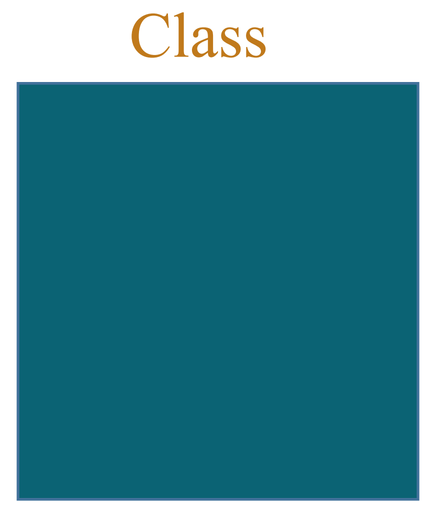
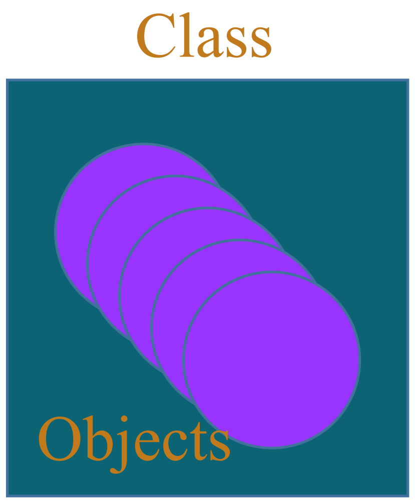
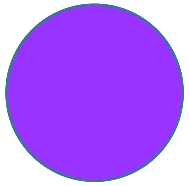
CMPUT 175 - Lab 02
Fall 2023

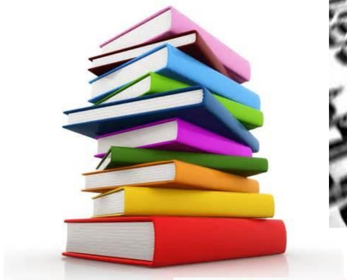
Classes in Python





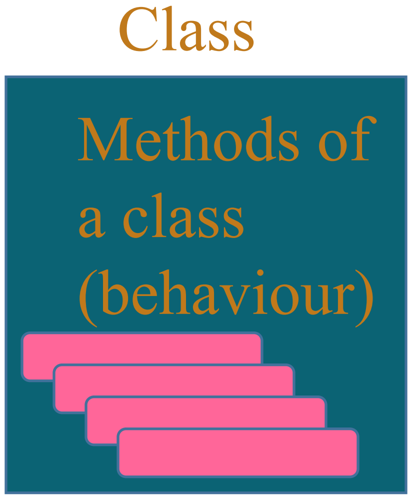
- You can define a new Abstract Data Type (ADT) by defining a new class in Python
- A class can have many instances
- For example you can have a class for books, a class for cars, a class of students, etc.

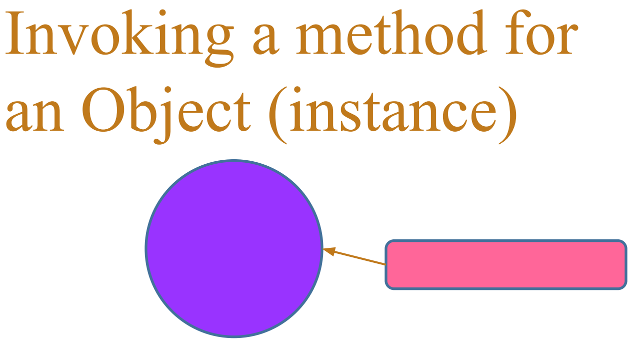




Classes in Python

- A class can have methods that describe the expected behaviour of instances of the class
- You create a new instance for a class myClass by assigning anInstance=myClass(*arguments*)
- You can invoke a method for an instance by calling anInstance.myMethod(*arguments*)

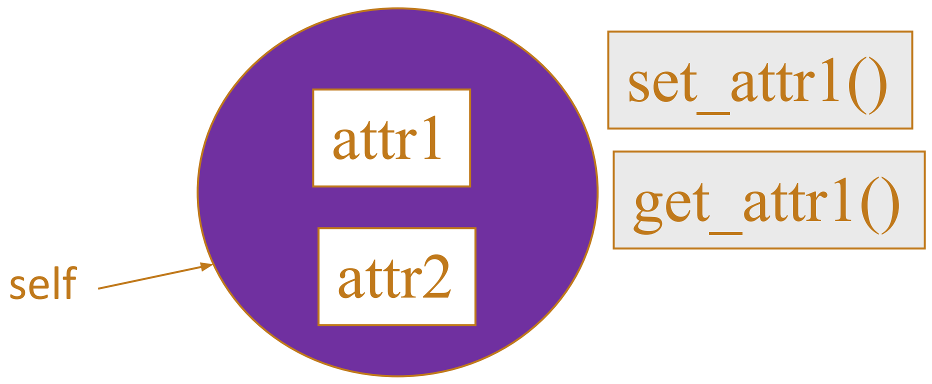




Classes in Python

- Instances of a class have attributes. All instances of the same class have the same attributes but possibly with different values.
- For example all cars have a colour, a model, a horsepower value, etc.
- Methods are functions or procedures defined in the class by def menthodName (*arguments*)
- self is always the first arguments when defining methods. It is a reference that is bound to the instance.
- The constructor is called __init__() which creates a new instance and initializes its attributes

General Example



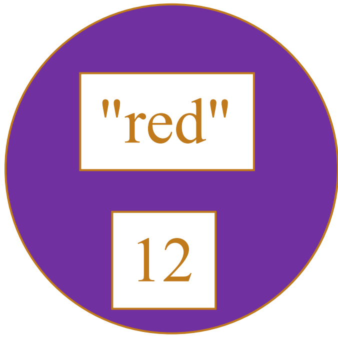
```
class class_name :
    def __init__ (self , init_attr1 , init_attr2 ):
        self . attr1 = init_attr1
        self .attr2 = init_attr2
    def get_attr1 ( self ):
        return self .attr1
    def set_attr1 (self , new_attr1 ):
        self .attr1 = new_attr1
```

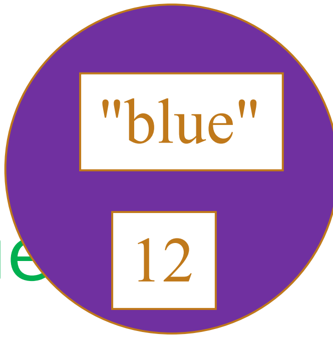
General Example, cont’d

```
def main ():
    # creating an instance of the class
    oneInstance = class_name ("red ", 12)

    # invoking get_attr1() method of the
instance
    print ( oneInstance . get_attr1 ()) # red

    oneInstance . set_attr1 (" blue ")
    print ( oneInstance . get_attr1 ()) # blue
```





Lab 2 Guidelines:

- Useful functions: chr(), ord()

- Caesar cipher:
  - **Encryption:** key =5, letter= a

    a,b,c,d,e,f,g,........u,v,w,x,y,z

  - **Decryption:** key =5, letter= a

    a,b,c,d,e,f,g,........u,v,w,x,y,z

- what if key=31?

Lab 2 Guidelines:

- Create a file called **lab2.py**
- Need to define two functions:
  - def **getInputFile():**
    - ask the user for input file name
    - validate the .txt extension
    - if valid return the filename
  - def **decrypt(filename):**
    - read data from file and clean it
    - decrypt given text with key
    - display decrypted words using single space.

Lab 2 Guidelines:

- Do not forget to add docstring in functions;
- Example:
  - def getInputFile():

    “”””

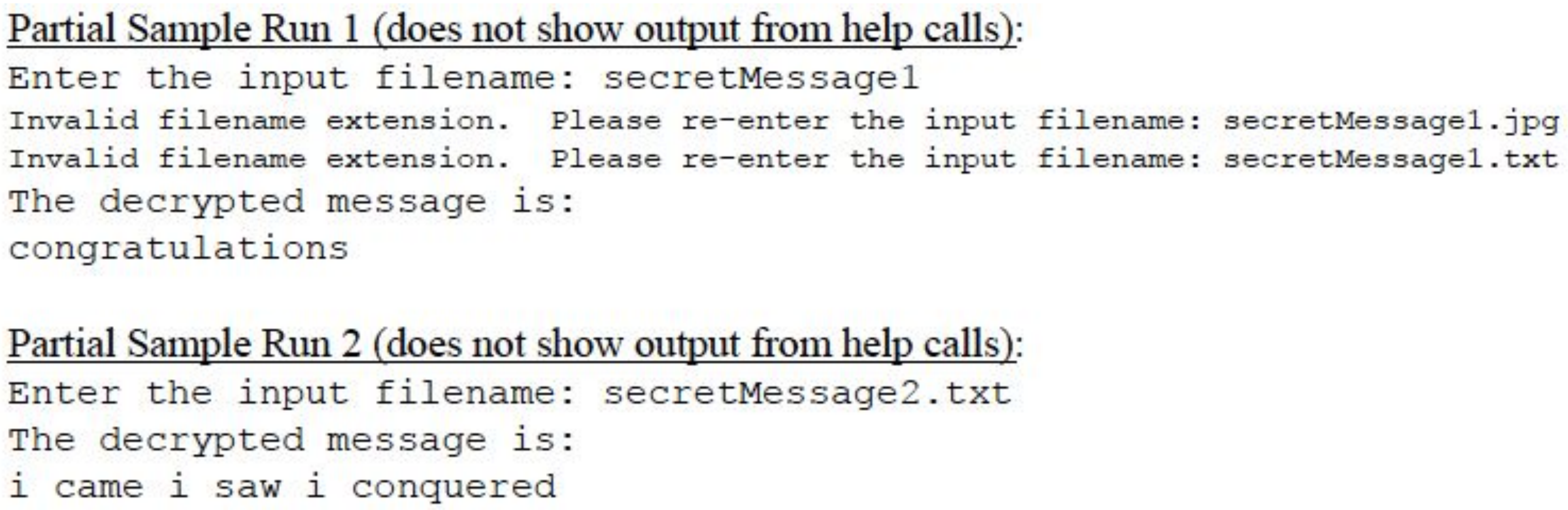
    Brief description about the function and parameters

    “”””

    #body
  - help(getInputFile)

Lab 2 Guidelines:

- Sample Input/Output:



<u>Partial Sample Run 1 (does not show output from help calls):</u>

```
Enter the input filename: secretMessage1
Invalid filename extension.  Please re-enter the input filename: secretMessage1.jpg
Invalid filename extension.  Please re-enter the input filename: secretMessage1.txt
The decrypted message is:
congratulations
```

<u>Partial Sample Run 2 (does not show output from help calls):</u>

```
Enter the input filename: secretMessage2.txt
The decrypted message is:
i came i saw i conquered
```