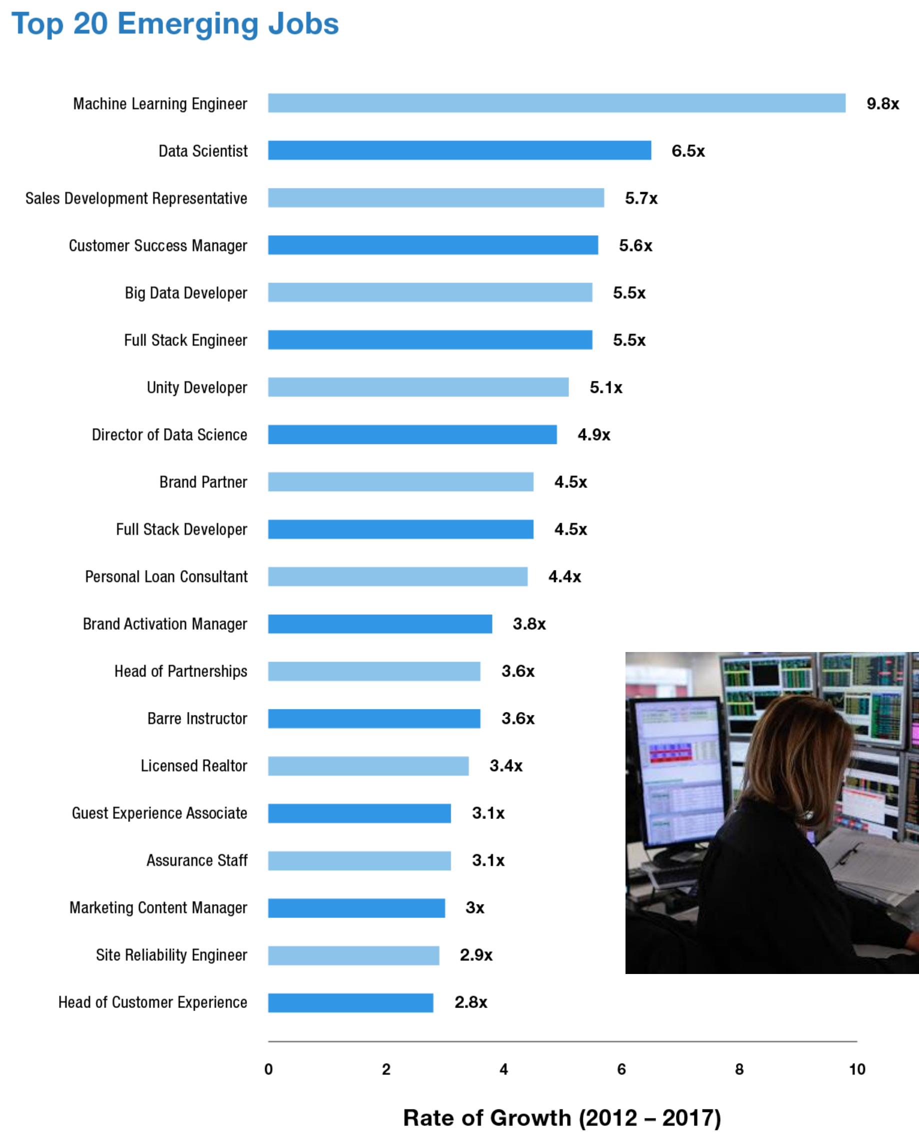
- Large and still growing Python community

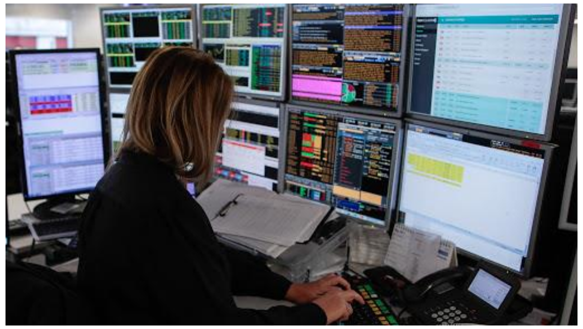


Why Python?

- Powerful, multi-purpose
  - Accounting
  - Finance (e.g. quant, trader)
  - Digital marketing
  - Supply chain analytics
  - Academic research
  - Etc.

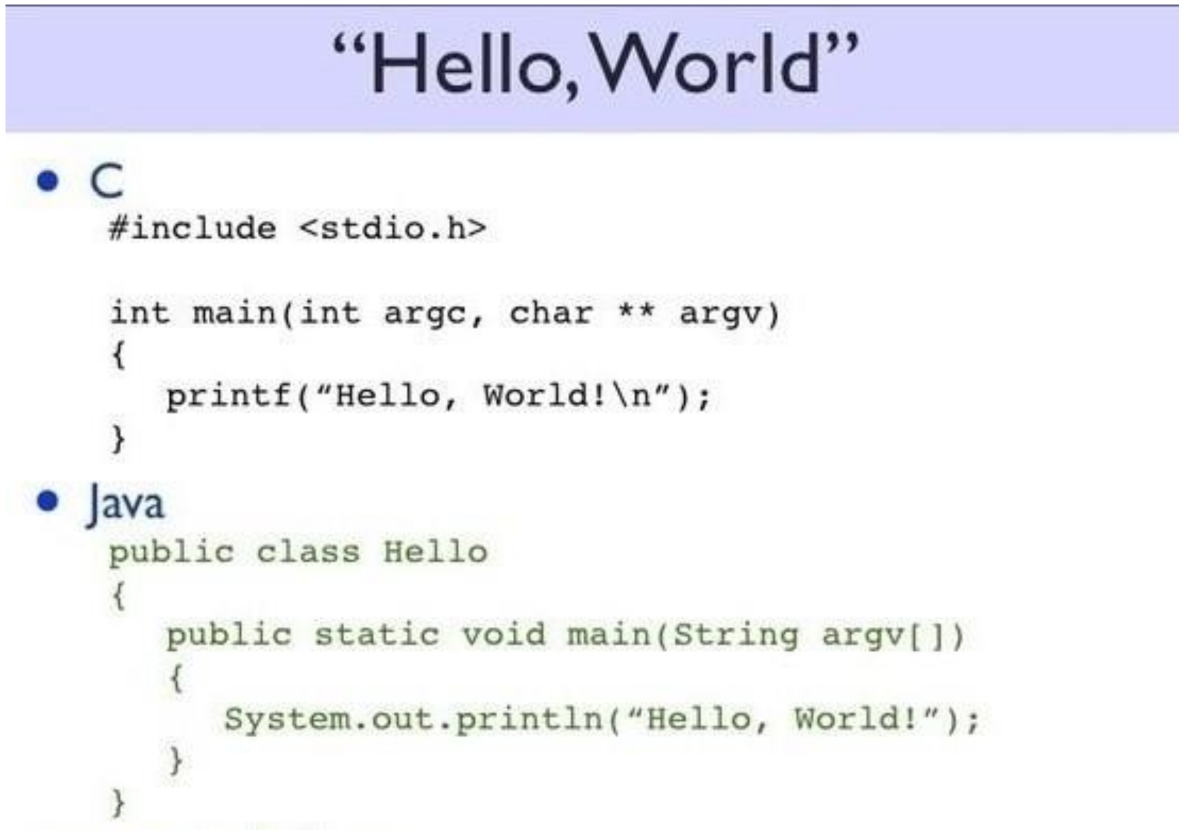
- Huge demand in the job market





Why Python?

- Beginner-friendly
  - Plain English
  - Vast open-source libraries



"Hello, World"

- C

```c
#include <stdio.h>

int main(int argc, char ** argv)
{
   printf("Hello, World!\n");
}
```

- Java

```java
public class Hello
{
   public static void main(String argv[])
   {
      System.out.println("Hello, World!");
   }
}
```

- **Python**

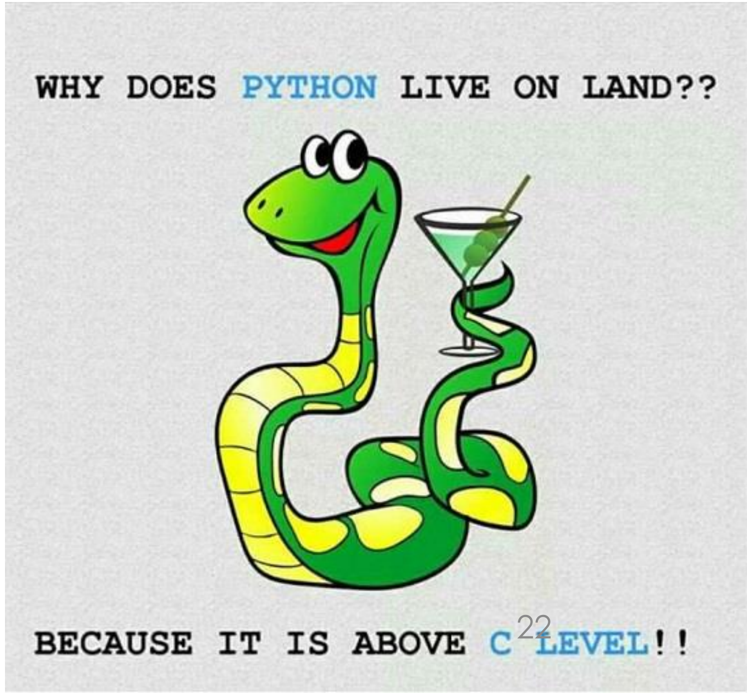
```python
>>> print("Hello, World!")
```

Real world application example

- Walmart and Amazon scrape each other's product data.

About Python

- Python is named after the British comedy group Monty Python.
- Python is a high-level programming language.

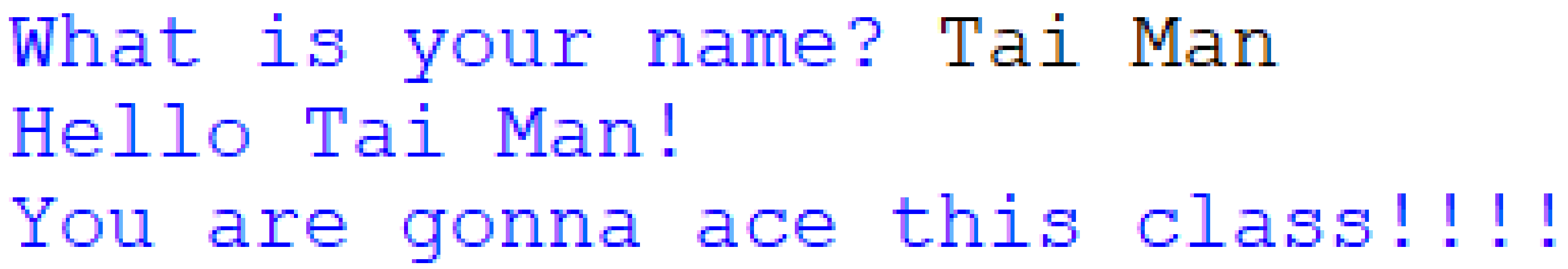


Questions and Answers

- How did the language Python get its name?
  - Named for the British comedy group Monty Python (really!)
- Book uses the editor IDLE to create programs. How did IDLE get its name?
  - Stands for **I**ntegrated **D**eve**L**opment **E**nvironment
- How are problems solved with a program?
  - Step-by-step procedure devised to process given data and produce requested output.

A simple program

- What is an example of a program?



```
What is your name? Tai Man
Hello Tai Man!
You are gonna ace this class!!!!
```

Problem Solving

- Examine a small problem
- Please identify the largest number from the following list:

18	23	53	08

- Most of you know what the answer is, but how did you arrive at this answer?

Problem Solving

- (Step 1) Write down the first number on a piece of paper (largest so far and is the largest if it is the only number)
- (Step 2) Repeat the following steps until no more numbers:
  - (a) Take the next number from the list
  - (b) Compare the two numbers, if the new number is bigger it replace the old number on the paper
  - (c) Return to the first step
- After step 2, the last number of the paper is the answer.

Questions and Answers

- Where will new programs be saved?
  - Create a special folder to hold your programs
- Where can I research questions I have about Python?
  - Documentation at https://www.python.org/doc/
  - W3schools https://www.w3schools.com/python/

Program Planning and Tools

Program Planning

1. ***Analyze:*** Define the problem.
2. ***Design:*** Plan the solution to the problem.
3. ***Code:*** Translate the algorithm into a programming language.
4. ***Test and correct:*** Locate and remove any errors in the program.
5. ***Complete the documentation:*** Organize all the material that describes the program.

Think before you code

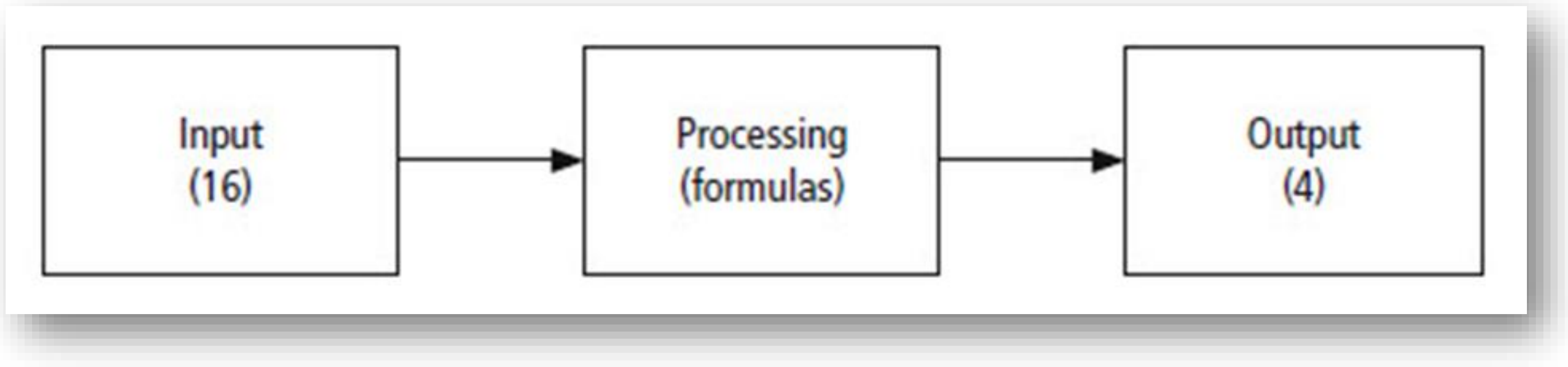
- Flowcharts
- Pseudocode
- Hierarchy charts

Algorithm Development

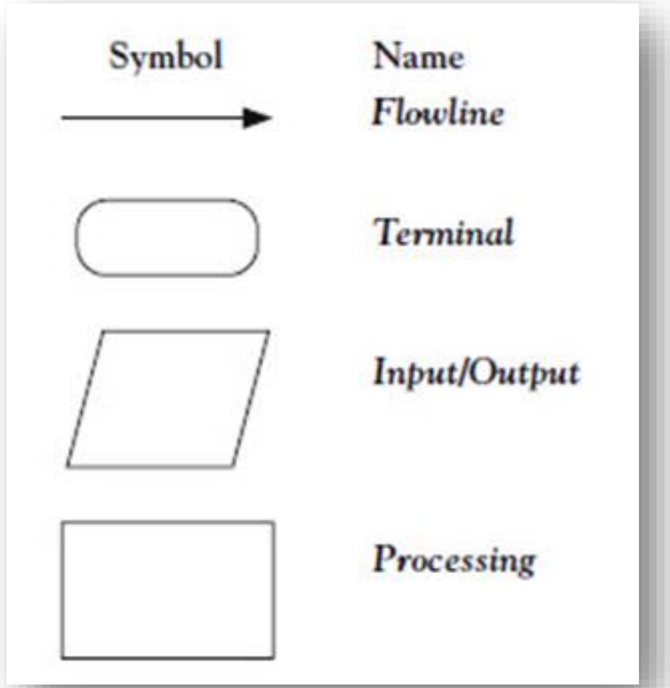
Algorithm to determine the number of stamps for a letter

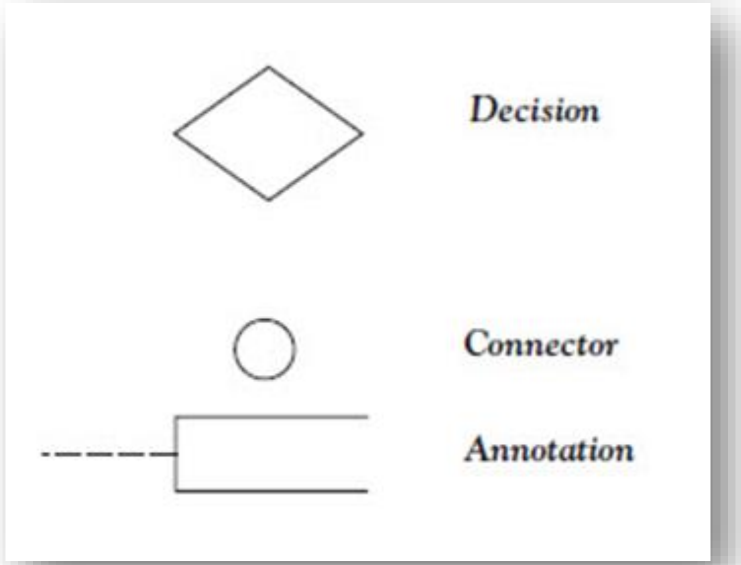
- Rule of thumb: 1 stamp for every 5 sheets of paper
  1. Request sheets of paper
  2. Divide by 5
  3. Round quotient up to next whole number
  4. Reply with the number of stamps

Problem Solving for Stamps

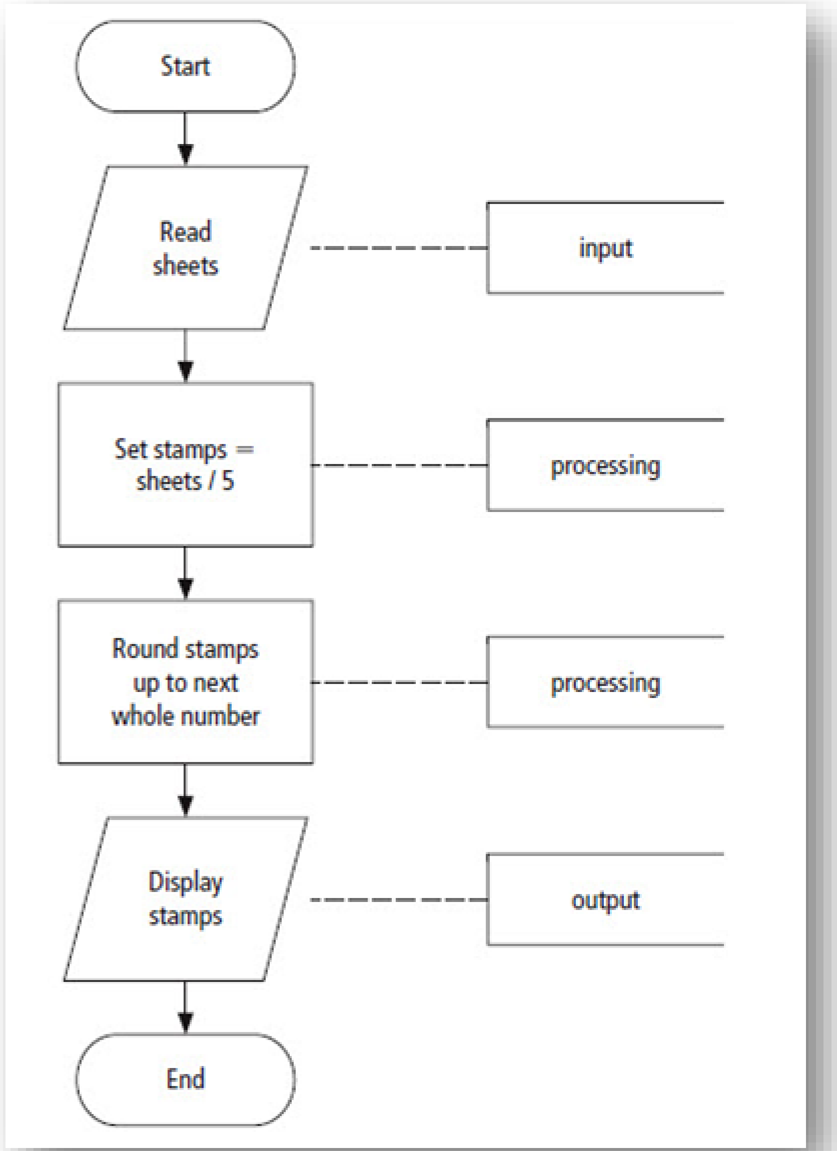


Flowcharts


Symbol
Name
Flowline
Terminal
Input/Output
Processing


Decision
Connector
Annotation

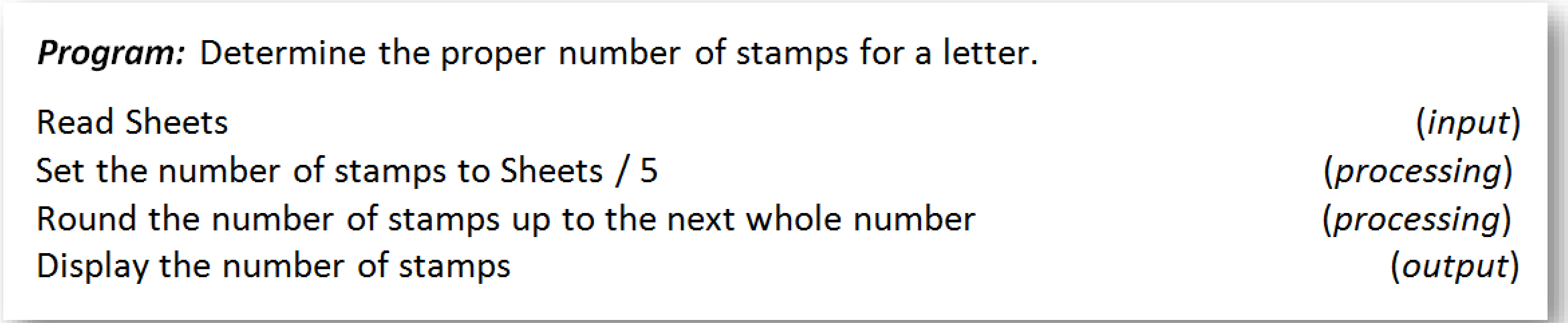
Example Flowchart



Pseudocode

- Abbreviated plain English version of actual computer code
- Symbols used in flowcharts replaced by English-like statements
- Allows programmer to focus on steps required to solve problem

Example Pseudocode



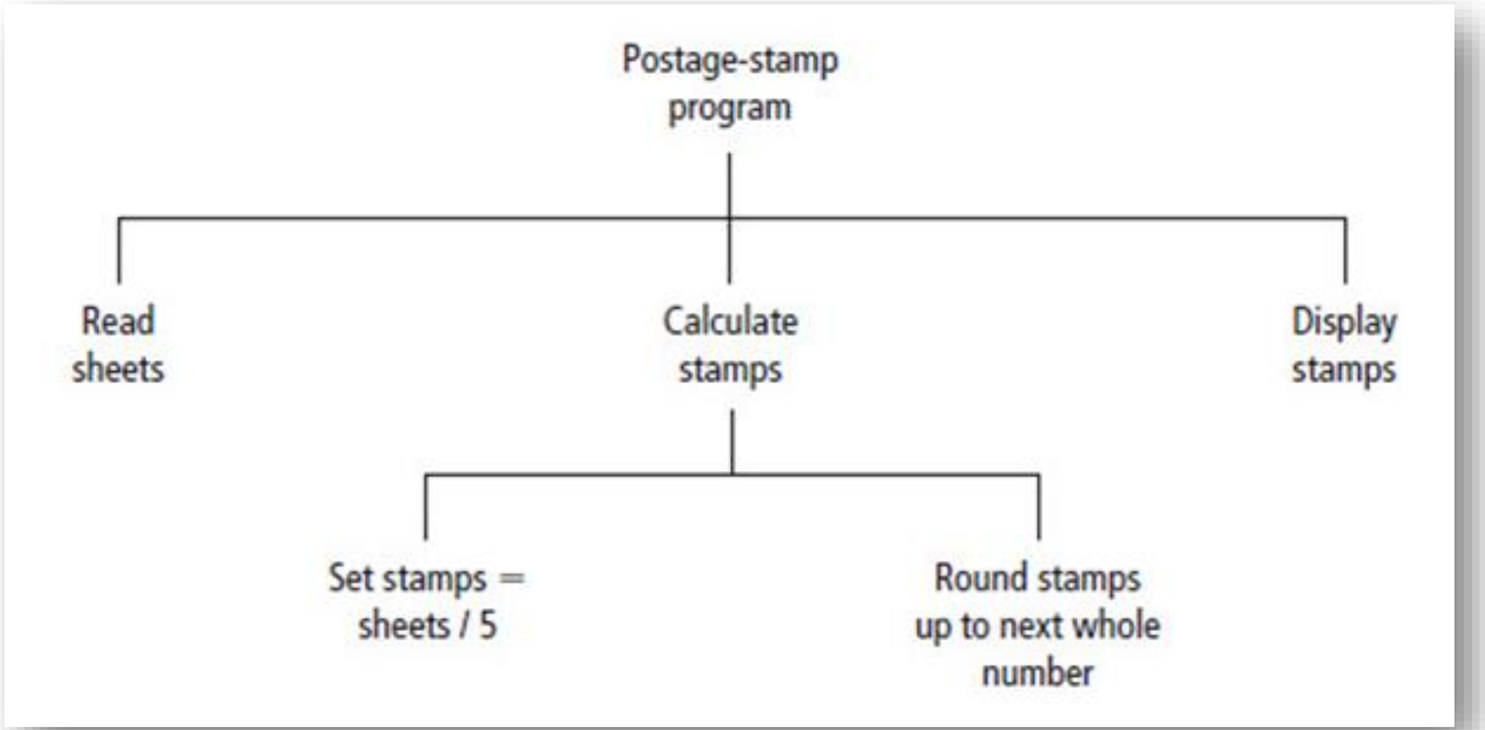
Program: Determine the proper number of stamps for a letter.

```
Read Sheets                                                  (input)
Set the number of stamps to Sheets / 5                       (processing)
Round the number of stamps up to the next whole number       (processing)
Display the number of stamps                                 (output)
```

Hierarchy Chart

- Shows the overall program structure
- Depict organization of program, omit specific processing logic
- Describe what each part, or module, of the program does
- Each module subdivided into a succession of submodules

Example Hierarchy Chart

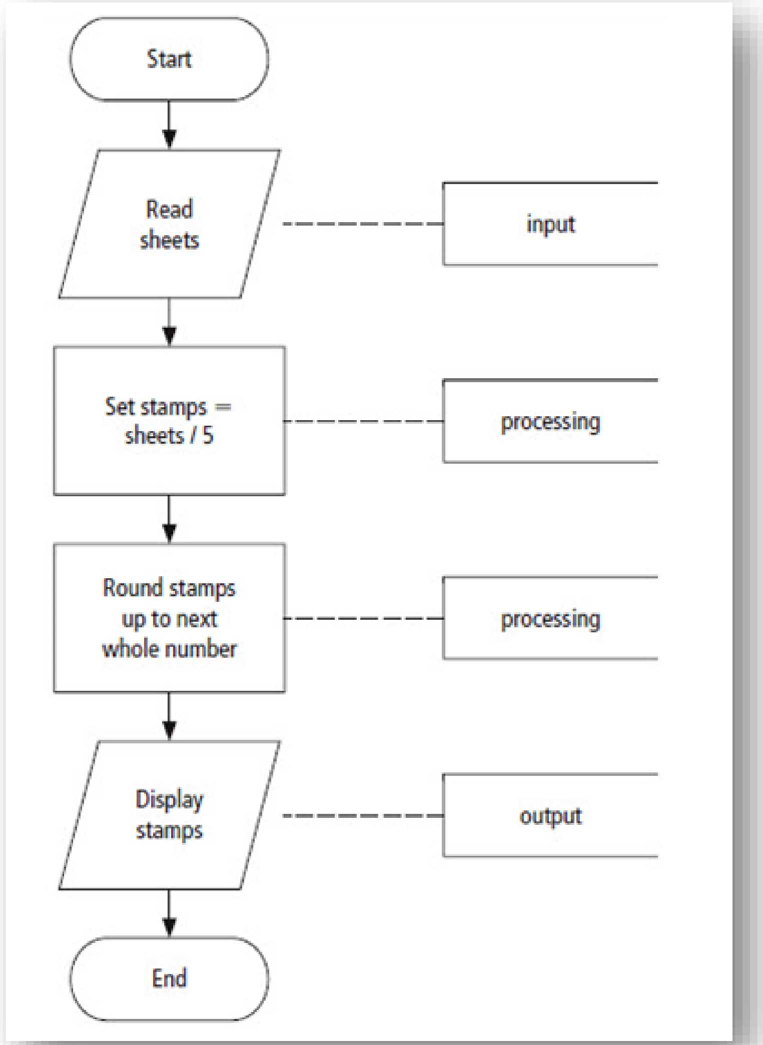


Three structures

- Sequence structure
- Decision structure
- Repetition/loop structure

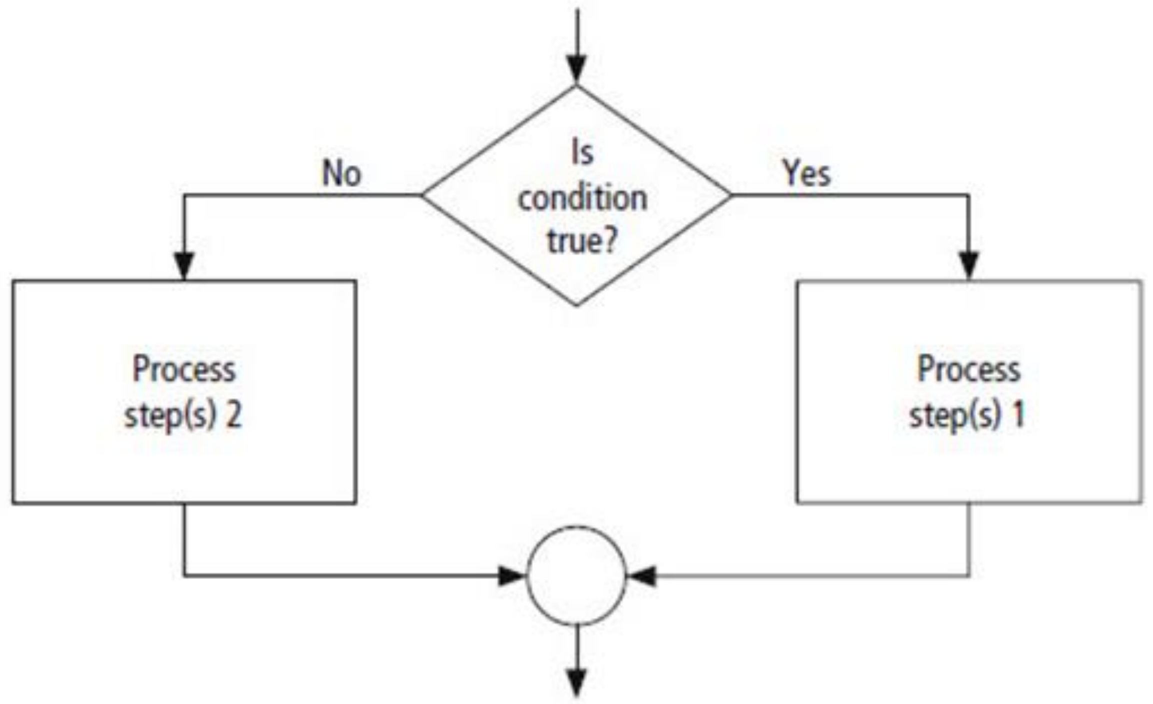
Sequence structure

- The postage-stamp problem has a **sequence structure**:
  - Moved from one line to the next without skipping any lines.



Decision Structure

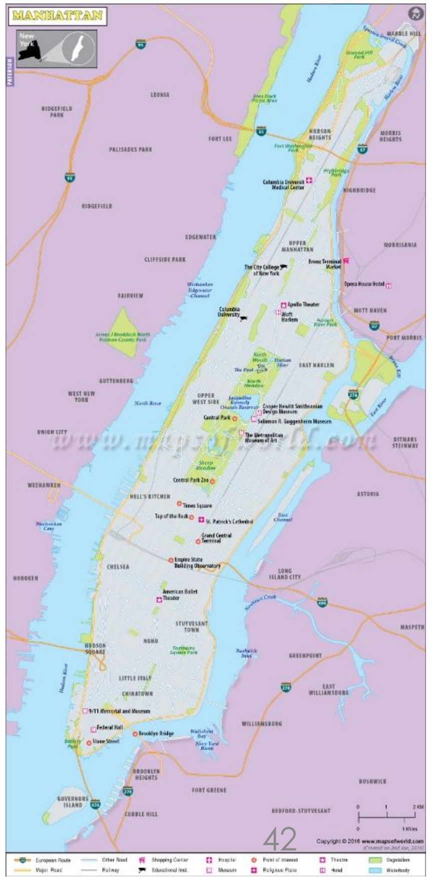
```
if condition is true
   Process step(s) 1
else
   Process step(s) 2
```



Direction of Numbered NYC Streets Algorithm

- ***Problem:*** Given the street number of a one-way street in New York City, decide the direction of street - eastbound or westbound.

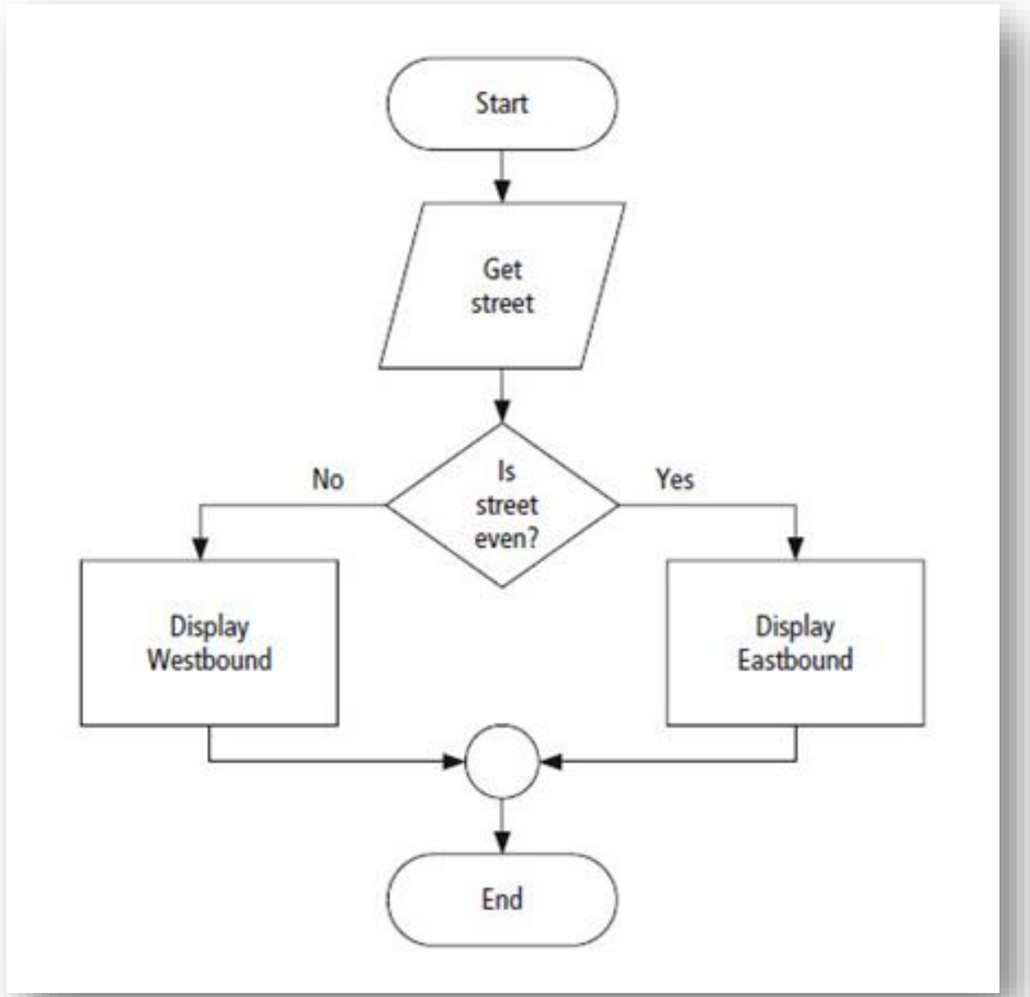
- In NYC
  - Even-numbered streets: eastbound
  - Odd-numbered streets: westbound



Direction of Numbered NYC Streets Algorithm

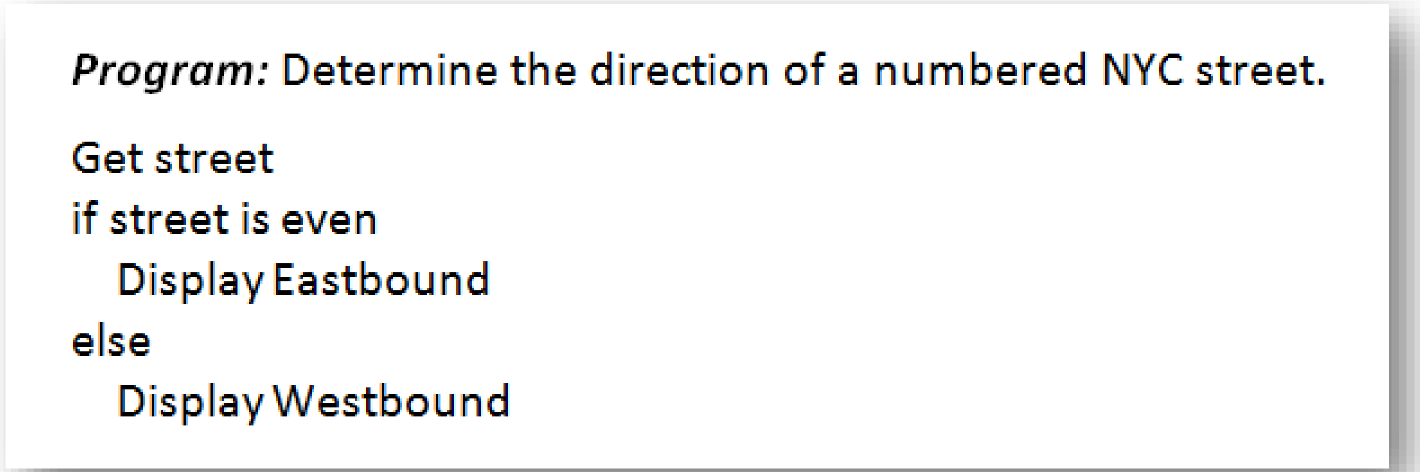
- ***Input:*** Street number.
- ***Processing:*** Decide if the street number is divisible by 2.
- ***Output:*** “Eastbound” or “Westbound”.

Street Direction Algorithm



Flowchart for the numbered New York City streets problem.

Street Direction Algorithm

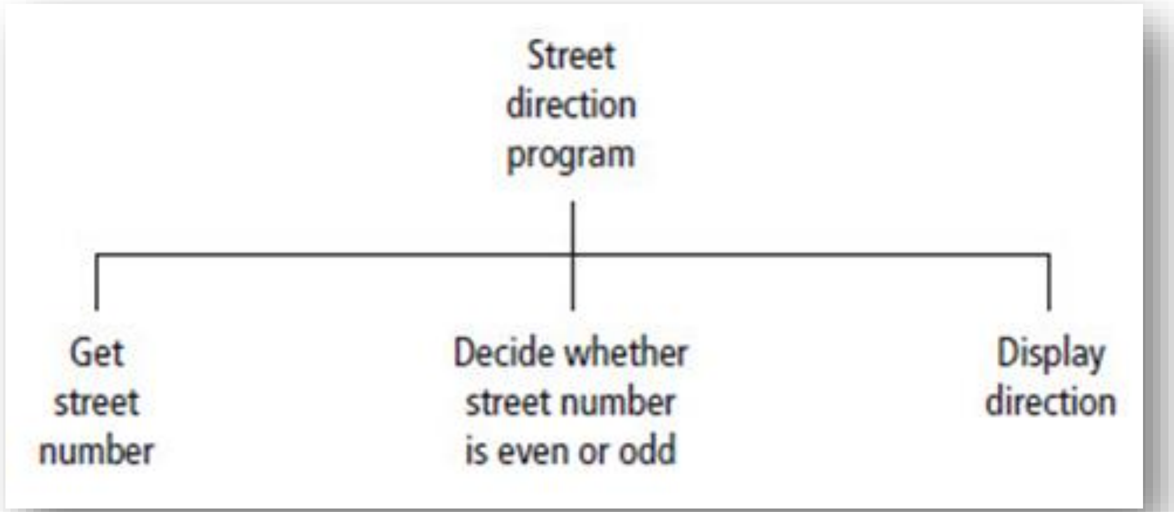


Program: Determine the direction of a numbered NYC street.

```
Get street
if street is even
   Display Eastbound
else
   Display Westbound
```

Pseudocode for the numbered New York City streets problem.

Street Direction Algorithm

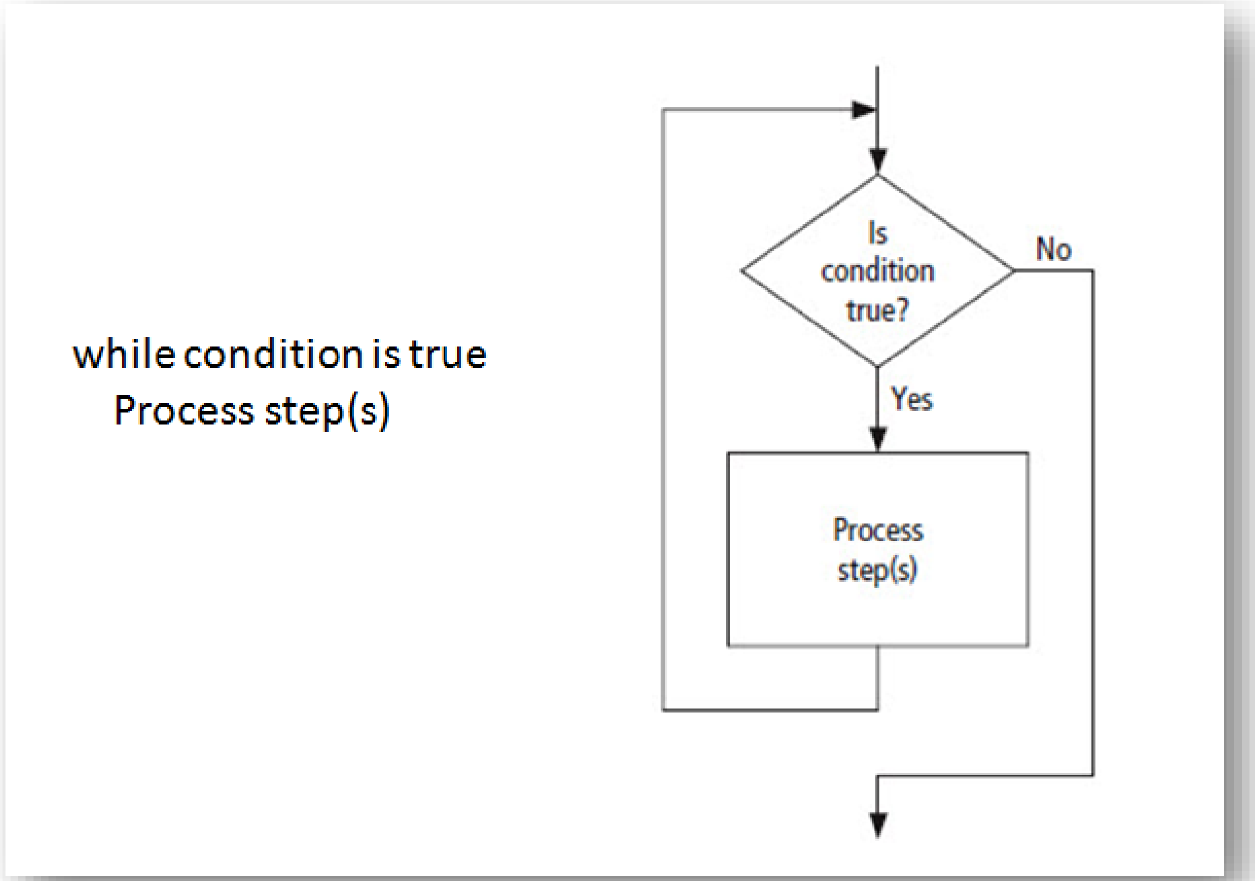


Hierarchy chart for the numbered New York City streets problem.

Repetition Structure

- A programming structure that executes instructions many times.
- Need a test (or condition) to tell when the loop should end
  - Check condition before each pass through loop

Repetition Structure

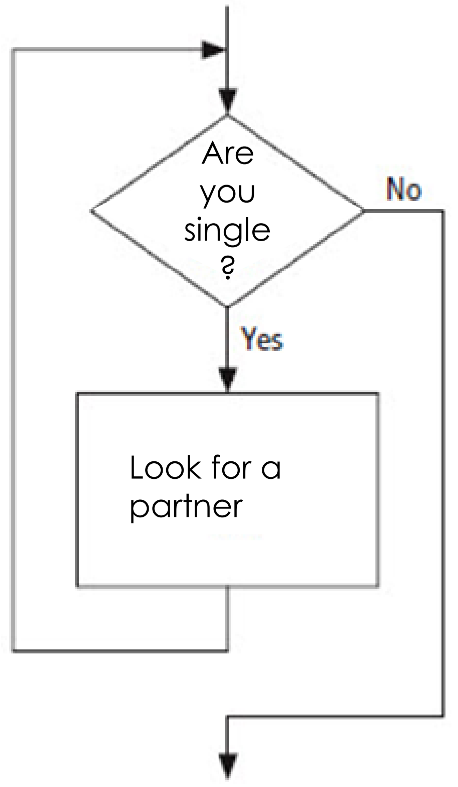


```
while condition is true
    Process step(s)
```

Pseudocode and flowchart for a loop.

Dating algorithm

```
While you are single
        Look for a partner
```



Class Average Algorithm

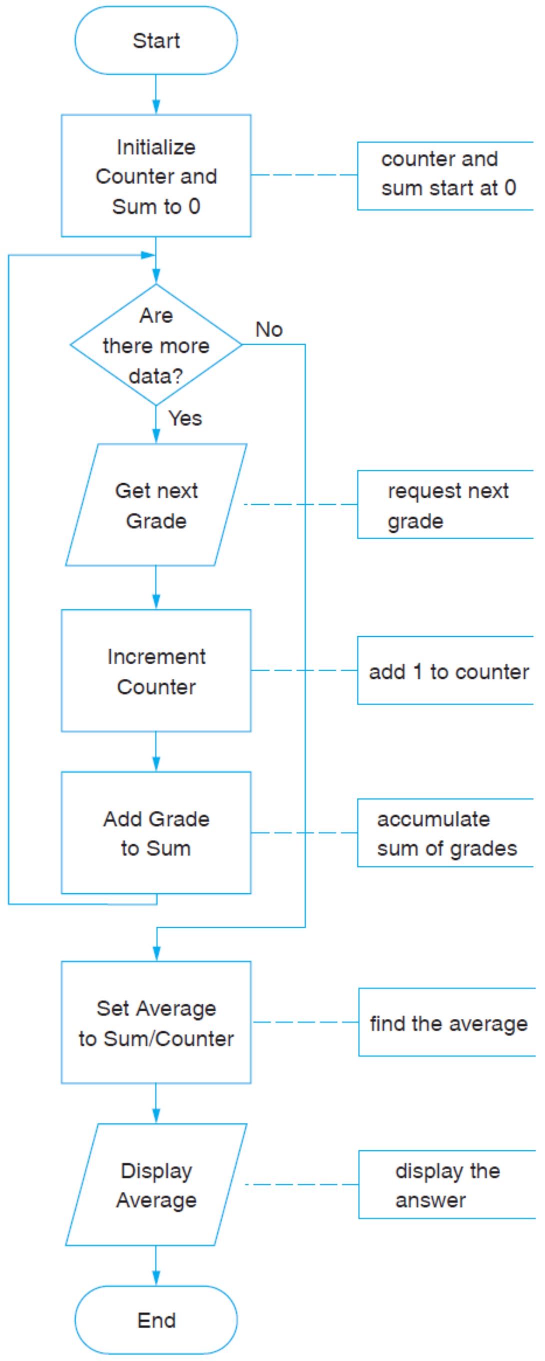
- ***Problem:*** Calculate and report the average grade for a class.

- ***Discussion:*** Average grade equals sum of all grades divided by number of students.
  - Need loop to read and then add (accumulate) grades for each student in class.
  - Inside the loop, we also need to total (count) number of students in class.

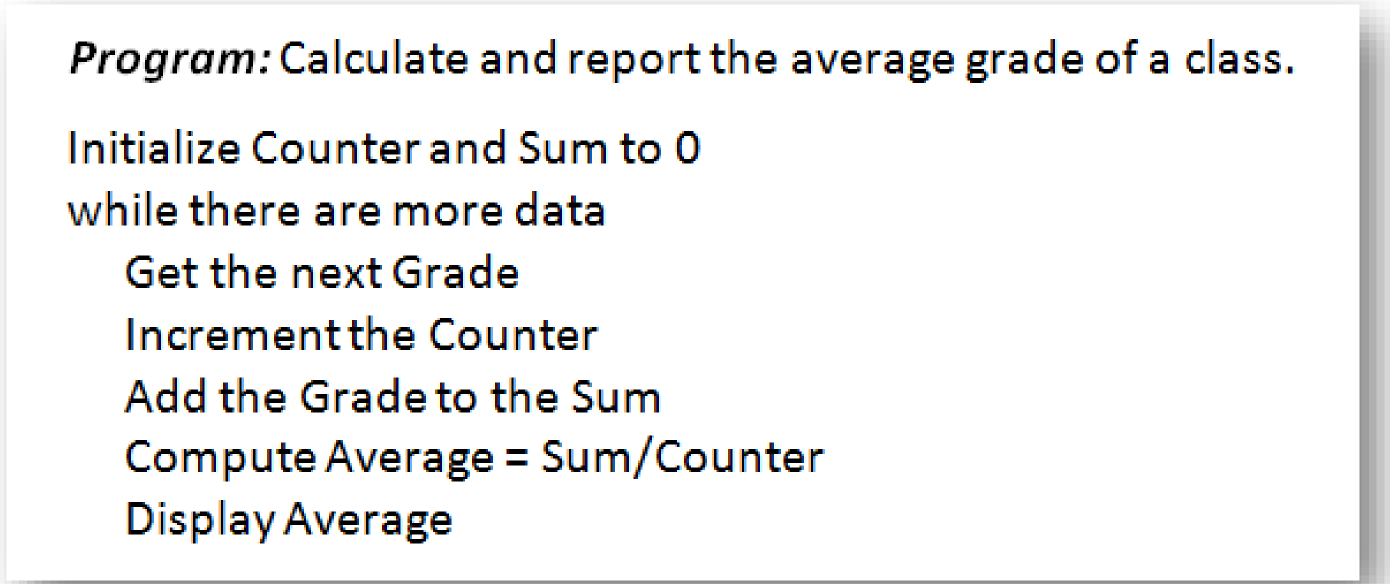
Class Average Algorithm

- ***Input:*** Student grades.
- ***Processing:*** Find the sum of the grades; count the number of students; calculate average grade = sum of grades / number of students.
- ***Output:*** Average grade

Class Average Algorithm



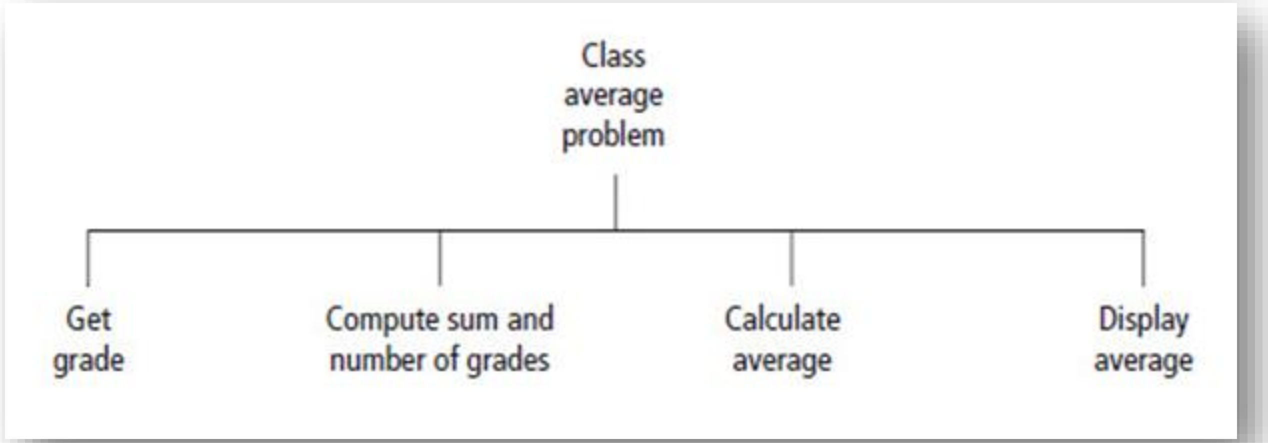
Class Average Algorithm



```
*Program:* Calculate and report the average grade of a class.

Initialize Counter and Sum to 0
while there are more data
    Get the next Grade
    Increment the Counter
    Add the Grade to the Sum
    Compute Average = Sum/Counter
    Display Average
```

Pseudocode for the class average problem.

Class Average Algorithm



Hierarchy chart for the class average problem.

Lab 1

Objective

- Install python
- The print, input functions
- Save & run programs
- Good programming practices

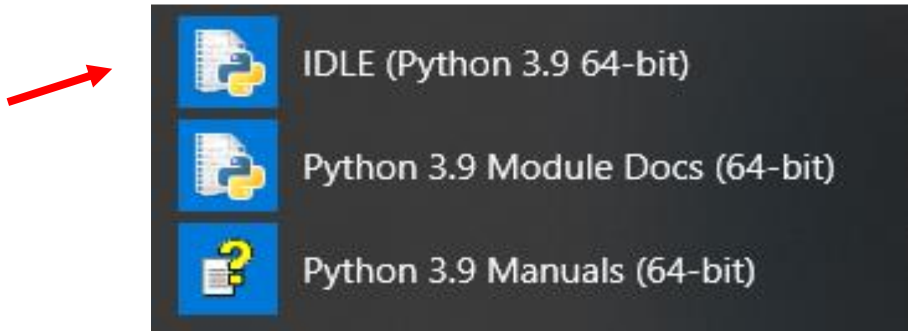
Python installation

- Install Python
  - https://www.python.org/downloads/

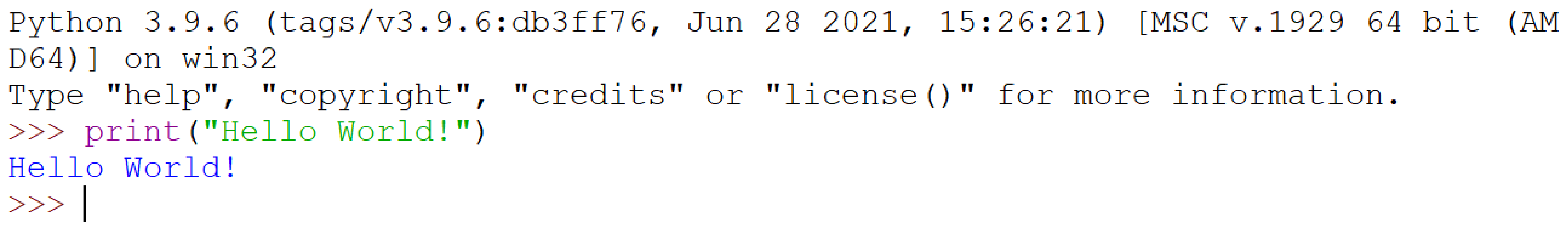
- Versions:
  - Some operating systems come with earlier versions of Python by default.
  - All material in this class will be based on **Python 3.9.6**.

Open IDLE

- After installation, open IDLE.
  - IDLE stands for **I**ntegrated **D**eve**L**opment **E**nvironment

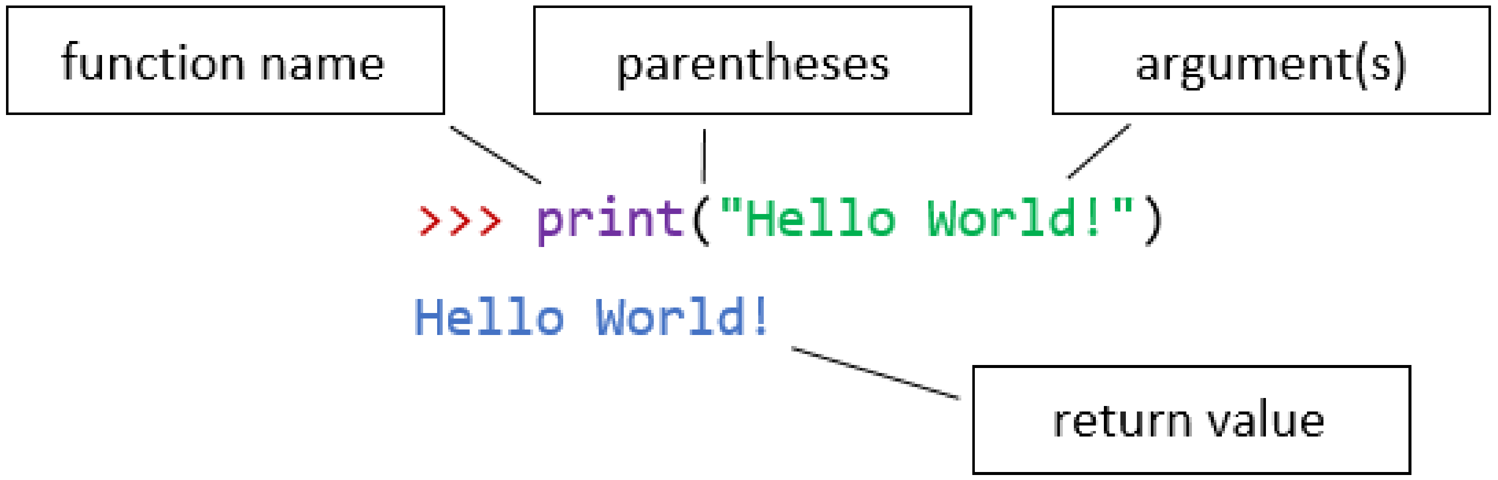


Hello World!



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello World!")
Hello World!
>>> 
```

The print function



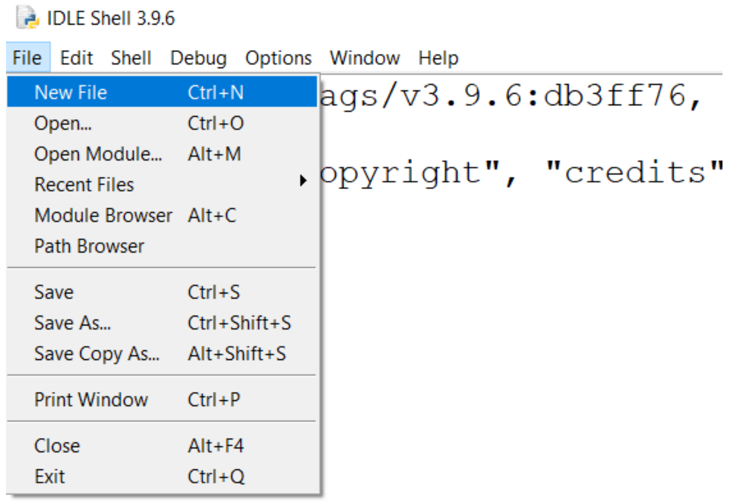
```
>>> print("Hello World!")
Hello World!
```

Good programming practices

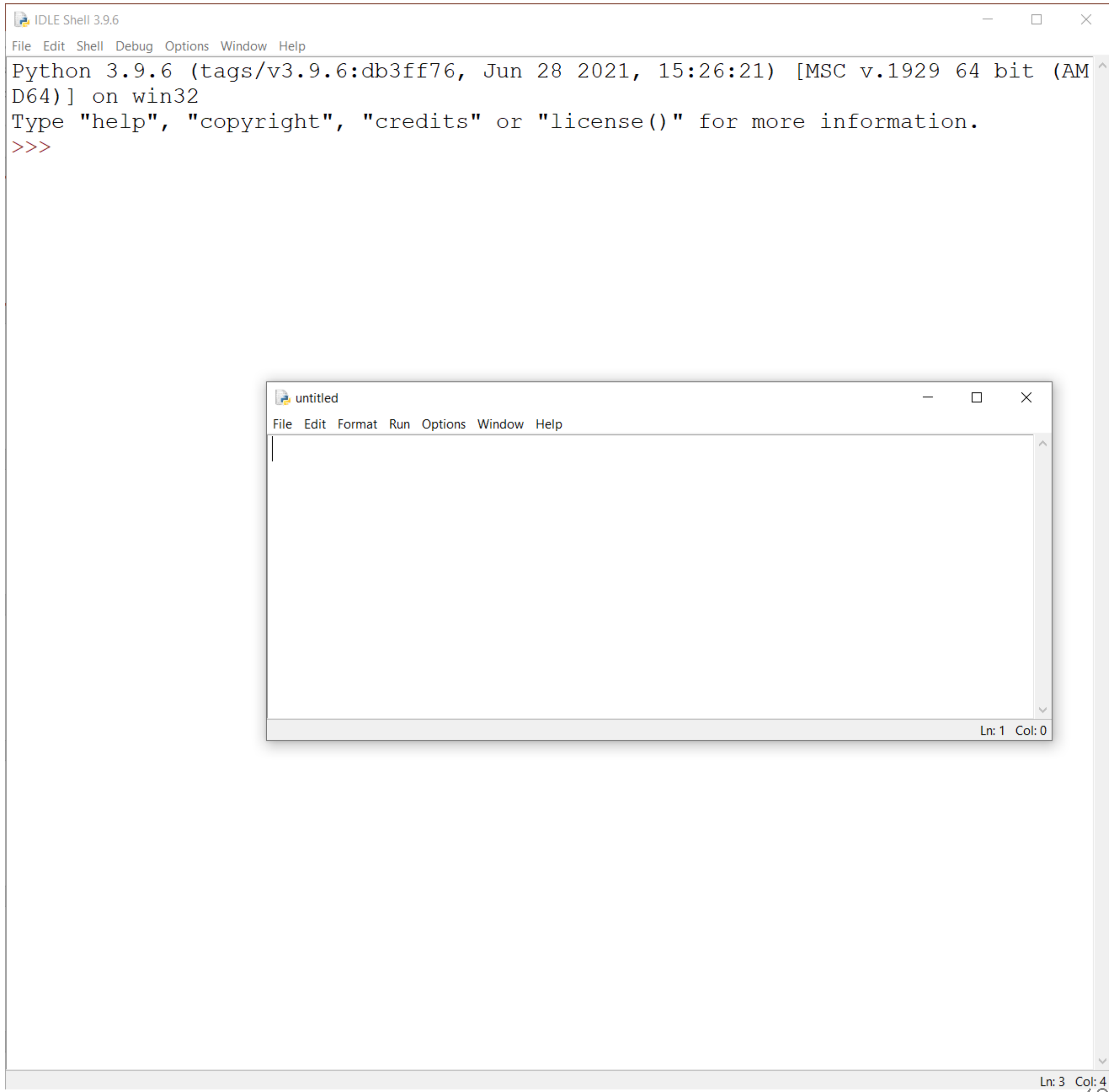
- There are a number of symbols in programming languages which work in pairs, e.g. parentheses () and quotations marks “”.
- To prevent omission of the closing pair, it is a common practice for programmers to **complete the pair first**, before filling in the contents.

```
print()      ➔      print("")      ➔      print("Hello World!")
```

Editor

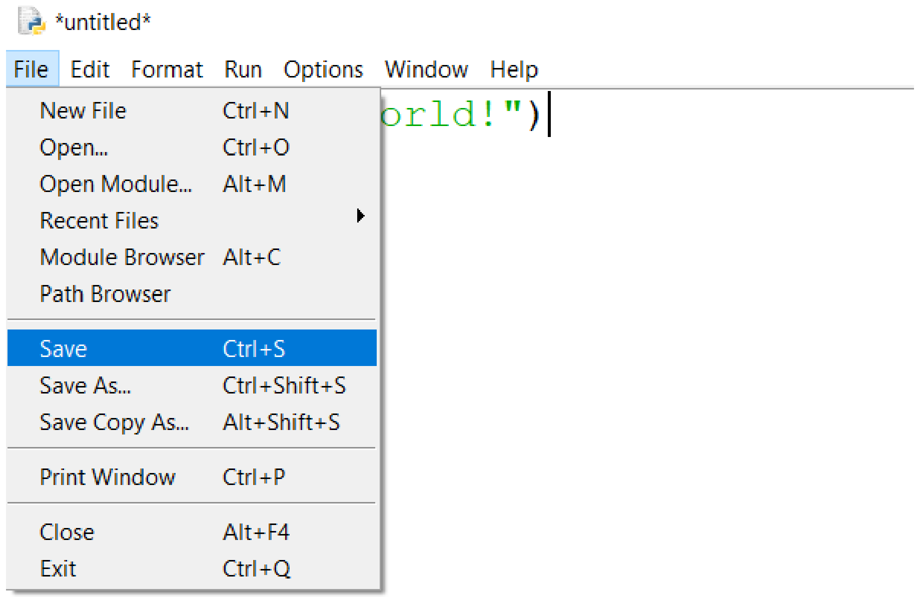


```
ags/v3.9.6:db3ff76,
opyright", "credits"
```

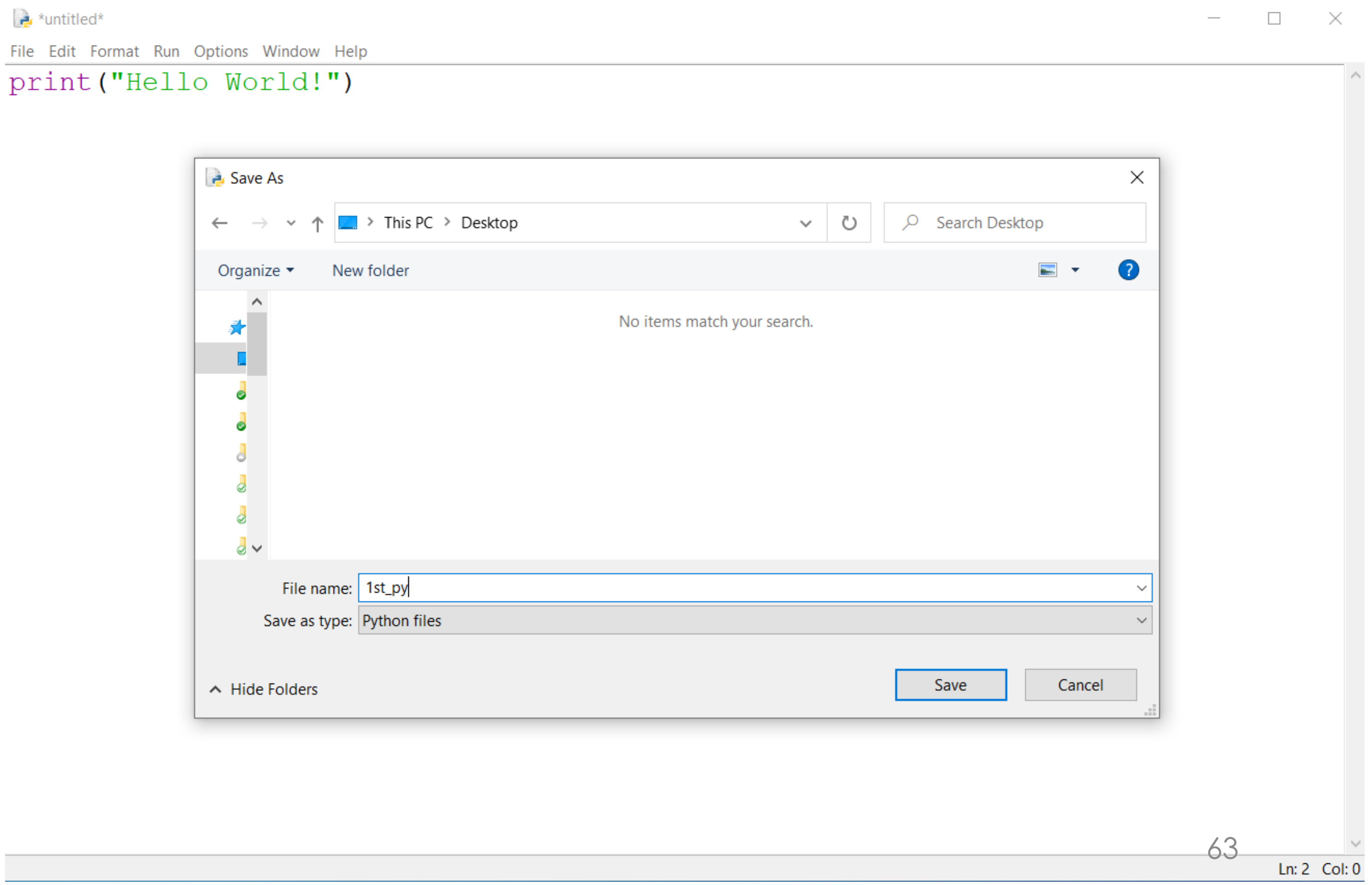


```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Save program



```
orld!")
```

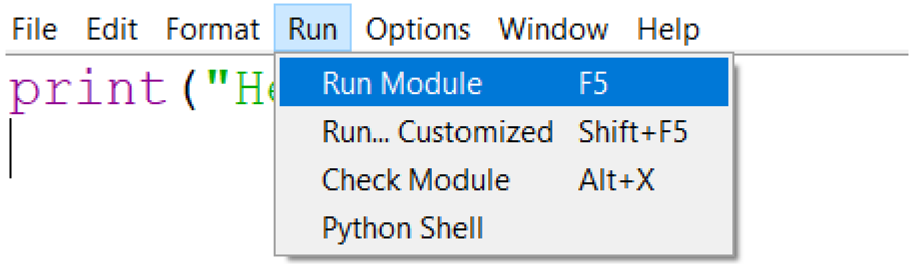


```
print("Hello World!")
```

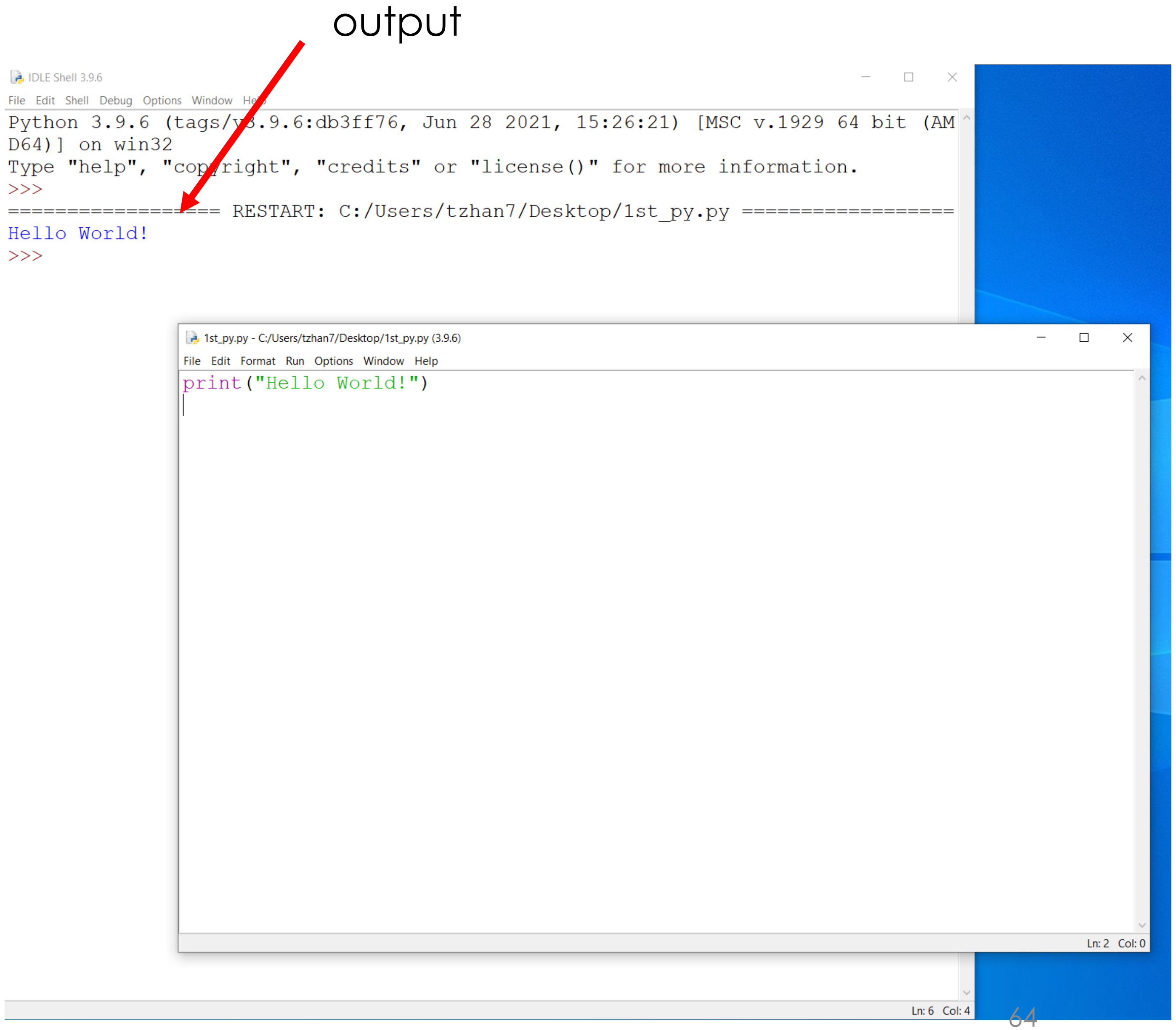
File name: 1st_py

Save as type: Python files

Run program



```
print("He
```

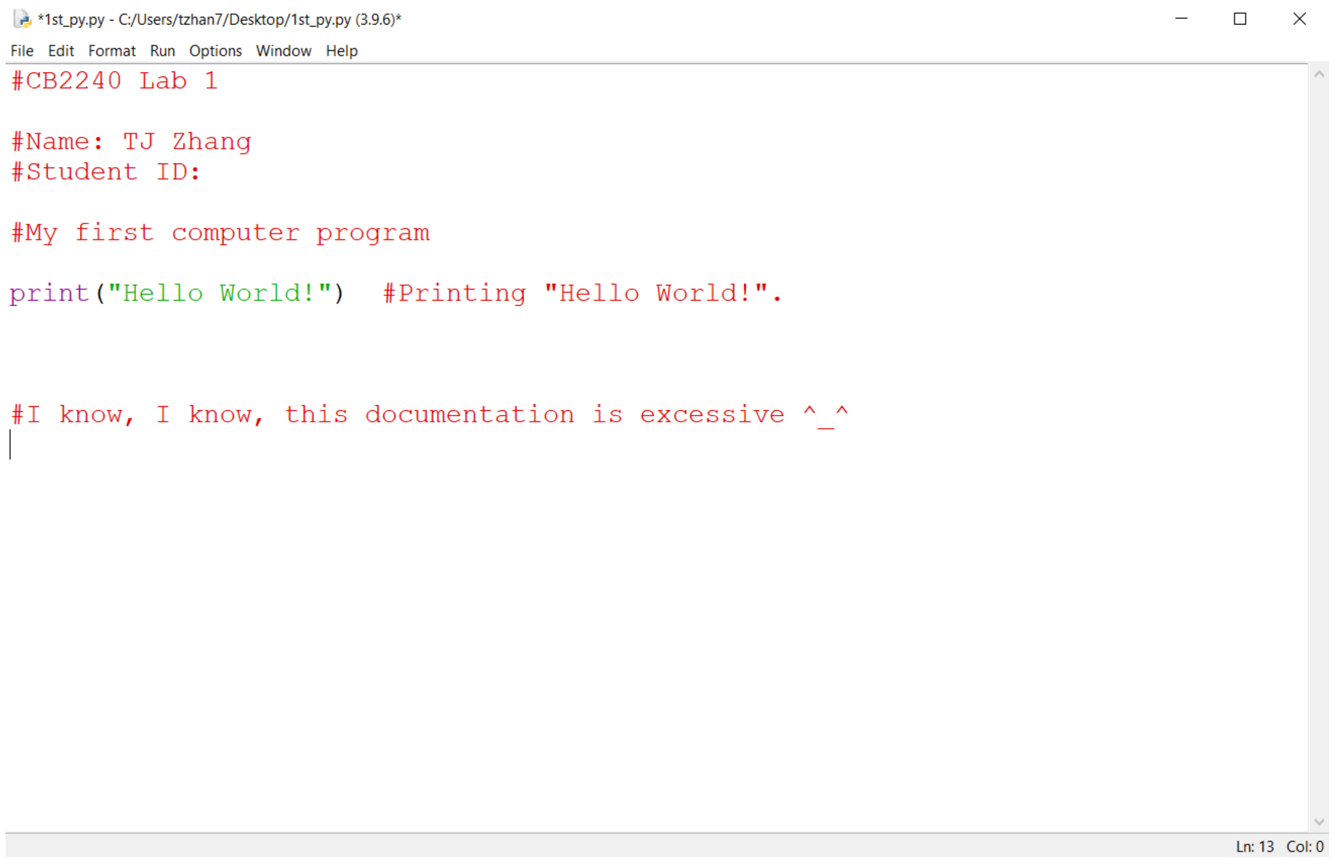

IDLE Shell 3.9.6
File Edit Shell Debug Options Window Help
1st_py.py - C:/Users/tzhan7/Desktop/1st_py.py (3.9.6)
File Edit Format Run Options Window Help
Ln: 2 Col: 0
Ln: 6 Col: 4

```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:/Users/tzhan7/Desktop/1st_py.py ==================
Hello World!
>>>
```

```
print("Hello World!")
```

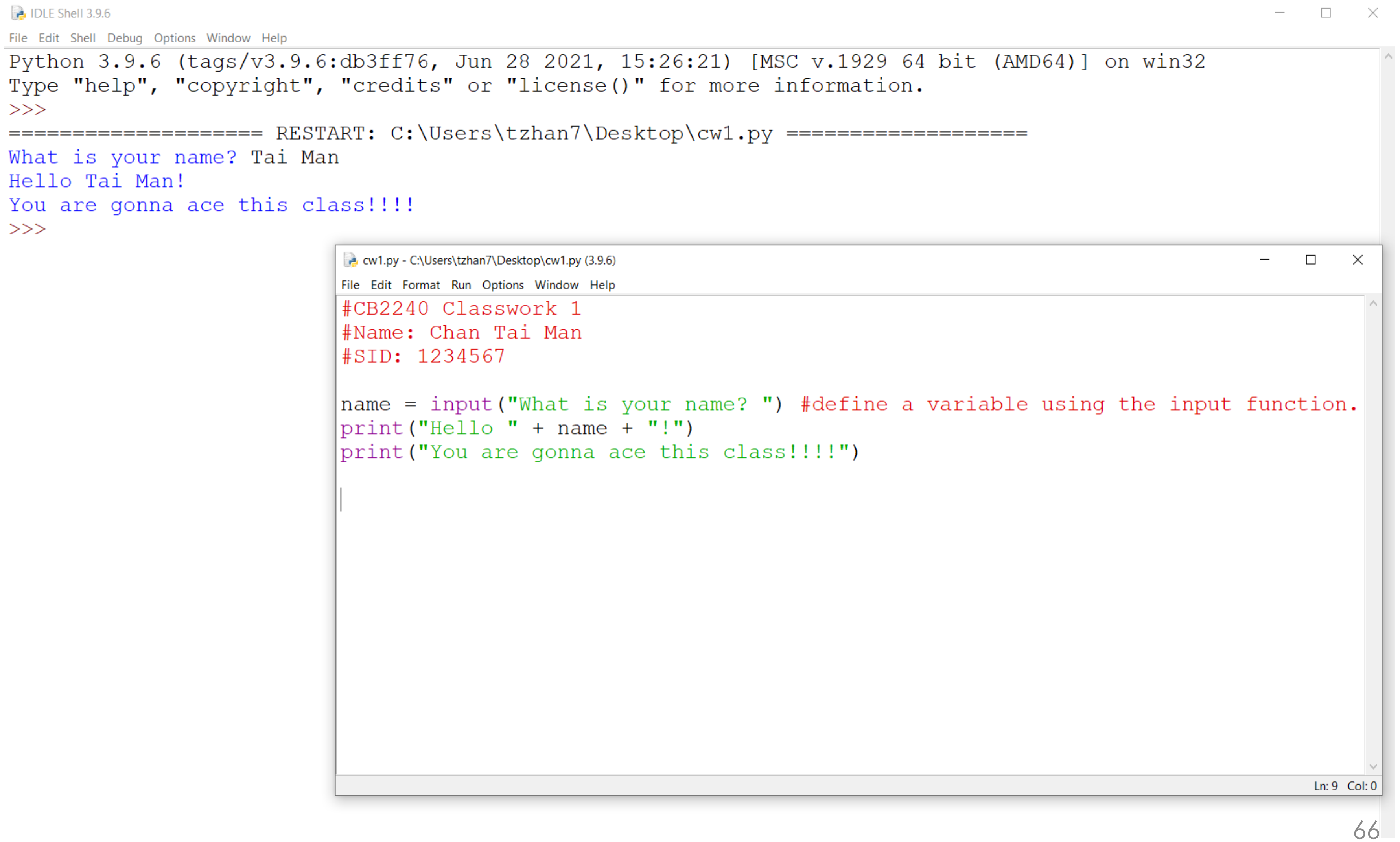
Documentation

- Document what you did in your program so that it will be more understandable by other people or by yourself in the future.
- Use # for comments in Python.
- Python will skip the red part when running the program.


1st_py.py - C:/Users/tzhan7/Desktop/1st_py.py (3.9.6)
File Edit Format Run Options Window Help
Ln: 13 Col: 0

```
#CB2240 Lab 1

#Name: TJ Zhang
#Student ID:

#My first computer program

print("Hello World!")  #Printing "Hello World!".



#I know, I know, this documentation is excessive ^_^
```

The input function

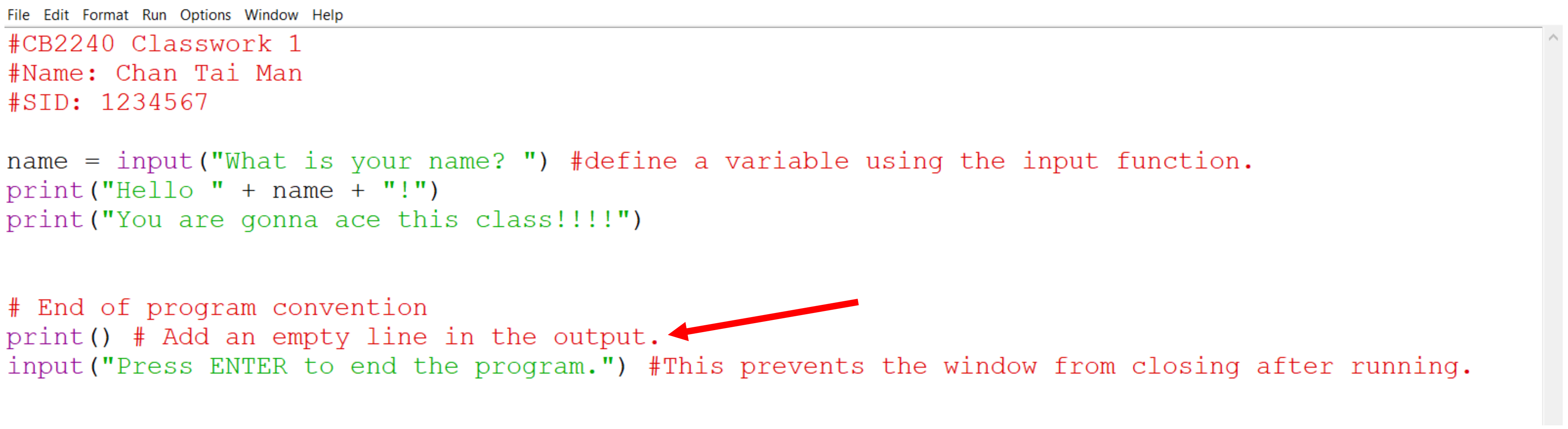

IDLE Shell 3.9.6
File Edit Shell Debug Options Window Help

```
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.
>>> 
==================== RESTART: C:\Users\tzhan7\Desktop\cw1.py ===================
What is your name? Tai Man
Hello Tai Man!
You are gonna ace this class!!!!
>>> 
```

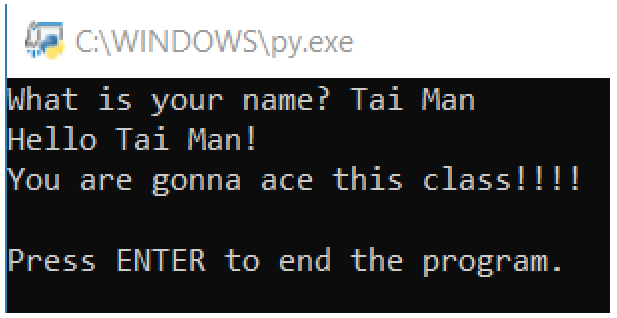
cw1.py - C:\Users\tzhan7\Desktop\cw1.py (3.9.6)
File Edit Format Run Options Window Help
Ln: 9 Col: 0

```python
#CB2240 Classwork 1
#Name: Chan Tai Man
#SID: 1234567

name = input("What is your name? ") #define a variable using the input function.
print("Hello " + name + "!")
print("You are gonna ace this class!!!!")
```

End of program convention



```
#CB2240 Classwork 1
#Name: Chan Tai Man
#SID: 1234567

name = input("What is your name? ") #define a variable using the input function.
print("Hello " + name + "!")
print("You are gonna ace this class!!!!")


# End of program convention
print() # Add an empty line in the output.
input("Press ENTER to end the program.") #This prevents the window from closing after running.
```

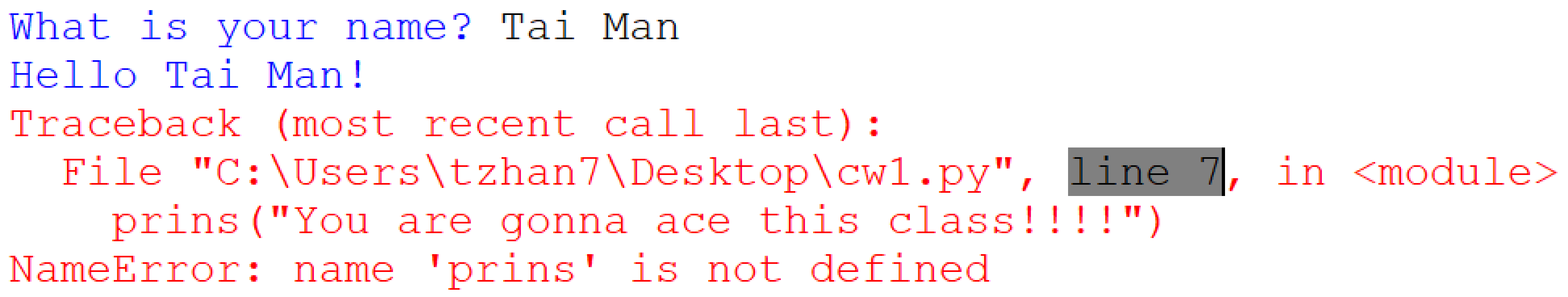


```
What is your name? Tai Man
Hello Tai Man!
You are gonna ace this class!!!!

Press ENTER to end the program.
```

Submit the previous .py file as Classwork 1.

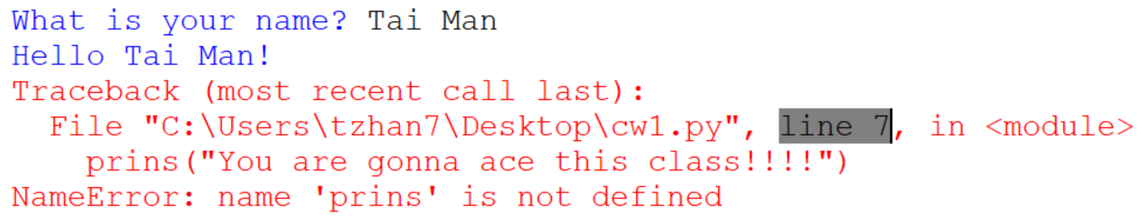
Errors

- Don’t panic.
- Read the error message.
- Find the corresponding line and correct it.

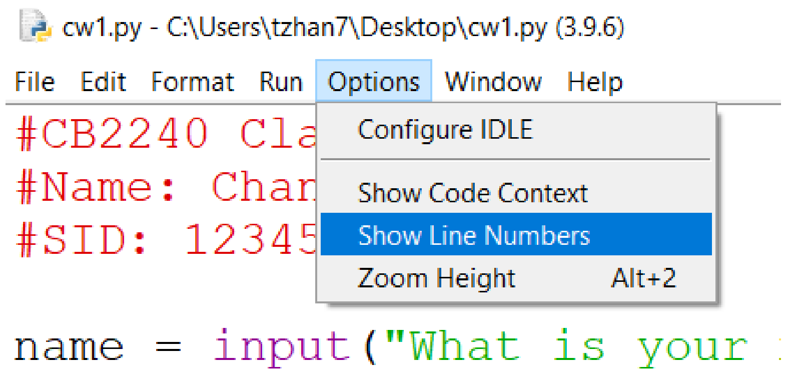


```
What is your name? Tai Man
Hello Tai Man!
Traceback (most recent call last):
  File "C:\Users\tzhan7\Desktop\cw1.py", line 7, in <module>
    prins("You are gonna ace this class!!!!")
NameError: name 'prins' is not defined
```

Show line numbers

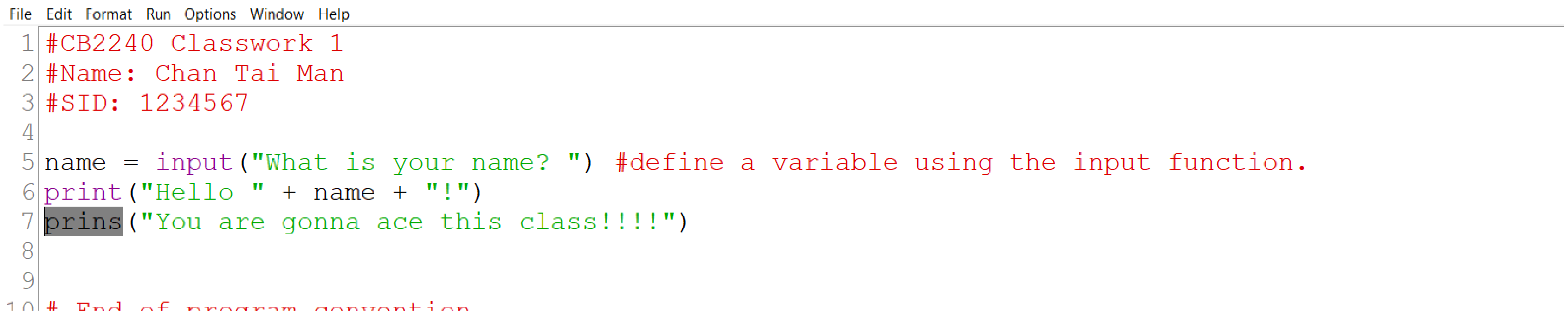


```
What is your name? Tai Man
Hello Tai Man!
Traceback (most recent call last):
  File "C:\Users\tzhan7\Desktop\cw1.py", line 7, in <module>
    prins("You are gonna ace this class!!!!")
NameError: name 'prins' is not defined
```



cw1.py - C:\Users\tzhan7\Desktop\cw1.py (3.9.6)

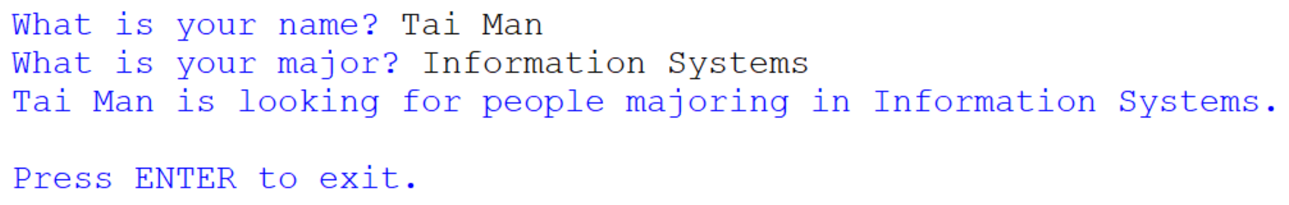
```
#CB2240 Cla
#Name: Chan
#SID: 12345

name = input("What is your
```



```
 1 #CB2240 Classwork 1
 2 #Name: Chan Tai Man
 3 #SID: 1234567
 4
 5 name = input("What is your name? ") #define a variable using the input function.
 6 print("Hello " + name + "!")
 7 prins("You are gonna ace this class!!!!")
 8
 9
10 # End of program convention
```

Homework 1

- Write a program that asks for two inputs:
  - Name
  - Major
- Output a sentence as shown on the right.
- Remember to include your name and ID in the documentations.
- Upload your .py file and code output file (.jpg) to Canvas.



```
What is your name? Tai Man
What is your major? Information Systems
Tai Man is looking for people majoring in Information Systems.

Press ENTER to exit.
```