香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

INTRODUCTION TO COMPUTER SCIENCE: PROGRAMMING METHODOLOGY

TUTORIAL 8

OBJECT ORIENTED PROGRAMMING

- Download this slides and codes at:

The tutorial materials can be downloaded on:

https://cuhko365-my.sharepoint.com/:f:/g/personal/220019030_link_cuhk_edu_cn/EnHq0qwnvi1Jg6tSZeqwkGUBvpAZnqrKYSDOO_JlHNnyvw?e=fLKmbU

Here is the information of TAs' office hours.

ZP Zibin Pan (SSE, 220019030) > **2310 - CSC1001 files**

名称	修改时间	修改者	文件大小	共享	活动
T01_T02_T03	9月8日	Zibin Pan (SSE, 220019030	5 个项目	已共享	
T04_T05_T06	9月8日	Zibin Pan (SSE, 220019030	5 个项目	已共享	
T07_T08_T09	9月8日	Zibin Pan (SSE, 220019030	2 个项目	已共享	
T10_T11_T12	9月8日	Zibin Pan (SSE, 220019030	4 个项目	已共享	
T13_T14_T15	9月8日	Zibin Pan (SSE, 220019030	3 个项目	已共享	
T16_T17_T18	9月8日	Zibin Pan (SSE, 220019030	3 个项目	已共享	
T19_T20_T21	9月8日	Zibin Pan (SSE, 220019030	5 个项目	已共享	

How to use .ipynb file?

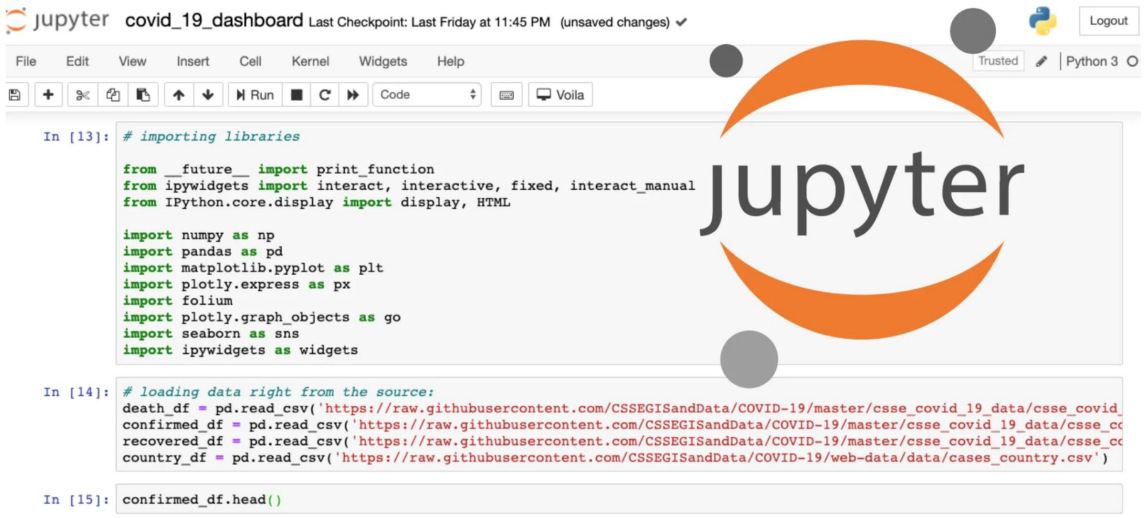
Online: Google Colab

colab.google
https://colab.google

Google Colab

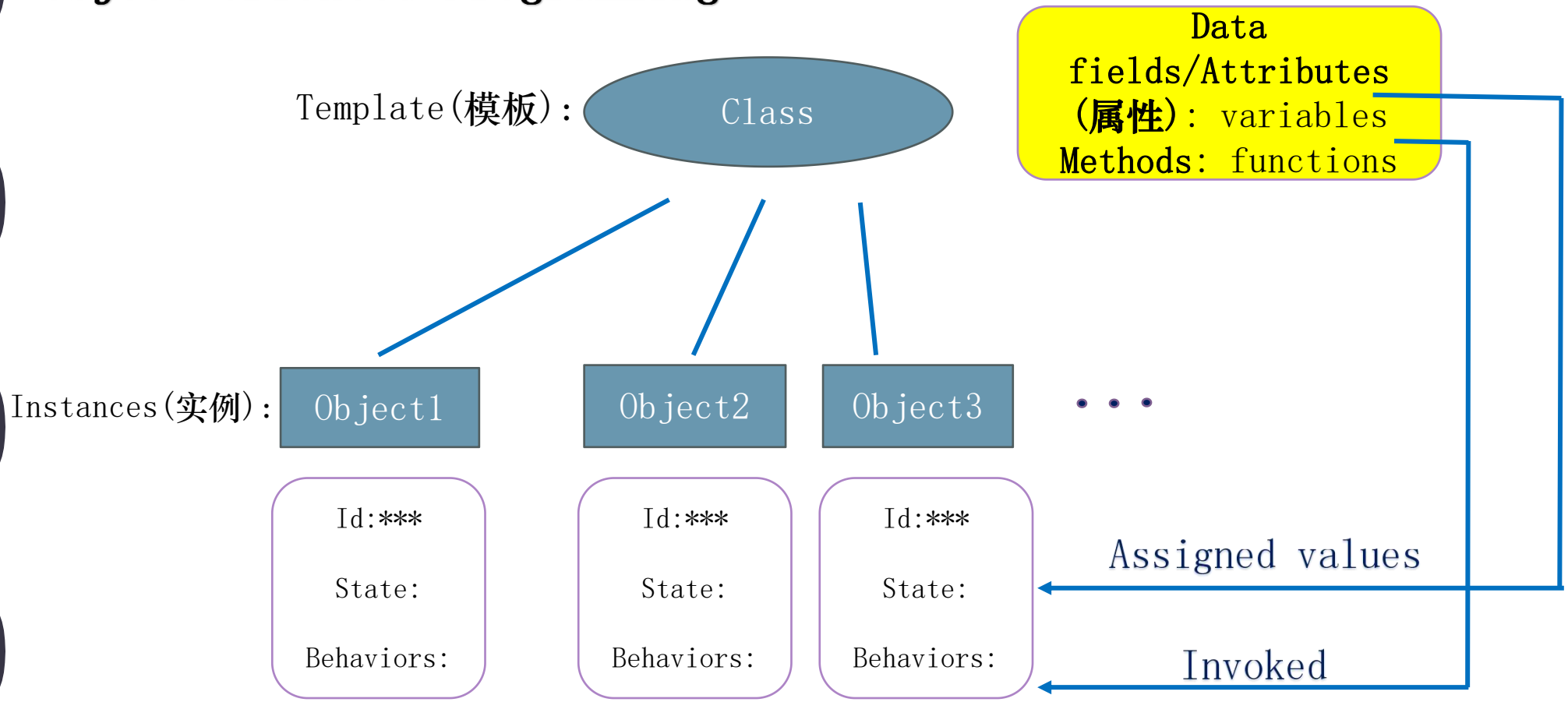
Google Colaboratory. Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs ...

Offline: Jupyter Notebook(in Anaconda3)



Class and Object

Object Oriented Programming



Class Definition

A default value can be set.

```
class className:
```

initializer

```
def __init__(self,parameter1=1,parameter2=1,...):
    self.parameterName1=parameter1
    self.parameterName2=parameter2
```

methods

```
def methodName1(self,...):
    statements
def methodName2(self,...):
    statements
```

(“.” operator here connects object with its attributes, indicating the affiliation(从属关系) between them.)

```
objectName=className()

#Constructor:
#①Create an object in the memory
#②__init__() method will be automatically invoked
```

- *①A parameter(named “self” here) must be given to __init__() method to refer to the object you create by this template.*
- *②Any methods you define in this class should have the parameter above(“self”) in order to invoke them using “.” operator to connect object and its method, i.e. when invoking you pass the object as a parameter to the method.*

Private Data Fields/Methods

- In Python, the private data fields are defined with two leading underscores. You can also define a private method named with two leading underscores
- Private data fields and methods can be accessed within a class, but they cannot be accessed outside the class

- ***Why we need private data fields and methods, which are not convenient to modify?***

- **Tip: If a class is designed for other programs to use, to prevent data from being tampered with(e.g. bank account, score etc.) and to make the class easy to maintain, define data fields as private. If a class is only used internally by your own program, there is no need to hide the data fields.**

- **For others to use your programs, you need to provide a way for the users to get access to(by "getter") and modify(but not directly and arbitrarily, by "setter") the private data fields.**
- **Getter:**

```
def getPropertyName(self):
```

- **Setter:**

```
def setPropertyName(self,propertyValue):
```

Mutable and Immutable Objects

- In Python, some objects are mutable(e.g. lists), others are immutable(e.g. strings, numbers, tuples).
- If you pass some objects to a function where you may do some modifications to the parameters, then after invoking it, mutable objects will be changed while immutable ones won't, compared with before.

Circle.py

```
from math import pi

class Circle:
    def __init__(self,radius=1):
        self.radius=radius

    def getPerimeter(self):
        return 2*self.radius*pi

    def getArea(self):
        return self.radius*self.radius*pi

    def setRadius(self,radius):
        self.radius=radius
```

```
from Circle import Circle

def printAreas(c,times):
    while times>=1:
        c.radius=c.radius+1
        times=times-1
        print(times,c.radius,c.getArea())

def main():
    myCircle=Circle()
    n=5
    printAreas(myCircle,n)
    print(myCircle.radius)
    print(n)

main()
```

```
Before invoking the function:
Radius of myCircle is 1
n is 5
-----------------------------
n           Radius      Area
4           2           12.57
3           3           28.27
2           4           50.27
1           5           78.54
0           6           113.10
-----------------------------
After invoking the function:
Radius of myCircle is 6
n is 5
```

(Output of an improved version.)

Q1: Private data fields

- i) What problems arise in running the program (a)? How to fix it?
- ii) Is the program (b) correct? If so, what will be the output?

```
class A:
    def __init__(self):
        self.__i=i

def main():
    a=A(5)
    print(a.__i)

main()
```

(a)

```
class A:
    def __init__(self,newS='Welcome'):
        self.__s=newS

    def myprint(self):
        print(self.__s)

def main():
    a=A()
    a.myprint()

main()
```

(b)

Q2: Mutable and immutable objects

- Show the output of the following programs (a) and (b).

```
class Count:
    def __init__(self,count=0):
        self.count=count

def increment(c,times):
    c.count+=1
    times+=1

def main():
    c=Count()
    times=0
    for i in range(100):
        increment(c,times)
    print("count is",c.count)
    print("times is",times)

main()
```

(a)

```
class Count:
    def __init__(self,count=0):
        self.count=count

def m(c,n):
    c=Count(5)
    n=3

def main():
    c=Count()
    n=1
    m(c,n)

    print('count is',c.count)
    print('n is',n)

main()
```

(b)

Q3: Account class

Design a class named Account that contains:

- A private int data field named ID for the account.
- A private float data field named balance for the account.
- A private float data field named annualInterestRate that stores the current interest rate.
- A constructor that creates an account with the specified ID(default 0), initial balance(default 100), and annual interest rate(default 0).
- The accessor and mutator methods for ID,balance, and annualInterestRate.
- A method named getMonthlyInterestRate() that returns the monthly interest rate.
- A method named getMonthlyInterest() that returns the monthly interest.
- A method named withdraw that withdraws a specified amount from the account.
- A method named deposit that deposits a specified amount to the account.
- Use this formula to calculate the monthly interest: $MonthlyInterest = balance \times monthlyInterestRate$, where $monthlyInterestRate = annualInterestRate/12$.
- Write a test program that creates an Account object with an account ID of 1122, a balance of $20000, and an annual interest rate of 4.5%. Use the withdraw method to withdraw $2500, use the deposit method to deposit $3000, and print the ID, balance, monthly interest rate, and monthly interest.

Q4: ATM machine

- Use the **Account** class created in the previous practice to simulate an ATM machine. Create ten accounts in a list with the ids 0,1,..,9, and an initial balance of $100. The system prompts the user to enter an id. If the id is entered incorrectly, ask the user to enter a correct id. Once an id is accepted, the main menu is displayed as shown in the sample run next page. You can enter a choice of 1 for viewing the current balance, 2 for withdrawing money, 3 for depositing money, and 4 for exiting the main menu. Once you exit, the system will prompt for an id again. So, once the system starts, it won't stop.

Q4: ATM machine(Sample run)

```
Enter your ID:5
--------------------
Main menu
1:check balance
2:withdraw
3:deposit
4:exit
--------------------
Enter a choice:1
The balance is 100
--------------------
Main menu
1:check balance
2:withdraw
3:deposit
4:exit
--------------------
Enter a choice:3
Enter an amount to deposit:1000
--------------------
Main menu
1:check balance
2:withdraw
3:deposit
4:exit
--------------------
Enter a choice:1
The balance is 1100
```

```
--------------------
Main menu
1:check balance
2:withdraw
3:deposit
4:exit
--------------------
Enter a choice:4
Good Bye!

Enter your ID:5
--------------------
Main menu
1:check balance
2:withdraw
3:deposit
4:exit
--------------------
Enter a choice:1
The balance is 1100
--------------------
Main menu
1:check balance
2:withdraw
3:deposit
4:exit
--------------------
Enter a choice:2
Enter an amount to withdraw:500
```

```
--------------------
Main menu
1:check balance
2:withdraw
3:deposit
4:exit
--------------------
Enter a choice:1
The balance is 600
--------------------
Main menu
1:check balance
2:withdraw
3:deposit
4:exit
--------------------
Enter a choice:4
Good Bye!

Enter your ID:
```

(For next user to input.)