# BASIC OF ALGORITHMS

PROGRAMMING TOOLS

NOR HAMIZAH MISWAN

- Algorithms were used for many thing such as making decision and performing tasks.

- Suppose a friend asks you to determine number of stamps to place on an envelope where for every five pages of paper in the envelope, one stamp will be required:

  1. Request the number of sheets of paper; call it **Sheets**. (*Input*)
  2. Divide **Sheets** by 5. (*Processing*)
  3. If necessary, round it up to a whole number; call it **Stamps**. (*Processing*)
  4. Reply with the number of **Stamps**. (*Output*)

- For example, for 16 sheets of paper:

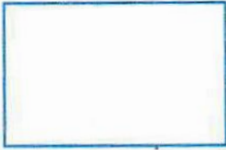| Input (16) | Processing (formula) | Output (4) |
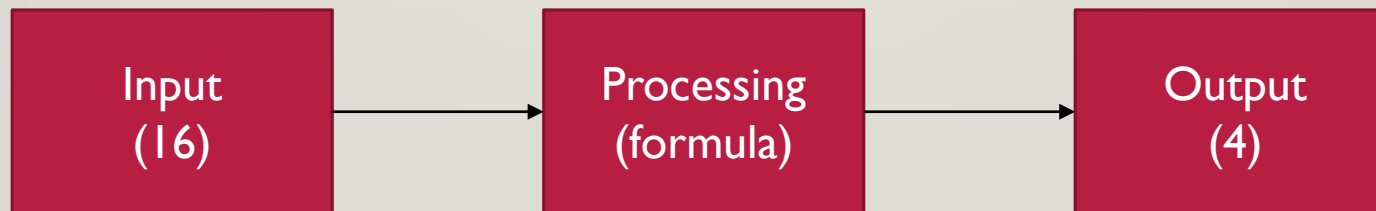
- The most popular program design tools are:
  - **Flowcharts**: Graphically depict the logical steps to carry out a task and show steps relate to each other.
  - **Pseudocode**: Uses English-like phrases with some programming language terms to outline the task
  - **Hierarchy Charts**: Show how the different parts of a program relate to each other.

FLOWCHARTS

- Consists of special geometric symbols connected by arrows.

- Within each symbol is a phrase representing the activity.

- The shape indicates the type of operation.

- The arrow connecting the symbols, called **flowlines**.

- It should flow from top to the bottom.

- The flowchart provides a graphical representation of the task, thereby making it easier to follow.

- However, for a very large program, the flowcharts will be very large which require several pages, make it harder to follow and modify.

Symbol	Name	Meaning
→	*Flowline*	Used to connect symbols and indicate the flow of logic.
(rounded rectangle)	*Terminal*	Used to represent the beginning (Start) or the end (End) of a task.
(parallelogram)	*Input/Output*	Used for input and output operations. The data to be input or output is described in the parallelogram.
(rectangle)	*Processing*	Used for arithmetic and data-manipulation operations. The instructions are listed inside the rectangle.
(diamond)	*Decision*	Used for any logic or comparison operations. Unlike the input/output and processing symbols, which have one entry and one exit flowline, the decision symbol has one entry and two exit paths. The path chosen depends on whether the answer to a question is "yes" or "no."
(circle)	*Connector*	Used to join different flowlines.
(annotation bracket)	*Annotation*	Used to provide additional information about another flowchart symbol.

Flowchart for the postage-stamp problem:

Start

Obtain Sheets — — — input

Set Stamps = Sheets / 5 — — — processing

Round Stamps up to a whole number — — — processing

Display Stamps — — — output

End
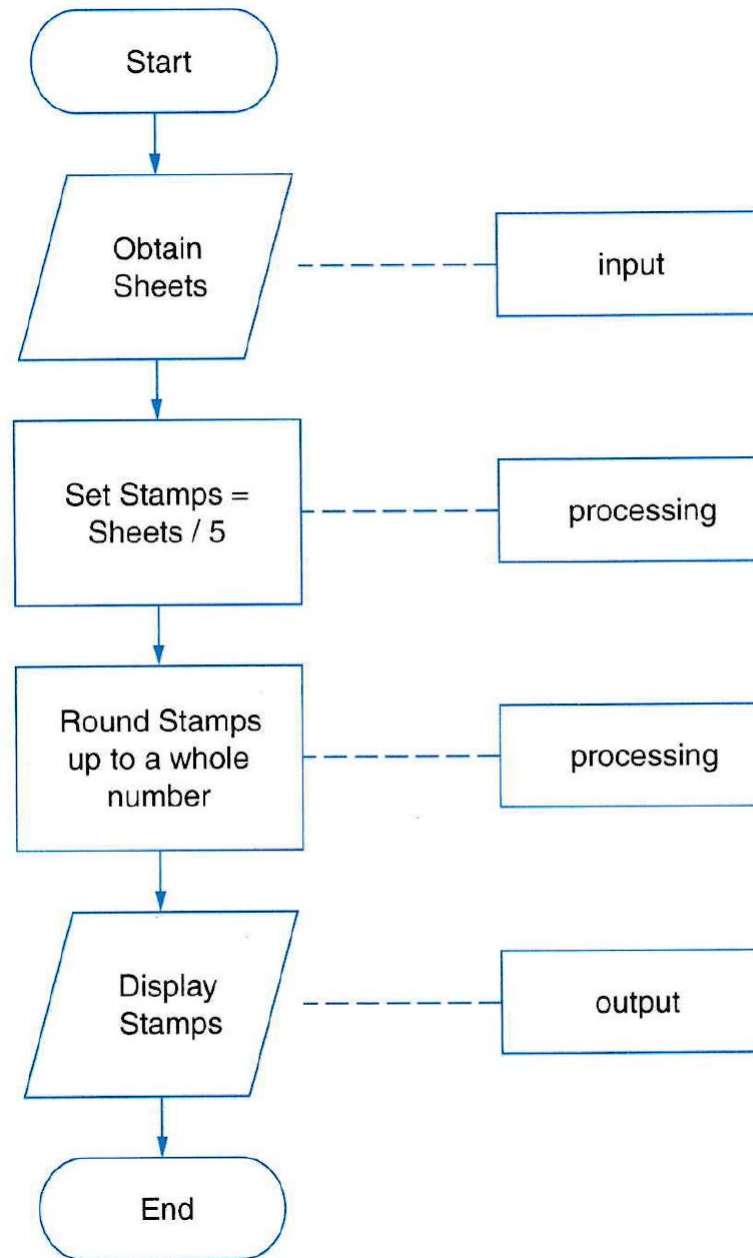
PSEUDOCODE

- An abbreviated plain English version of actual computer code.

- English statements will replace the flowcharts that outline the process.

- It look more like computer code.

- It allows the programmer to focus on the steps rather than on the coding language.

- Once prepared, it can be easily translated into the coding language.

- It is compact and looks like the code to be written.

Pseudocode for the postage stamp problem

Program: Determine the proper number of stamps for an envelope.

Obtain number of sheets (Sheets)   (*input*)

Set the number of stamps to sheets / 5   (*processing*)

Round the number of stamps up to a whole number   (*processing*)

Display the number of stamps   (*output*)

Algorithm 1 MICE algorithm

Require: Original data, D

Begin

Let the variable be X_a^b with $b = 1, ..., n$ missing series in every $a = 1, ..., m$ variable that has missing values

Perform the mean imputation for each variable a

for *each variable a,* **do**

    Calculate mean$[a]$= (sum of non b)/(Total number of non b)

    Fill the corresponding mean$[a]$ in the corresponding b

end

Perform imputation for every specified cycle C where $C = 1, ..., c$

for *each cycles C,* **do**

    for *each variable a ("one variable"),* **do**

        Set back the mean imputed cells for "one variable" in a to missing

        Perform regression on "one variable"

        Impute the missing value in "one variable" from the predicted regression model

    end

end

HIERARCHY CHART

- Also known as structure charts, HIPO (Hierarchy plus Input-Process-Output) charts, top-down charts or VTOC (Visual Table of Contents) charts.

- It depict the organization of a program but omit the specific processing language.

- It describe what each part of the program does and how the parts relate to each other.

- It read from top to bottom and from left to right.

- Each part may be divided into a succession of subparts that branch out under it.

- After the activities in the succession of subparts are carried out, the part to the right of the original part is considered.

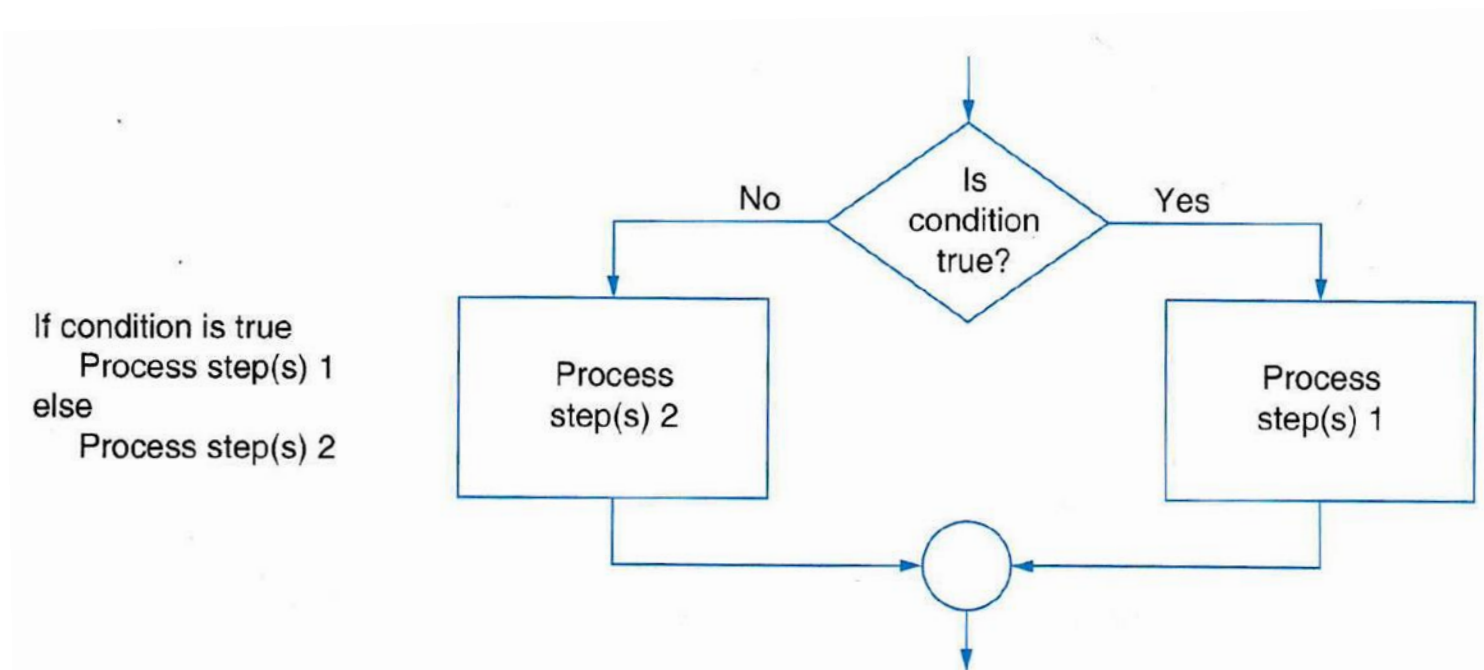# Hierarchy chart form the postage-stamp problem
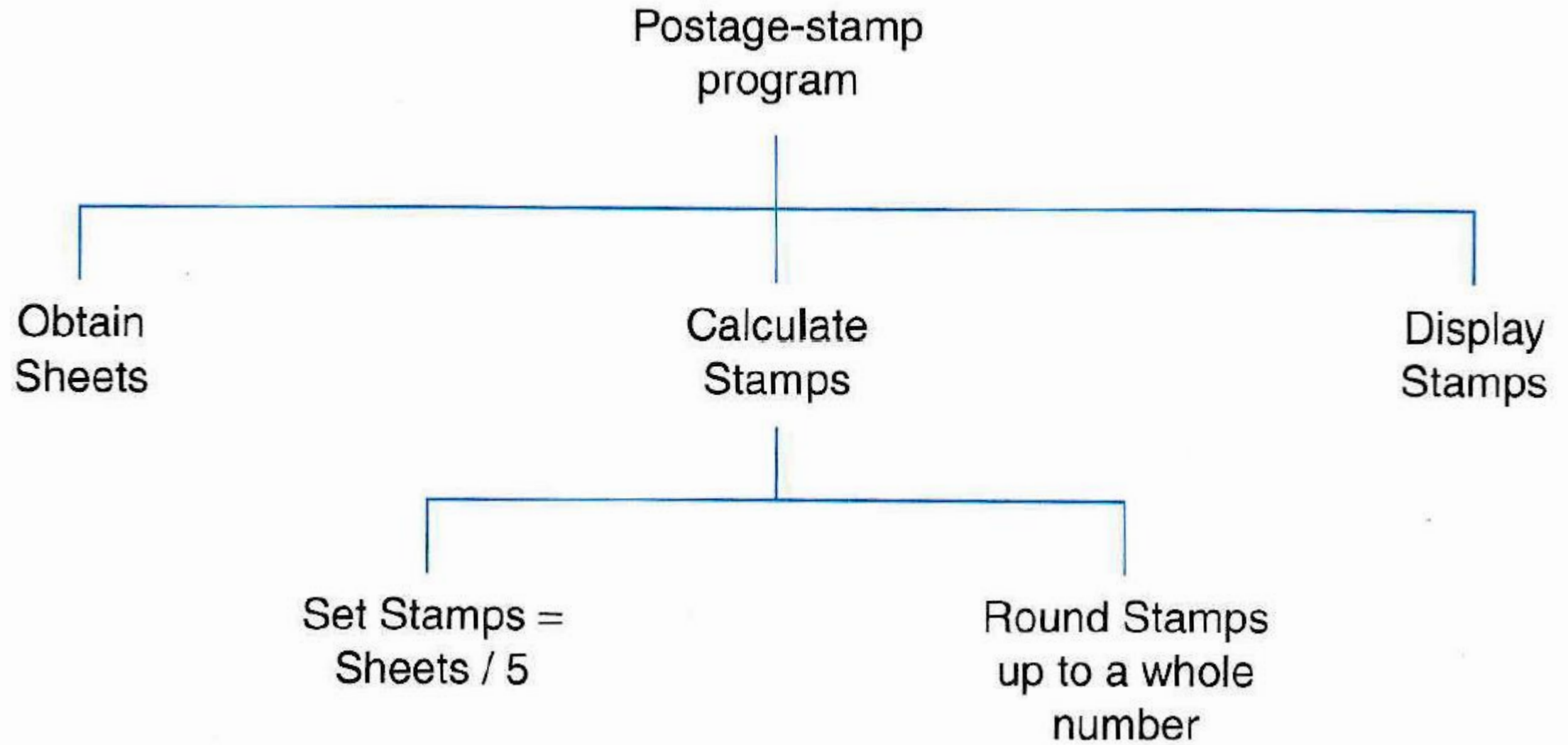
- The advantage is in the initial planning of a program.

- It allow us to break down the major parts of a program so we can see what must be done in general.

- Then, each part can be refined into more detailed plans using flowchart or pseudocode.

- This process is called the **divide and conquer** method.
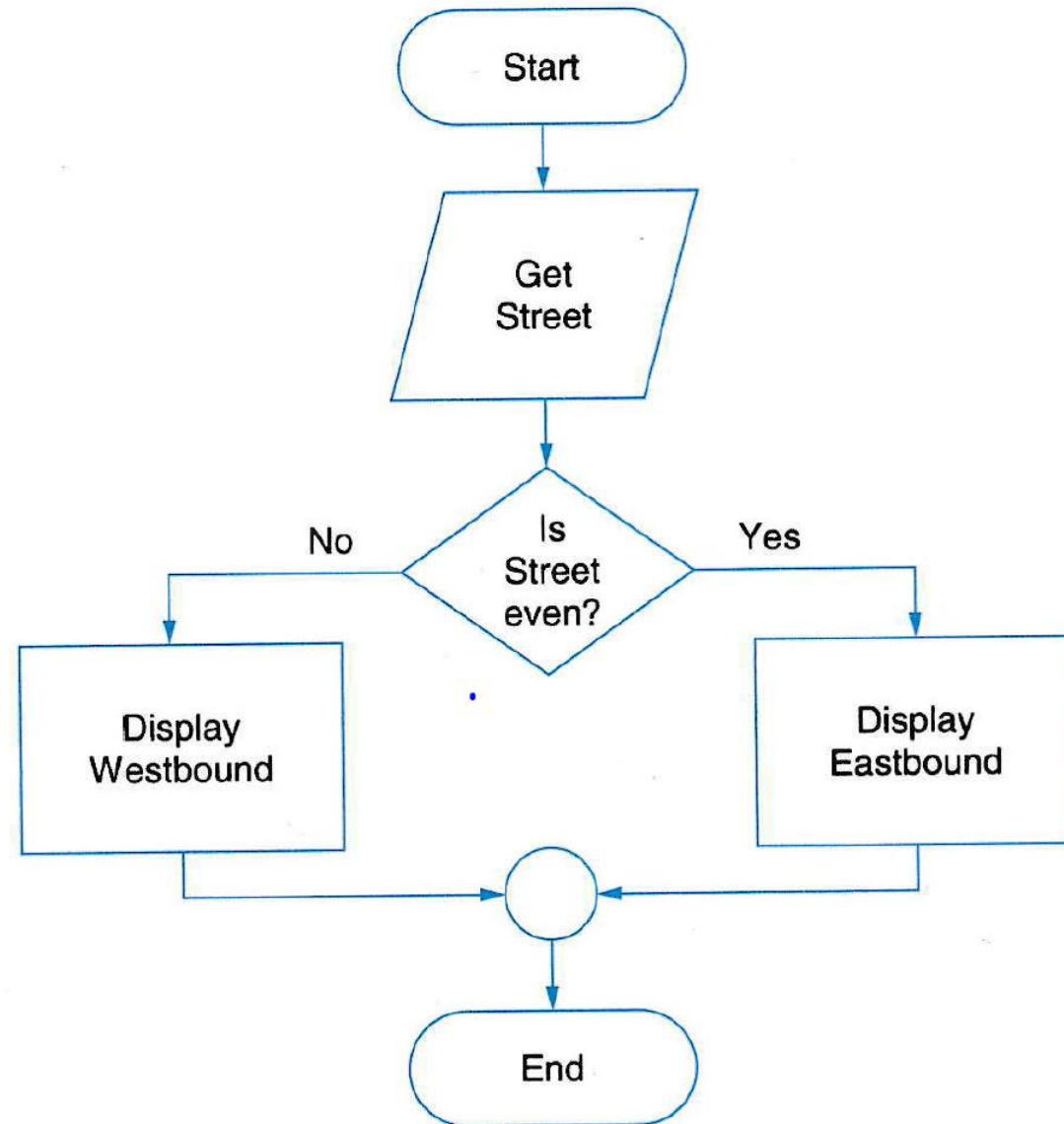
Decision Structure

- If the problem was solved by a series of instructions without skipping any lines, it is called a **sequence structure**.

- However, many problems require a decision whether a series of instructions should be executed. If "No" the another is executed.

- This is called a **decision structure**.

If condition is true
    Process step(s) 1
else
    Process step(s) 2

No — Is condition true? — Yes

Process step(s) 2

Process step(s) 1

EXAMPLE: DIRECTION OF NUMBERED NYC STREETS ALGORITHM

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

- **Input**: Street number.

- **Processing**: Decide if the street number is divisible by 2.

- **Output**: "Eastbound" or "Westbound"

Flowchart

PSEUDOCODE

Program: Determine the direction of a numbered NYC street.
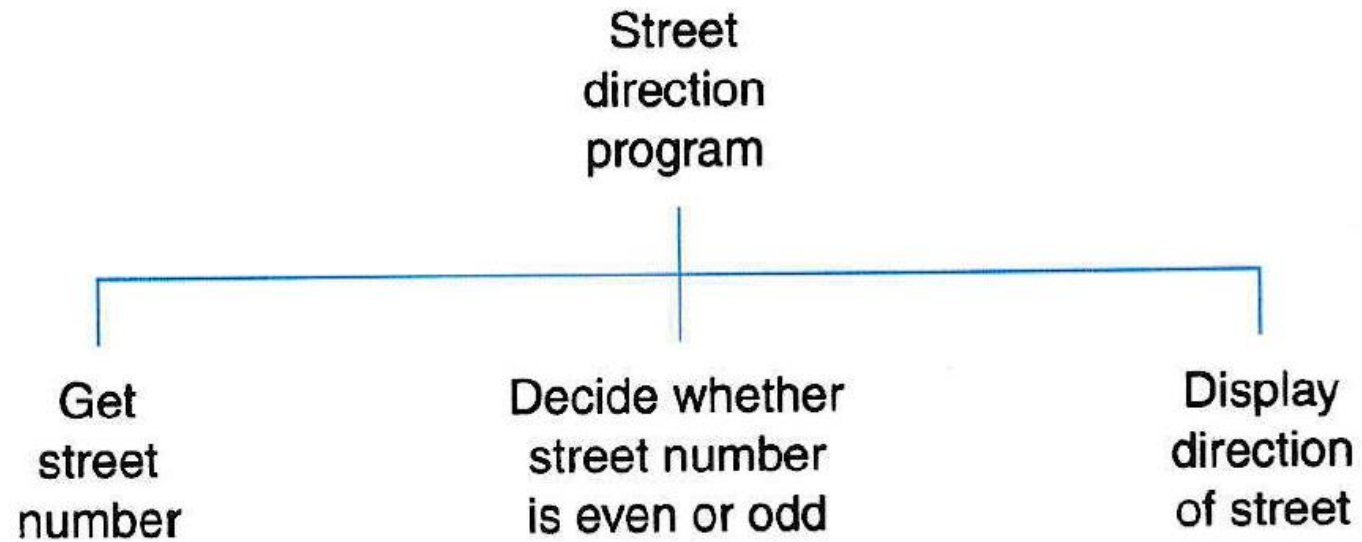
Get street

if street is even

       Display Eastbound

else

       Display Westbound

Hierarchy Chart

REPETITION STRUCTURE

- Executing instruction many times.

- Also known as loop structure.

- Need a test or condition to tell when the loop should end.

- At each iteration, the condition will be checked and the loop will continue if the condition is true.

Do While condition is true
    Process step(s)
Loop

Is condition true?

No

Yes

Process step(s)
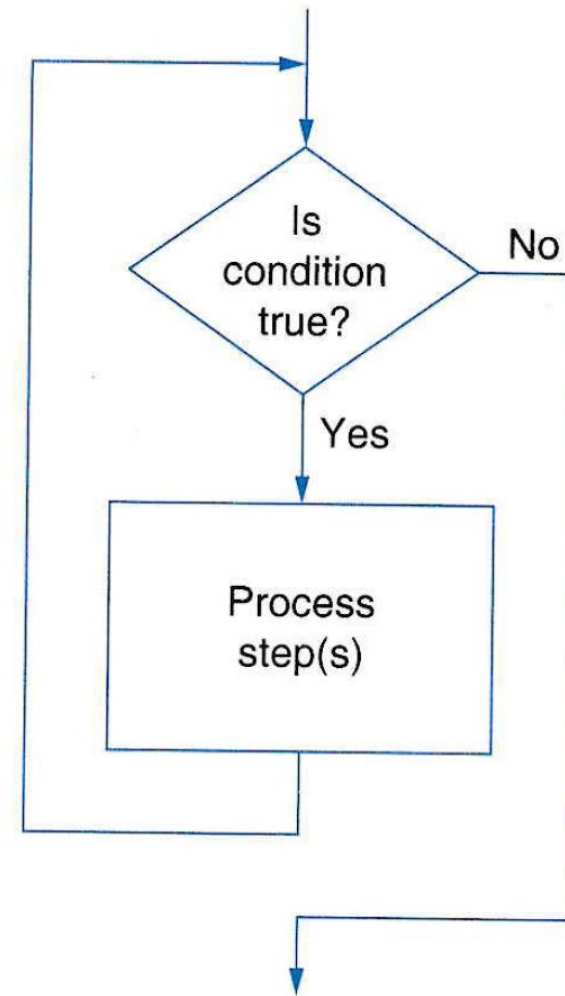
EXAMPLE: CLASS AVERAGE ALGORITHM

Problem: Calculate and report the average grade for a class. The average equals the sum of all grades divided by the number of students.
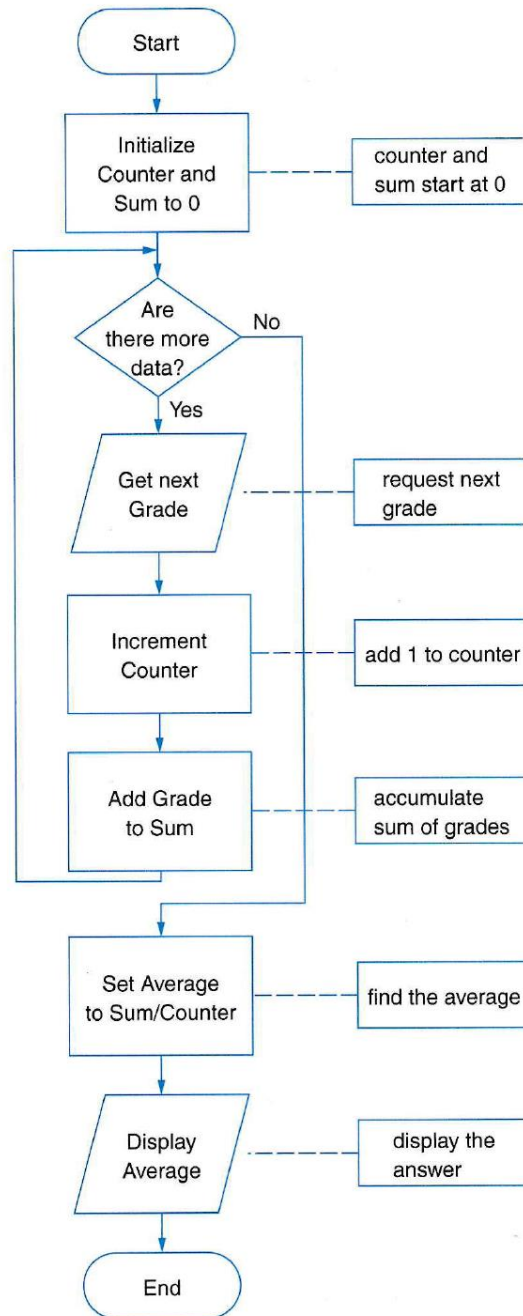
Input: Student grades.

Processing: Find the sum of the grades; count the number of students; calculate average grade

Output: Average grade

Flowchart

PSEUDOCODE

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

Initialize Counter and Sum to 0

while there are more data

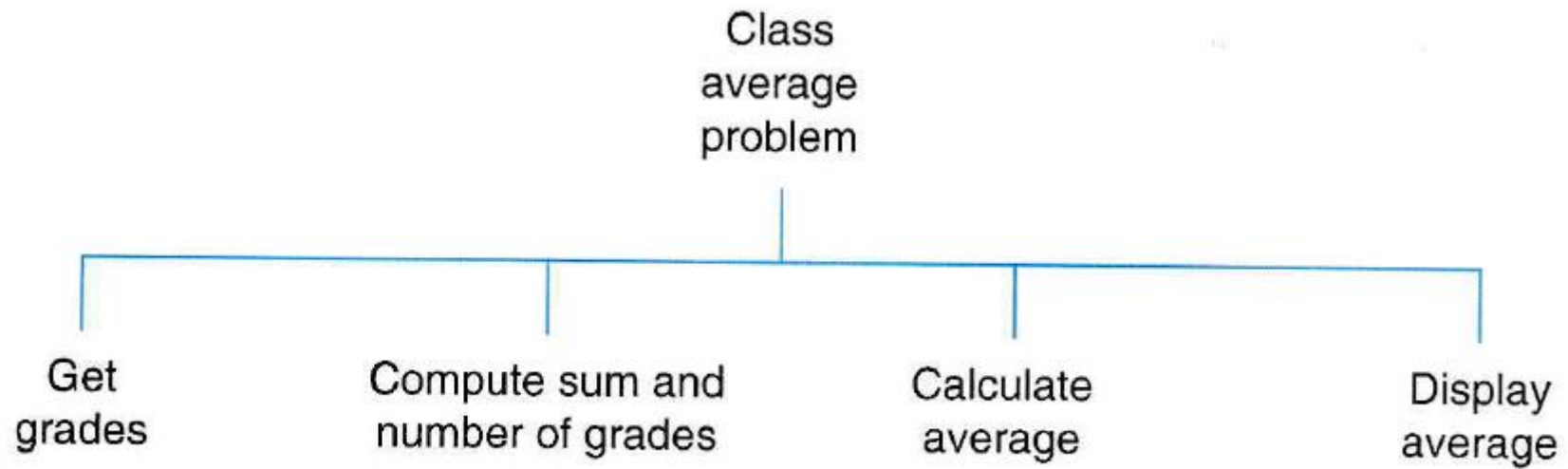   Get the next Grade

      Increment the Counter

      Add the Grade to the Sum

Set Average to Sum/Counter

Display Average

Hierarchy Chart

SIDE NOTES:

- Tracing a flowchart is like playing board games. Begin at the Start symbol and proceed until reach the End symbol. At any time, we will be at just one symbol. The path taken will depends on the input.

- The algorithm should be tested at the flowchart stage before being coded. Perform *desk checking*, where different data will be used as input and check the output. Try with nonstandard data as well as typical data.

INTRODUCTION TO PYTHON

PYTHON:

- Guido van Rossum - creator for Python programming language [*https://bit.ly/3TY2G0y*]

- General purpose programming language: build anything

- Open source! Free!

- Share codes with other people

- Swiss army knife → can do almost anything

https://bit.ly/3DbiBlm

PYTHON:

- Python is a cross-platform programming language, meaning, it runs on multiple platforms like Windows, MacOS, Linux and has even been ported to the Java and .NET virtual machines. It is free and open source.

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

- Open command prompt, type cmd in your computer search

- Then, type python

```
Microsoft Windows [Version 10.0.17134.648]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>python
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52
) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more i
nformation.
>>> 1 + 1
2
>>> quit()

C:\Users\ASUS>_
```

PYTHON AS CALCULATOR

- Summation
- Substraction
- Division
- Multiplication
- Modulo
- Exponention

ADDING COMMENTS

- We can add comments to our Python scripts
- Comments are important to make sure that we and others can understand what our code is about

- To add comments, use the "**#**" tag
- Comments are not run as Python code, so they will not influence our result

INTEGRATED DEVELOPMENT ENVIRONMENT (IDE):

- An integrated development environment (IDE) is a software application that helps programmers develop software code efficiently.

- It increases developer productivity by combining capabilities such as software editing, building, testing, and packaging in an easy-to-use application.

- Just as writers use text editors and accountants use spreadsheets, software developers use IDEs to make their job easier.

- Most IDEs include functionality that goes beyond text editing. They provide a central interface for common developer tools, making the software development process much more efficient.

INTEGRATED DEVELOPMENT ENVIRONMENT (IDE):

- Among the Python IDE:

  ➢ Pycharm, Pydev, IDLE, Virtual Studio Code (VS Code), Sublime Text, Jupyter Notebook, Spyder, Wing, GNU, Thonny

  Reference: https://hackr.io/blog/best-python-ide

- Top Python IDE in data science:

  ➢ Spyder, Pycharm, Jupiter Notebook, VS Code

  Refer:

  https://www.youtube.com/watch?v=VbyGrQWxL8A

Featured Python IDEs & Editors
Pycharm: Hugely popular, powerful, and feature-rich Python IDE.
Visual Studio Code: Lightweight and extensible code editor.
Jupyter Notebook: Data Science IDE, easily run simulations and share data.

JUPYTER NOTEBOOK SETUP

Step 0 - go to the link below and download Anaconda (python 3+)
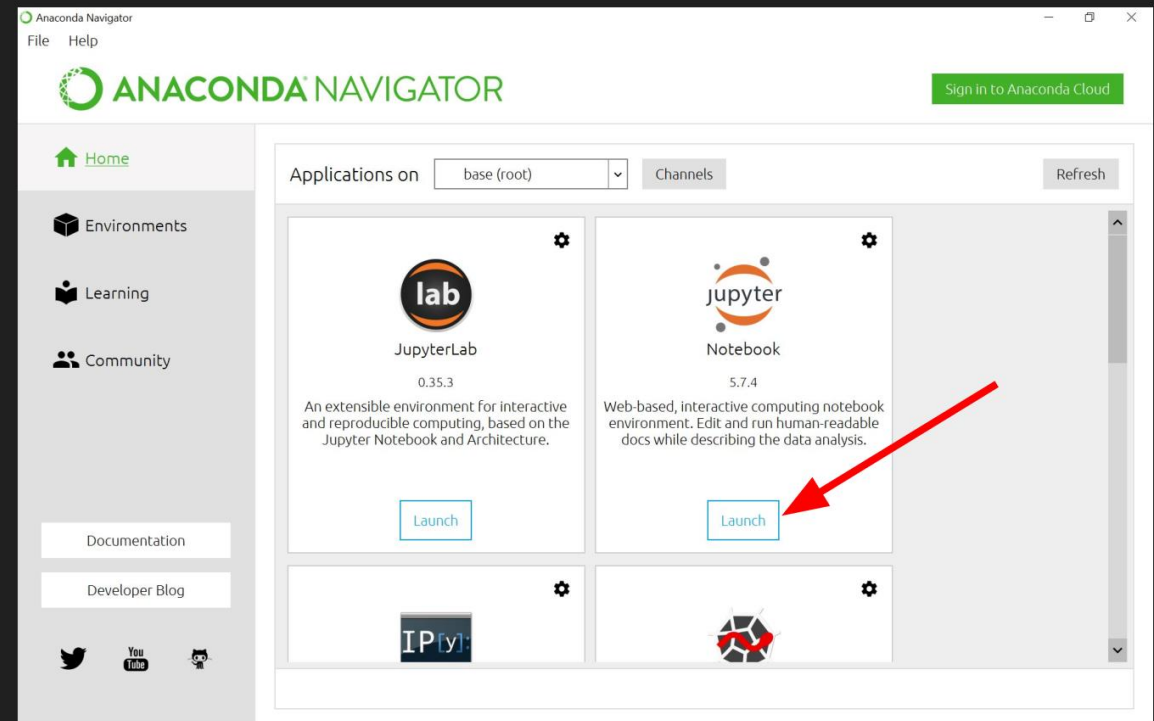
http://www.anaconda.com/download

Step 1 - Click on the Anaconda navigator Icon on your desktop
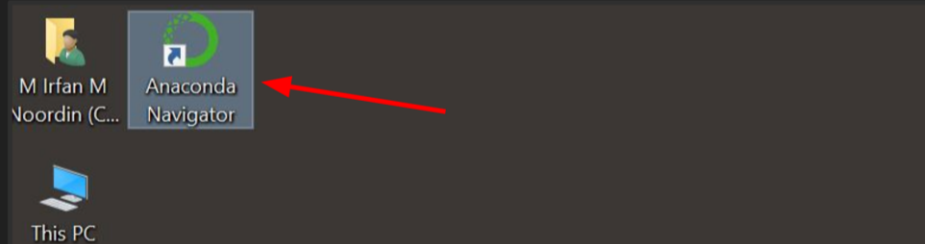
Step 2 - You will see the Anaconda loading icon
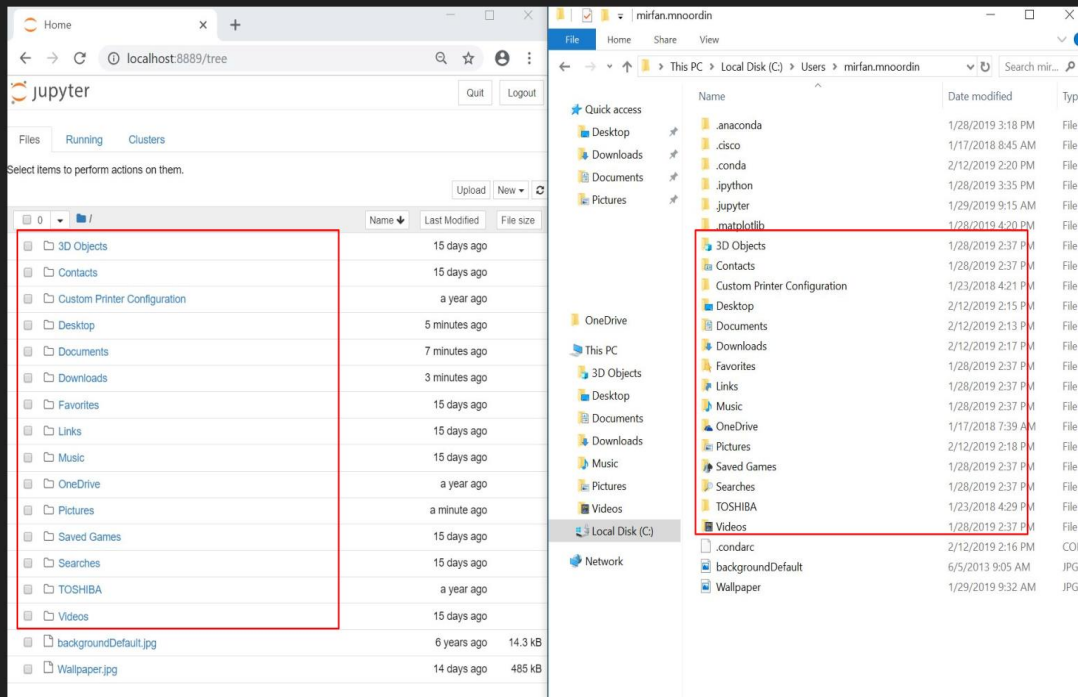Have patience young padawan (it may take a while)
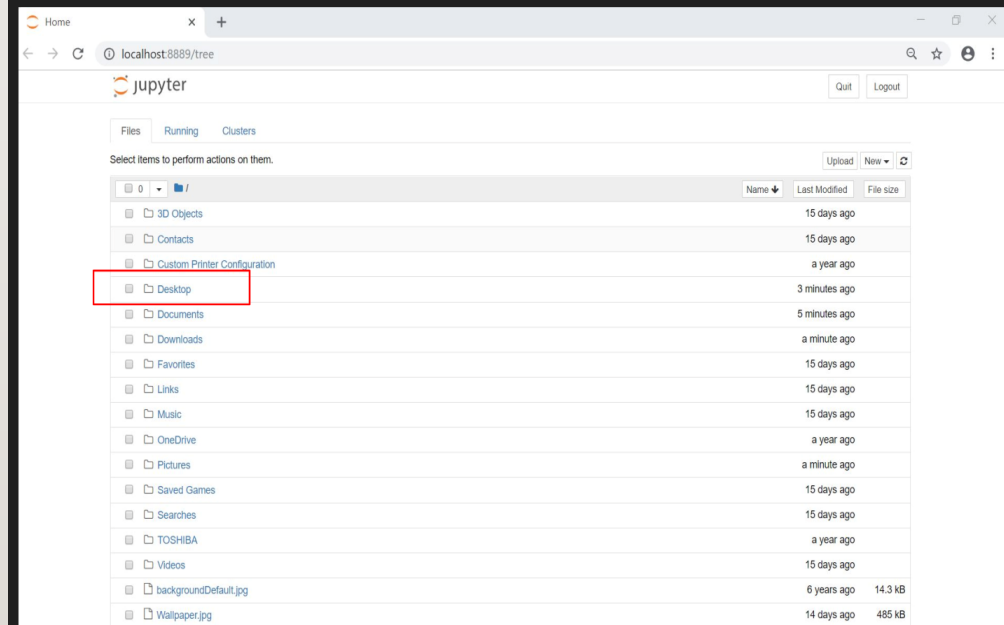
Step 3 - Open Jupyter Notebook

JUPYTER NOTEBOOK SETUP

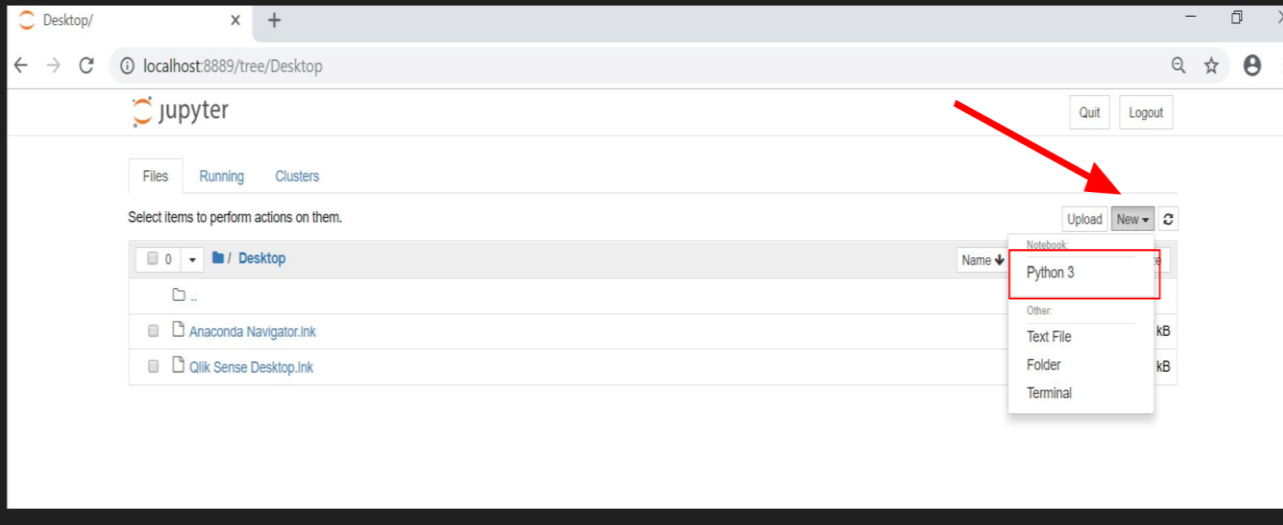Step 4 - Jupyter will show a directory which will be the same as your user's directory

Step 5 - Choose a folder to save your file in. I will choose Desktop
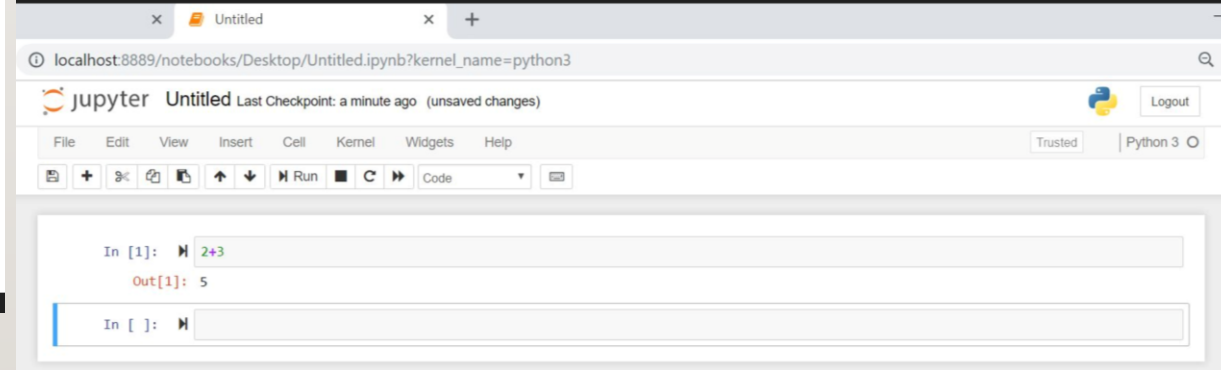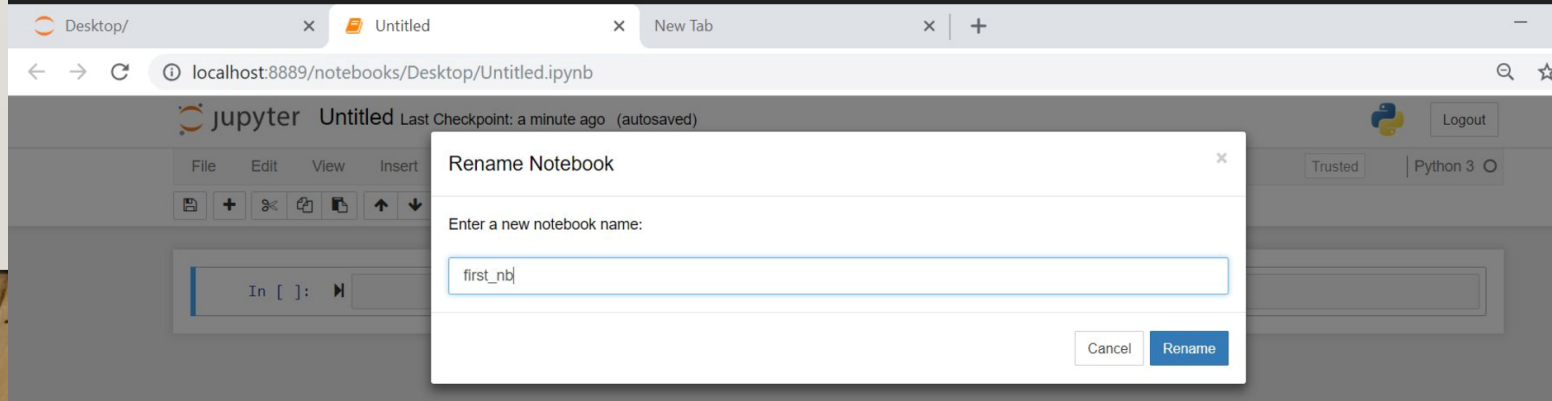
JUPYTER NOTEBOOK SETUP

JUPYTER NOTEBOOK SETUP

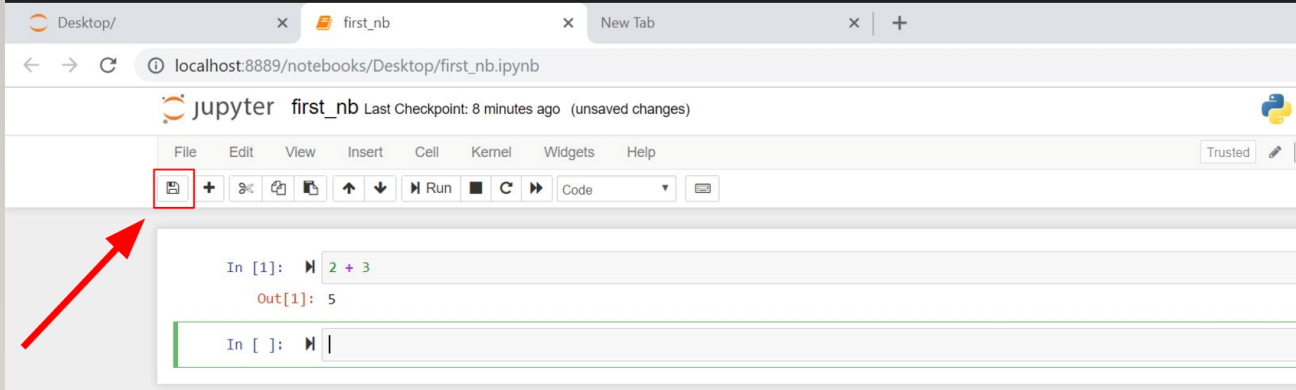## Step 8 - Rename Jupyter Notebook
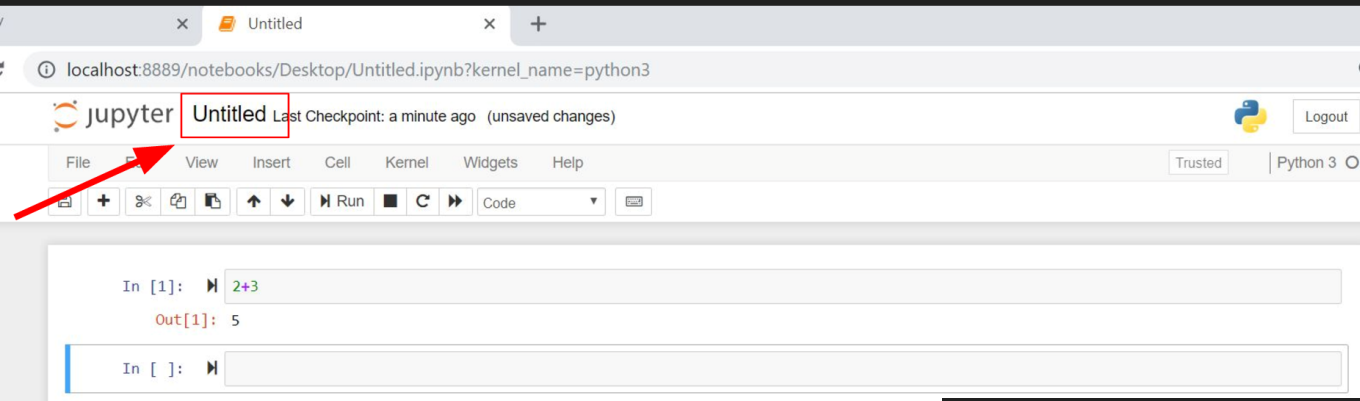
Step 8 - Rename and Save Jupyter Notebook

JUPYTER NOTEBOOK SETUP

Step 9 - Save Jupyter Notebook

You can view the first_nb.ipynb file at your desktop now

JUPYTER NOTEBOOK SETUP

- Try to produce your first python program
  - ➢print("Hello Python World!")

PYTHON

VARIABLES & OBJECTS

DATA TYPES

- Int
  - Real integers
  - Eg. 5, 7, 1000

- Float
  - Foating-point number (denoted using a decimal point)
  - Eg. 3.14
  - If 4.0 is a float, even though mathematically it is an integer

- Bool
  - Boolean values (True or False)
  - Python is case-sensitive. True is not the same as true

- str
  - A string is a sequence of characters enclosed within single or double quotes
  - You can use the len(x) function to get the number of characters of any string x

PRINTING AND TAKING INPUT

- print(x)
  - ➤x : what do you want to print
  - ➤What it does: shows it on the screen
  - ➤Result: nothing
- input(s)
  - ➤s : what do you want to show on the screen before waiting for input
  - ➤What it does: print s onto the screen and waits for user to type something
  - ➤Result: a string of what the user typed

VARIABLES

- Storing data inside "containers" or variables
- We want to save values while coding , to do more complex calculations/tasks
- Data type is inferred
- Define a specific, case-sensitive variable name

Try:

message = "Hello Python World!"

print(message)

Print(message)

STRINGS

- String is a series of characters
  - "This is a string"
  - 'This is also a string'
- Gives flexible in writing related to quotes
  - 'I told my mom, "I love Python!" '
  - "My brother's favorite language programming is Python"

- Changing case in a string with methods

Try:

name = "ada lovelace"

print(name.title())

print(name.upper())

print(name.lower())

STRINGS

- Combining/concatenating strings

Try:

first_name = "ada"

last_name = "lovelace"

full_name = first_name + " " + last_name
print(full_name)

print("Hello, " + full_name.title() + "!")

- Tabs or newline the string
  - Use \t for tab or \n for newline

- Strip whitespace
  - variable.rstrip() – remove whitespace on the right
  - variable.lstrip() – remove whitespace on the left
  - variable.rstrip() – remove whitespace