COEN 166 Artificial Intelligence

Lab Assignment #1 Report

Gagan Gupta **SCU#00001478479**

Part I:

Exercise 1:

```
>>> a=123+222
>>> print(a)
345
>>> b=1.5*4
>>> print(b)
6.0
>>> c=2**10
>>> print(c)
1024
>>> import math
>>> print(math.pi)
3.141592653589793
>>> print(math.sqrt(36))
6.0
>>> import random
>>> a=random.random()
>>> print('a=',a)
a= 0.6544249412322283
>>> b=random.choice([1,2,3,4])
>>> print('b=',b)
b= 4
```

Exercise 2:

```
>>> S='Spam'
>>> len(S)
4
>>> S[0]
'S'
>>> S[1]
'p'
>>> S[-1]
'm'
>>> S[-2]
'a'
>>> S[len(S)-1]
'm'
>>> S[1:3]
'pa'
>>> S='z'+S[1:]
>>> S
'zpam'
>>>
```

Exercise 3:

```
>>> L=[123, 'spam', 1.23]
>>> len(L)
3
>>> L[0]
123
>>> L[:-1]
[123, 'spam']
```

```
>>> L+[4,5,6]
[123, 'spam', 1.23, 4, 5, 6]
>>> L*2
[123, 'spam', 1.23, 123, 'spam', 1.23]
>>> L
[123, 'spam', 1.23]
>>> M=['bb','aa','cc']
>>> M.sort()
>>> M
['aa', 'bb', 'cc']
>>> M.reverse()
>>> M
['cc', 'bb', 'aa']
>>> M = [[1,2,3], [4,5,6], [7,8,9]]
>>> M[1]
[4, 5, 6]
>>> M[1][2]
6
>>> diag = [M[i][i] for i in [0, 1, 2]]
>>> diag
[1, 5, 9]
>>> doubles = [c * 2 for c in 'spam']
>>> doubles
['ss', 'pp', 'aa', 'mm']
>>> list(range(4))
[0, 1, 2, 3]
>>> list(range(-6,7,2))
[-6, -4, -2, 0, 2, 4, 6]
```

```
>>> [[x ** 2, x ** 3] for x in range(4)]
[[0, 0], [1, 1], [4, 8], [9, 27]]
>>> [[x, x/2, x * 2] for x in range(-6, 7, 2) if x > 0]
[[2, 1.0, 4], [4, 2.0, 8], [6, 3.0, 12]]
```

Exercise 4:

```
>>> D = {'food': 'Spam', 'quantity': 4, 'color': 'pink'}
>>> D['food']
'Spam'
>>> D['quantity']+=1
>>> D
{'food': 'Spam', 'quantity': 5, 'color': 'pink'}
>>> D={}
>>> D['name']='Bob'
>>> D['job'] = 'dev'
>>> D['age'] = 40
>>> D
{'name': 'Bob', 'job': 'dev', 'age': 40}
>>> print(D['name'])
Bob
>>> bob1 = dict(name='Bob', job='dev', age=40)
>>> bob1
{'name': 'Bob', 'job': 'dev', 'age': 40}
>>> bob2 = dict(zip(['name', 'job', 'age'], ['Bob', 'dev', 40]))
>>> bob2
{'name': 'Bob', 'job': 'dev', 'age': 40}
```

Exercise 5:

```
>>> T = (1, 2, 3, 4)
>>> len(T)
```

```
4
>>> T + (5, 6)
(1, 2, 3, 4, 5, 6)
>>> T[0]
1
>>> T.index(4)
3
>>> T.count(4)
1
>>> T[0]=2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
>>> T = (2,) + T[1:]
>>> T
(2, 2, 3, 4)
>>> T = 'spam', 3.0, [11, 22, 33]
>>> T[1]
3.0
>>> T[2][1]
22
```

Exercise 6:

```
block2
block1
block0
1.99
```

Exercise 7:

```
>>> x='spam'
```

```
>>> while x:
...     print(x,end='')
...     x=x[1:]
...
spampamamm>>> a=0;b=10
>>> while a<b:
...     print(a,end='')
...     a+=1
...
0123456789>>> x=10
>>> while x:
...     x=x-1
...     if x%2 !=0: continue
...     print(x,end='')
...
86420>>> for x in ["spam","eggs","ham"]:
...     print(x,end='')
...
spameggsham>>> sum=0
>>> for x in [1,2,3,4]:
...     sum=sum+x
...
>>> sum
10
>>> prod=1
>>> for item in [1,2,3,4]: prod*=item
...
>>> prod
```

24

Part II:

Exercise 8:

Answer:

Fun1:

8

OkOkOkOk

Fun2:

['S', 'A', 'M']

[1]

Comments:

Fun1 multiplies the two parameters passed in. With 2 numbers it'll mathematically multiple them while with a string and number it'll repeat the string that many times. Fun2 finds the intersection of the 2 datatype arrays passed in. For stings it treats it as a char array.

Exercise 9:

Answer:

Test1_module:

30

Test2_module:

15

200

Test3_module:

30

200

Comments:

Test1 imports module and uses the adder function and the a and b variables. Test2 first imports adder from module to use with its local c and d variables and then imports multiplier, a, and b from module to run module's multiplier with module's a and b variables. Test3 imports all the attributes from module and it then able to use everything without the "module.".

Exercise 10:

Answer:

Minmax:

1

6

Minimax2:

1

6

After import statements:

>>> import minmax

1

6

>>> import minimax2

Comments:

When Minmax is ran both manually and through using import in the shell, it output the 1 and the 6. When Minimax2 is run manually it outputs 1 and 6 however when it is ran though the shell using the import statement it does not output the 1 and the 6. This is due to the print statements being put into “if __name__ == '__main__':” for the minimax2 file.

Exercise 11:

Answer:

Class1:

King Arthur

-5

QQ

-3

QQ

-3

spam

Class2:

10

20

-15

Class3:

['dev', 'mgr']

('Sue', ['dev', 'cto'])

Class_as_module:

['dev', 'mgr']

('Sue', ['dev', 'cto'])

30

('Jane', ['dev', 'mgr'])

35

('Mike', ['dev', 'mgr'])

Comments:

Class1 constructs a class called FirstClass and then makes an object for that class. It then procceds to change various aspects of the object and displays the variables changing accordingly. Class2 inherits its own FirstClass into SecondClass and shows how due to the inheritance objects of SecondClass can use the function of FirstClass. Class3 demonstrates the constructor method by initializing two separate objects from the person class and then uses separate methods from the class on each of them. Class_as_module imports class3 to use the person class to make rec3 and rec4. These objects work as intended however before either of their function calls the "import class3" line cause the print statements from the class3 file to run therefore showing them in the output. "from class3 import Person" does not create this issue as it only import the class and not the statements after it.