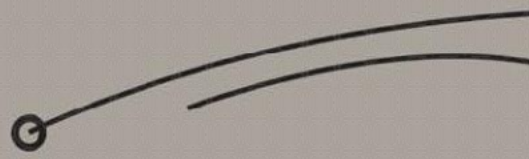
UNIVERSITY OF TWENTE.

Programming Basics

Planning Sustainable Cities (2021-1B)

Jon Wang



ITC FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

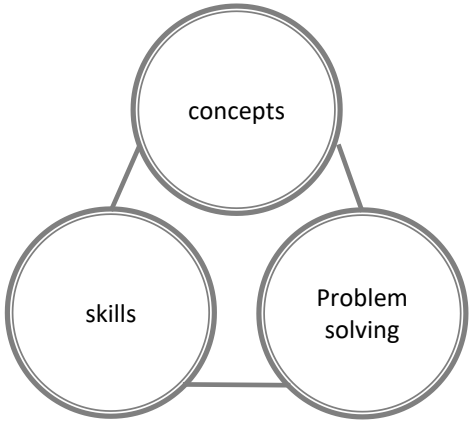
About this lecture

This lecture IS NOT about:

- Following *PowerPoint* slides
- Every aspect of programming
- Learning programming language

This lecture IS about

- Practice: ***fast pace !***
- Basic concepts/features about programming
- Skills: recipe/instructions for computers
- Think like a computer scientist for problem solving



How many times…

You encounter:

- Choosing and switching back and forth among different GIS and RS software products, and repeat…
- Knowing more details about algorithms without success…
- Problems with sharing workflows and methodologies…

How many times…

You expect:

To have a streamlined, fit-for-purpose and automated tool that can be reused by you and your colleagues.

The WORST way to learn programming

- Try to learn too many things at once
- Read too much without practice

First program!

First program

This is a short program which would help you to look at vegetation cover around Enschede, The Netherlands.

The program will load Landsat 8 imagery data to derive the Normalized Difference Vegetation Index for you.

Could you please try to run each of sections, and interpret what is going on?

In [11]:
```
# Import necessary modules

import matplotlib.pyplot as plt
import numpy as np
#% pip install geopandas
import geopandas as gpd
#% pip install rasterio
import rasterio  # For more, please refer to "https://rasterio.readthedocs.io/en/latest/topics/reading.html"
from rasterio.plot import show
from sklearn import cluster
```

In [ ]:
```
# Before reading the data we need to first clone the data on Github to our Colab workspace
!git clone https://github.com/jonwangio/Programming-Basics
```

In [12]:
```
# Read files

b5_file_location = 'Programming-Basics/data/20200521_B5.TIF'  #'uu_ml/data/b5_2015.TIF'
b5 = rasterio.open(b5_file_location, nodata=0)

b4_file_location = 'Programming-Basics/data/20200521_B4.TIF'
b4 = rasterio.open(b4_file_location, nodata=0)
```

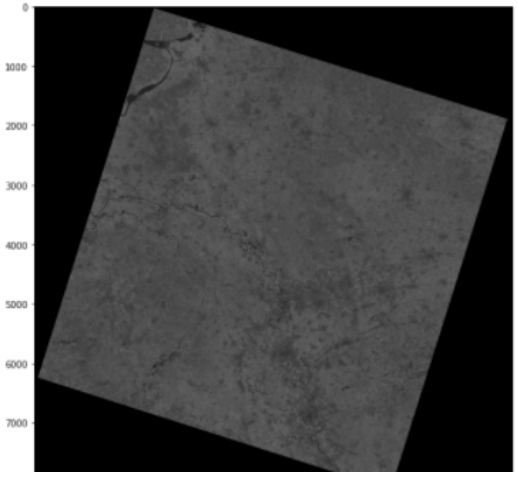
In [79]:
```
# File structure and metadata

# type(b5)
# b5.name
print('The coordinate system of the file is: ' + str(b5.crs))
print('The number of band in the file is: ' + str(b5.count))
print('The spatial extent is bounded by: ' + str(b5.bounds))
```

```
The coordinate system of the file is: EPSG:32632
The number of band in the file is: 1
The spatial extent is bounded by: BoundingBox(left=217485.0, bottom=5606685.0, right=459615.0, top=5851515.0)
```

In [80]:
```
# Layer extraction and visualization

b5_layers = b5.read(1)
b4_layers = b4.read(1)

print('The size of b5 imagery data is ' + str(b5_layers.shape))

# Visualize input files
%matplotlib inline

plt.figure(figsize = (10,10))
plt.imshow(b5_layers, cmap='gray')
```

```
The size of b5 imagery data is (8161, 8071)
<matplotlib.image.AxesImage at 0x13031557ac8>
```

Out[80]:

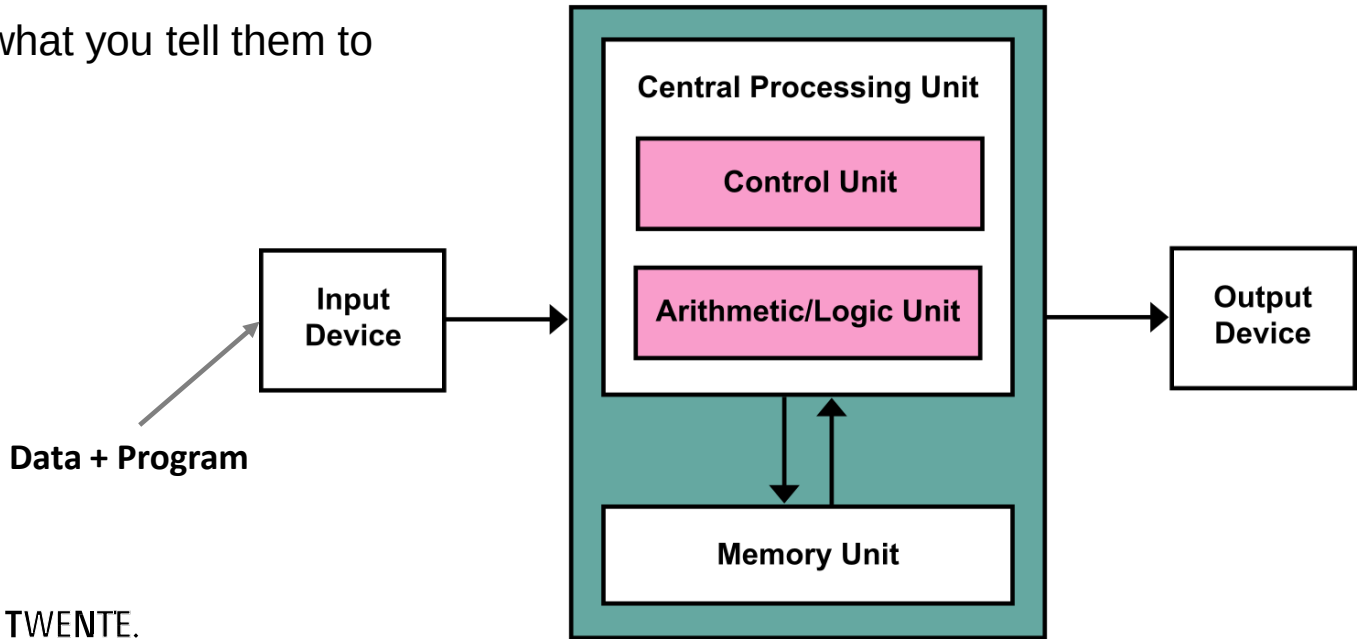


ITC

UNIVERSITY OF TWENTE.

What do programs/computers do?

- Follow **instructions** to perform **calculations**
- Remember results and make **storage**
- Leverage **built-in** functionalities
- Accomplish **functionalities defined** by you
- Only do what you tell them to



Source: https://en.wikipedia.org/wiki/Von_Neumann_architecture

So, what is in programs?

- **Objects, or data objects:** variables with data types, normally as inputs
- **Operators:** convert, sum, or other types of processing of data objects
- **Expressions:** combine objects and operators, in formal language
- **Statements:** series of expressions
- **Recipe:** Steps, flow control, stop, related to iterations

Debugging

- **Very normal it would cost over 50% of your programming time**
- Mainly because programs strictly follow syntax

Other IDE and modules

Recommended IDE in:

- Anaconda

Commonly used modules:

- numpy
- GDAL
- Geo Pandas
- Rasterio
- PIL
- Pickle
- …