

# Python 程式設計作業

範圍： 類別、模組的應用

銘傳大學電腦與通訊工程系

班 級	電通四甲
姓 名	趙啟翔
學 號	05050130
作業成果	應繳作業共 <u>7</u> 題，前 3 題每題 20 分，後 4 題每題 10 分，滿分為 100 分 我共完成 <u>7</u> 題，應得 <u>100</u> 分
授課教師	陳慶逸

■ 請確實填寫自己寫完成題數，並且計算得分。填寫不實者(如上傳與作業明顯無關的答案，或是計算題數有誤者)，本次作業先扣 50 分。

■ 確實填妥封面的內容，完成後請上傳 pdf 檔。

EX 1: 試寫出一個命名為 `strSplitClass` 的 Python 類別，該類別裡有一個 `StrSplit` 方法(methods)可將輸入的字串進行切割，得到以字(word)組成的串列。

例如：

```
str1 = strSplitClass()
str1.strSplit()

machine learning .ipynb

Out[42]: ['machine', 'learning', '.ipynb']
```

圖 5-16 輸入與輸出結果

```
class strSplitClass:
    def __init__(self):
        self.str = ""
        print(self.str)
    def strSplit(self):
        self.str=input()
        self.str=self.str.split(" ")
        print(self.str)
str1=strSplitClass()
str1.strSplit()
```

```
In [3]: 1 class strSplitClass:
2         def __init__(self):
3             self.str = ""
4             print(self.str)
5         def strSplit(self):
6             self.str=input()
7             self.str=self.str.split(" ")
8             print(self.str)
9 str1=strSplitClass()
10 str1.strSplit()
```

```
machine learning .ipynb
['machine', 'learning', '.ipynb']
```

EX 2: 試寫出一個命名為 cartesian 的 Python 類別，該類別的輸入參數為 x 和 y 兩個數值(為平面座標中的一個點(x,y))，而類別裡的方法(distanceToOrigin)可以計算出(x,y)這個點離原點的距離。

例如：

```
a = cartesian(3,3)
a.distanceToOrigin()

4.242640687119285
```

圖 5-167 輸入與輸出結果

```
class cartesian:
    def __init__(self, x, y):
        self.x=x
        self.y=y
    def distanceToOrigin(self):
        print(((self.x-0)**2+(self.y-0)**2)**(1/2))
a=cartesian(3,3)
a.distanceToOrigin()
```

```
In [5]: 1 class cartesian:
2         def __init__(self, x, y):
3             self.x=x
4             self.y=y
5         def distanceToOrigin(self):
6             print(((self.x-0)**2+(self.y-0)**2)**(1/2))
7 a=cartesian(3,3)
8 a.distanceToOrigin()
```

4.242640687119285

EX 4: 試寫出一個命名為 **Shape** 的 Python 類別，該類別由計算矩形面積(area)、矩形周長(perimeter)和尺度大小改變(scaleSize)等三個方法(methods)所構成。

例如：

```
rectangle = Shape(4, 8)

print(rectangle.area()) # finding the area
print(rectangle.perimeter()) #finding the perimeter

rectangle.scaleSize(0.5) #making the rectangle 50% smaller
print(rectangle.area()) #re-printing the new area of the rectangle
print(rectangle.perimeter()) #re-printing the new perimeter of the rectangle|
```

```
32
24
8.0
12.0
```

圖 5-19 輸入與輸出結果

```

class Shape:
    def __init__(self,width,height):
        self.width=width
        self.height=height
    def area(self):
        return self.width*self.height
    def perimeter(self):
        return self.width*2+self.height*2
    def scalesize(self,size):
        self.width=self.width*size
        self.height=self.height*size
rectangle=Shape(4,8)

print(rectangle.area())
print(rectangle.perimeter())

rectangle.scalesize(0.5)
print(rectangle.area())
print(rectangle.perimeter())

```

```

In [6]: 1 class Shape:
2         def __init__(self,width,height):
3             self.width=width
4             self.height=height
5         def area(self):
6             return self.width*self.height
7         def perimeter(self):
8             return self.width*2+self.height*2
9         def scalesize(self,size):
10            self.width=self.width*size
11            self.height=self.height*size
12 rectangle=Shape(4,8)
13
14 print(rectangle.area())
15 print(rectangle.perimeter())
16
17 rectangle.scalesize(0.5)
18 print(rectangle.area())
19 print(rectangle.perimeter())

```

32  
24  
8.0  
12.0

EX 5: 試寫出一個命名為 `IOString` 的 Python 類別，該類別由兩個兩個方法(methods)

所構成，其中 `getString` 負責接收使用者所輸入的字串，而另一個方法 `printString` 則是將字串改成大寫後列印出來。

例如：

```
str1 = IOString()
str1.getString()
str1.printString()
```

```
block
BLOCK
```

圖 5-20 輸入與輸出結果

```
class IOString:
    def __init__(self):
        self.string=""
    def getString(self):
        self.string=input()
    def printString(self):
        self.string=self.string.upper()
        print(self.string)
str1=IOString()
str1.getString()
str1.printString()
```

```
In [7]: 1 class IOString:
2         def __init__(self):
3             self.string=""
4         def getString(self):
5             self.string=input()
6         def printString(self):
7             self.string=self.string.upper()
8             print(self.string)
9 str1=IOString()
10 str1.getString()
11 str1.printString()
```

```
block
BLOCK
```

EX 6: 試寫出一個命名為 Employee 的 Python 類別，該類別由兩個方法 (methods) 所構成，其中 displayEmployee 可以列印受僱員的名字和薪水，displayCount 則是輸出受僱者的總數。

例如：

```
emp1 = Employee("Zara", 2000)
emp2 = Employee("Manni", 5000)
emp3 = Employee("Amy", 5000)
emp1.displayEmployee()
emp2.displayEmployee()
emp3.displayEmployee()
print("Total Employee %d" % Employee.empCount)
```

```
Name : Zara , Salary: 2000
Name : Manni , Salary: 5000
Name : Amy , Salary: 5000
Total Employee 3
```

圖 5-21 輸入與輸出結果

```
class Employee:
    empCount=0
    def __init__(self,Name,Salary):
        self.Name=Name
        self.Salary=Salary
        Employee.empCount=Employee.empCount+1
    def displayEmployee(self):
        print("Name : ",self.Name,",","Salary : ",self.Salary)
    def displayCount(self):
        print("Total Employee %d" % Employee.empCount)
emp1=Employee("Zara",2000)
emp2=Employee("Manni",5000)
emp3=Employee("Amy",5000)
emp1.displayEmployee()
emp2.displayEmployee()
emp3.displayEmployee()
emp3.displayCount()
```

```

In [9]: 1 class Employee:
2         empCount=0
3         def __init__(self,Name,Salary):
4             self.Name=Name
5             self.Salary=Salary
6             Employee.empCount=Employee.empCount+1
7         def displayEmployee(self):
8             print("Name : ",self.Name,",","Salary : ",self.Salary)
9         def displayCount(self):
10            print("Total Employee %d" % Employee.empCount)
11 emp1=Employee("Zera",2000)
12 emp2=Employee("Manni",5000)
13 emp3=Employee("Amy",5000)
14 emp1.displayEmployee()
15 emp2.displayEmployee()
16 emp3.displayEmployee()
17 emp3.displayCount()

Name : Zera , Salary : 2000
Name : Manni , Salary : 5000
Name : Amy , Salary : 5000
Total Employee 3

```

EX 7: 對於一個運動學公式:  $S = V_0 t + \frac{1}{2} g t^2$ ，其中  $V_0$  代表初速， $t$  為時間， $g = 9.8$ 。試寫出一個命名為 `kinematic` 的 Python 類別，該類別的輸入參數為初速  $V_0$ ，而類別裡的方法(`disFormula`)在給予時間  $t$  這個參數後，可以計算出移動距離  $S$ 。

例如：

```

computDis = kinematic(4) # V0 = 4 m/s
computDis.disFormula(2)  # t = 2s

```

27.62

圖 5-22 輸入與輸出結果



```

class kinematic:
    def __init__(self,v):
        self.v=v
    def disFormula(self,t):
        S=self.v*t+9.8*t*t*0.5
        print(S)
computDis=kinematic(4)
computDis.disFormula(2)

```

```

In [10]: 1 class kinematic:
2         def __init__(self,v):
3             self.v=v
4         def disFormula(self,t):
5             S=self.v*t+9.8*t*t*0.5
6             print(S)
7 computDis=kinematic(4)
8 computDis.disFormula(2)

```

27.6

EX 8: 試建立一個命名為 calculation 的 Python 模組，該模組提供加、減、乘、除等功能的運算。

例如：

```

import calculation

print(calculation.add(4,8))
print(calculation.sub(4,8))
print(calculation.mul(4,8))
print(calculation.div(4,8))

```

```

12
-4
32
0.5

```

圖 5-23 輸入與輸出結果

Calculation.py

```
# coding: utf-8
```

```
# In[ ]:
```

```
def add(x,y):
```

```
    return x+y
```

```
def sub(x,y):
```

```
    return x-y
```

```
def mul(x,y):
```

```
    return x*y
```

```
def div(x,y):
```

```
    return x/y
```

localhost:8888/notebooks/Untitled%20Folder%20/U...

應用程式 小狐狸事務所: 201... 用 Python 理財: ... Python 基礎語法 |... Online C Compiler...

jupyter Untitled1 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

- New Notebook
- Open...
- Make a Copy...
- Rename...
- Save and Checkpoint
- Revert to Checkpoint
- Print Preview
- Download as
  - Notebook (.ipynb)
  - Python (.py)
  - HTML (.html)
  - Reveal.js slides (.html)
  - Markdown (.md)
  - reST (.rst)
  - LaTeX (.tex)
  - PDF via LaTeX (.pdf)
- Trusted Notebook
- Close and Halt

```
def add(x,y):  
    return x+y  
def sub(x,y):  
    return x-y  
def mul(x,y):  
    return x*y  
def div(x,y):  
    return x/y
```

localhost:8888/notebooks/Untitled Folder 2/Untitled1.ipynb?kernel\_name=python3#

N-CHAP05 Ho....docx calculation.py 全部顯示

localhost:8888/edit/Untitled%20Folder%202/calculati...  
應用程式 小狐狸事務所: 201... 用 Python 理財: ... Python 基础语法 |... Online C Comp

jupyter calculation.py ✓ 27 分鐘前 Login

File Edit View Language Pyth

```
1  
2 # coding: utf-8  
3  
4 # In[ ]:  
5  
6  
7 def add(x,y):  
8     return x+y  
9 def sub(x,y):  
10    return x-y  
11 def mul(x,y):  
12    return x*y  
13 def div(x,y):  
14    return x/y  
15  
16  
17 |
```

FilesRunningClusters

Select items to perform actions on them. 

UploadNew↻

<input type="checkbox"/> 0 ▾	/ Untitled Folder 2	Name ▾	Last Modified	File size
	Ⓛ ..		幾秒前	
<input type="checkbox"/>	📄 Untitled.ipynb		Running 1 分鐘前	4.43 kB
<input type="checkbox"/>	📄 Untitled1.ipynb		Running 28 分鐘前	760 B
<input type="checkbox"/>	📄 calculation.py		28 分鐘前	151 B
<input type="checkbox"/>	📄 N-CHAP05 Homework2.docx		25 分鐘前	153 kB

In [8]:

```

1 import calculation
2
3 print(calculation.add(4,8))
4 print(calculation.sub(4,8))
5 print(calculation.mul(4,8))
6 print(calculation.div(4,8))

```

```

12
-4
32
0.5

```

In [ ]:

```

1

```