Classes And Objects

# Before Class

1. Familiarise yourself with object-oriented programming concept. Then watch the video:

<https://www.youtube.com/watch?v=SS-9y0H3Si8&list=PLck2iFB-00xnYlmq1V4CSN9I4JdSCiojg&index=7&t=0s>

1. Familiarise yourself with tutorials available on w3schools which deals with class creation and their components (fields and methods).

<https://www.w3schools.com/python/python_classes.asp>

1. How Python classes are defined? Watch the video:

<https://youtube.com/playlist?list=PLi01XoE8jYohWFPpC17Z-wWhPOSuh8Er->

1. Familiarise yourself with class, field and method definition on Python tutorial point 9.3

<https://docs.python.org/3/tutorial/index.html>

# During Class

1. Identify at least 3 states and 3 behaviors for each of the following object.
   1. Book
   2. Telephone connection
   3. A group of students

Then, for the listed states and behaviors, create corresponding fields (attributes) and methods. Remember to use verbs in method names because they describe activities.

1. Write a program in which create a University class, consisting of one field containing the name of the university and one method to display this name.

class University():  
  
 # object constructor (\_\_init\_\_ method)  
 def \_\_init\_\_(self):  
 # object states/attributes (fields)  
 self.name = 'CUE'   
  
 # object bahaviors (methods)  
 def print\_name(self):   
 print(self.name)

Then create a University class object and call the method to display the name of the university.

1. Add a set\_name() method in the University class that allows you to rename the university (change the field value).

def set\_name(self, name):  
 self.name = name

Then modify the program to change the name of the university in the created object. Enter MIT which stands for Massachusetts Institute of Technology, and display this changed name.

1. Add a fullname field to the University class that contains the full name of the university, and the print\_fullname() and set\_fullname() methods to display and change the full name of the university. Then create an object and:
   1. display short university name
   2. display full university name
   3. change university name
   4. change full university name
   5. display short university name
   6. display full university name
2. Write a program in which you create a TV class that describes a TV set. The class should contain one boolean field called 'is\_on' that specifies whether the TV set is turned on. By default, the TV is turned off. Add turn\_on() and turn\_off() methods in the class to turn the TV on and off, respectively. Also add a show\_status() method to display whether the TV is on or off. Sample message:

TV is on

Then try using the TV set in the program:

* 1. Create TV set
  2. Show TV status
  3. Turn TV on
  4. Show TV status
  5. Turn TV off
  6. Show TV status

1. In the TV class, add the channel\_no field indicating the number of the TV channel displayed by the TV set. Initially, the TV is set to channel 1. Modify the show\_status() method so that it also displays the TV channel number, but only if the TV is turned on:

TV is on, channel 1

Then try using the TV set.

1. Add the set\_channel(new\_channel\_no) method in the TV class to set the TV channel number. Then try using the TV set.
   1. Create a TV set
   2. Show TV status
   3. Turn TV on
   4. Show TV status
   5. Change TV channel to 5
   6. Show TV status
   7. Turn TV off
   8. Show TV status
2. In the TV class, add the channels field containing the list of available TV channel names (array). Initially, the array should be empty (TV not programmed, no available channels). Add set\_channels(channels\_list) and show\_channels() methods in the TV class, which allows you to set channels on the TV and display the list of available channels. Sample list of available channels:

Channel list:  
1. TVP1  
2. TVP2  
3. Polsat  
4. TVN  
5. Filmbox  
6. Discovery

Then try using the TV set:

* 1. Create a TV set
  2. Show TV status
  3. Turn TV on
  4. Show TV status
  5. Display the list of available channels
  6. Set TV channels: TVP1, TVP2, Polsat, TVN, Filmbox, Discovery
  7. Display the list of available channels
  8. Show TV status
  9. Turn TV offs

1. In the TV class, make changes to the show\_status() method so that it displays not only the selected channel number but also its name. When the selected channel number exceeds the list of available channels, the channel name is not displayed.

TV is on, channel 4 (TVN)

Then try using the TV. Set at least 7 channels (seven TV stations), change the channel numbers and display TV status every time.

# After Class

1. In the TV class, add support for volume adjustment in the range 0 to 10. The initial value of the volume level is 0. Add two methods to increase and decrease the TV volume level by one. Note that you cannot increase or decrease the volume beyond the specified range. Display the current volume level in the show\_status() method. Then check the operation of the TV by adjusting and displaying its volume level.
2. E-book is a digital book that can be read using a computer or other electronic devices (electronic book readers). Write a program in which define a class that describes states and behaviors of an e-book. Each book has a title, author, number of pages and the current page number that is currently being read. It is possible to open a book - then we can read it. If a book is open, it is possible to go to the next or previous page.

Place the class describing e-books in a separate file/module. In the main program file, try using the e-book:

* 1. Create a book with a title, author, number of pages (check how to set the initial values of the fields at the time of creating the object using the \_\_init\_\_ method / constructor and passing the initial values as arguments to the method call)
  2. Open a book
  3. Display a book status (title, author, page numbers, current page no)
  4. Read a few pages
  5. Display a book status
  6. Close a book
  7. Read a few pages (it should not be possible to perform this operation - display the message that the book is closed).

1. The medical thermometer measures the patient's temperature in the range from 34.0 to 42.0 degrees Celsius, with an accuracy of 0.1 degrees. Write a program in which define a class that describes the states and behaviors of the thermometer. The thermometer should enable temperature measurement (do it by generating a random number from the 34.0 to 42.0 range) and display the measured value. If the temperature is at least 37 degrees Celsius, the thermometer should additionally display the 'Fever' message, e.g.

Temperature: 37.2C (fever)

When the temperature is at least 41.0, the thermometer should additionally display the message 'CRITICAL TEMPERATURE!!'. Place the class definition and the main program in separate files. Then use the program and:

* 1. Create a thermometer
  2. Turn thermometer on
  3. Measure temperature
  4. Display temperature
  5. Turn thermometer off

1. The bank account has a 26-digit number assigned when creating an account. The initial account balance is PLN 0. You can deposit any amount on the account. You can also withdraw any amount from the account, provided that it does not exceed the account balance. If you try to withdraw a larger amount, the following message will be displayed: "Insufficient funds on the account". At any time, it is possible to display information about the number and balance of the bank account in the following format:

Bank Account No: 11 1111 1111 1111 1111 1111 1111  
Balance: PLN 25,38

Create a program that handles the bank account.

* 1. Familiarises yourself with a problem.
  2. Identify an object
  3. Define the states and behaviors of the object.
  4. Transform the states and behaviors of the object into the fields and methods of the class that will serve as a pattern for creating an object.
  5. Create a sketch of the class without creating any method content.
  6. Create the content of each method.

Then use the program and:

* 1. Create a bank account with the number 12 3456 5555 9090 1111 0000 7722
  2. Display account balance
  3. Deposit PLN 25,30
  4. Display account balance
  5. Withdraw PLN 31,70
  6. Display account balance
  7. Withdraw PLN 14
  8. Display account balance

1. Write a program containing a Statistics class that describes the properties of any set of numbers. The class should allow to:
   1. Add to the set of numbers, the next number read from the keyboard (store the numbers in the array)
   2. Display all numbers separated by a space
   3. Determine the greatest number
   4. Determine the smallest number
   5. Calculate the arithmetic mean of numbers
   6. Calculate of the median
   7. Display of calculated / determined statistical quantities (minimum, maximum, arithmetic mean, median)

Then use the program for numbers:

12, 37, 6, 9, 17

1. The Contact class contains the 'name', 'email' and 'telephone' fields enabling the description of a single contact on a smartphone. The Contact\_List class allows you to store contacts (store objects describing contacts in an array) and perform the following operations:
   1. Adds a new contact
   2. Displays the contact list

Write a program consisting of 3 files (smartphone.py, contact.py, contact\_list.py). In the mail program (smartphone.py) create an object representing a contact list and add the following people data:

John Brown brown@onet.pl 555234000  
Anna May am@o2.pl 232000199  
George Small smallg@google.pl 222999100  
Paola Big bigpaola@poczta.pl 100200300

Then display the contact list available on the smartphone.