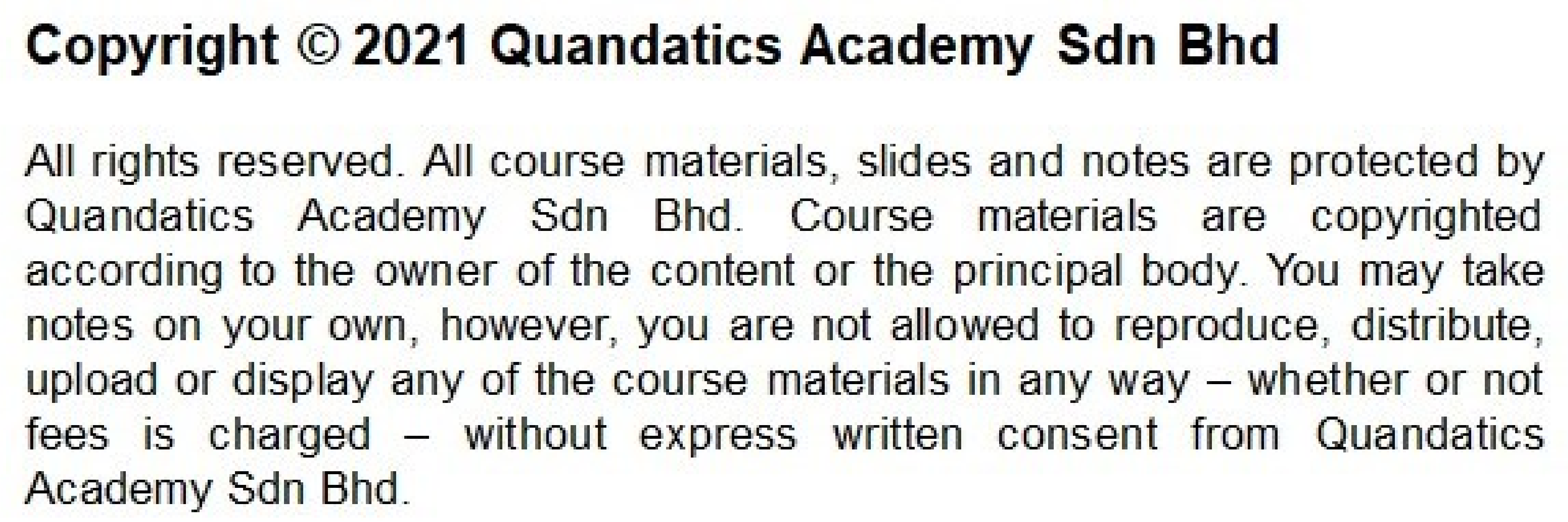
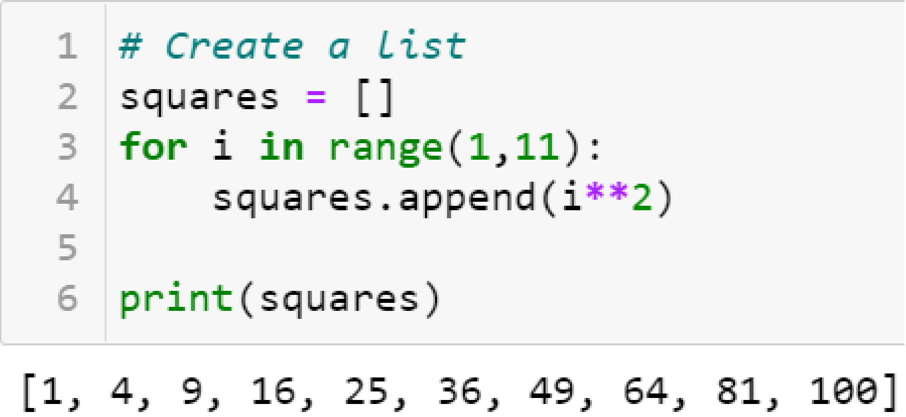


Additional Knowledge : List Comprehension

Trainer : Foo Chee Chuan

List Comprehension

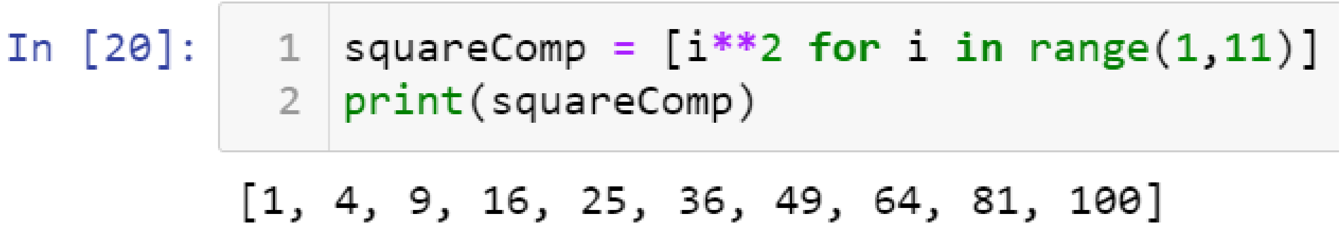
Create a list to store the squares of numbers (x_i^2)



```python
# Create a list
squares = []
for i in range(1,11):
    squares.append(i**2)

print(squares)
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

[*expression* for *value* in *collection*]



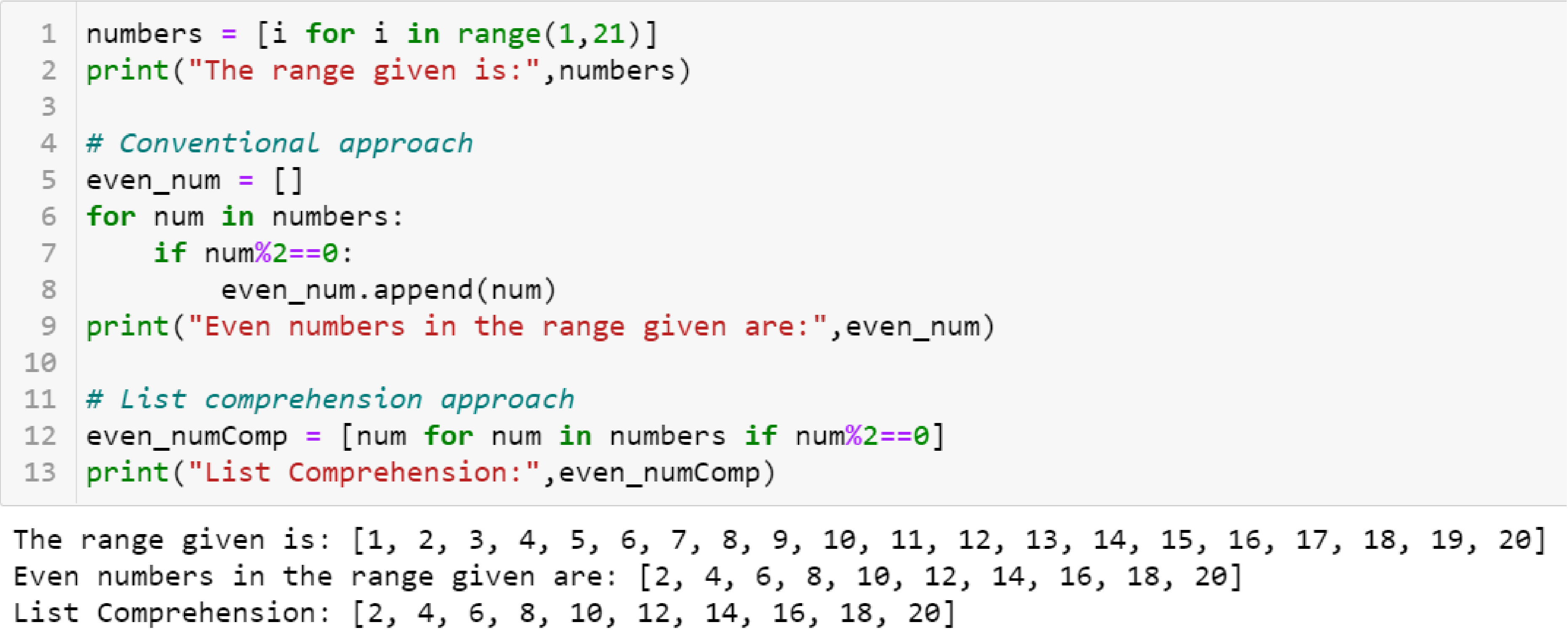
```
In [20]: squareComp = [i**2 for i in range(1,11)]
         print(squareComp)

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

expression : ***i**2***

value : ***i***

collection : ***range(1,11)***

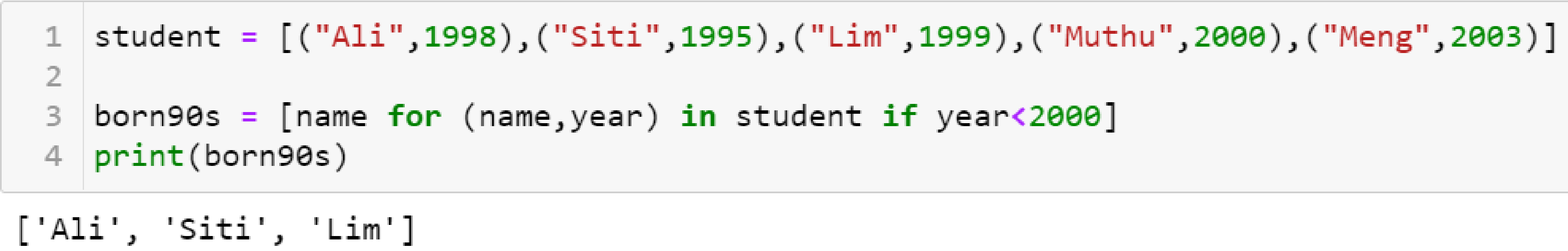
[***expression*** for ***value*** in ***collection*** if <***test>***]



```
numbers = [i for i in range(1,21)]
print("The range given is:",numbers)

# Conventional approach
even_num = []
for num in numbers:
    if num%2==0:
        even_num.append(num)
print("Even numbers in the range given are:",even_num)

# List comprehension approach
even_numComp = [num for num in numbers if num%2==0]
print("List Comprehension:",even_numComp)
```

```
The range given is: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
Even numbers in the range given are: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
List Comprehension: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

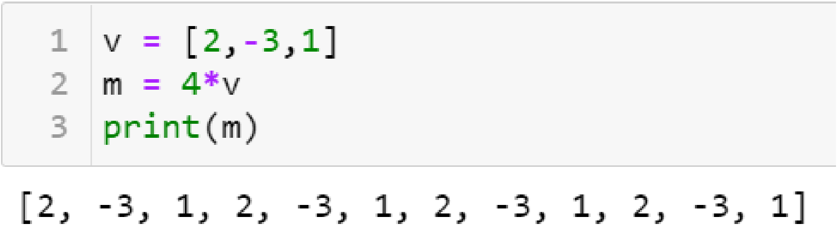
Multiple Variables



```
student = [("Ali",1998),("Siti",1995),("Lim",1999),("Muthu",2000),("Meng",2003)]

born90s = [name for (name,year) in student if year<2000]
print(born90s)
```

```
['Ali', 'Siti', 'Lim']
```

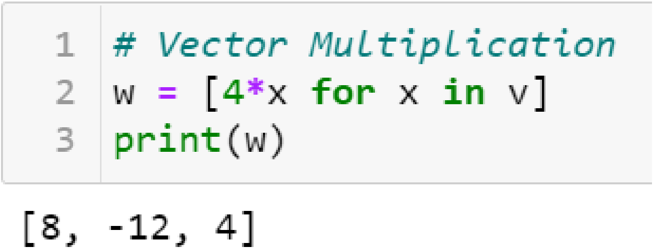
Practice: Vector Multiplication



```
v = [2,-3,1]
m = 4*v
print(m)
```

```
[2, -3, 1, 2, -3, 1, 2, -3, 1, 2, -3, 1]
```

Answer: List Comprehension

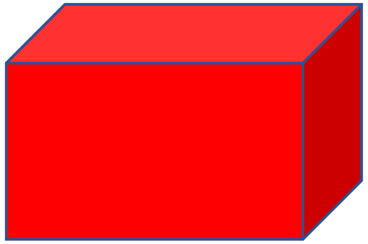


```
# Vector Multiplication
w = [4*x for x in v]
print(w)
```

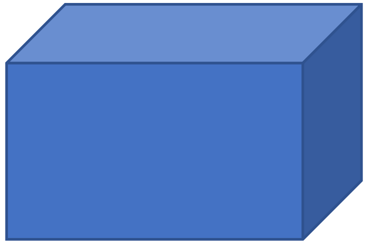
```
[8, -12, 4]
```

Cartesian Product:

If A and B are sets, then the Cartesian product is the set of pairs(a,b) where 'a' is in A and 'b' is in B.



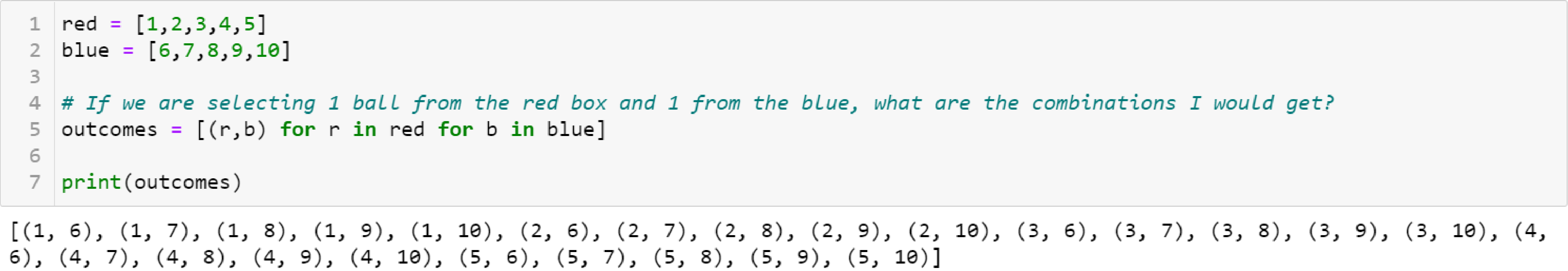
red = [1,2,3,4,5]



blue = [6,7,8,9,10]

If we are selecting 1 ball from the **red box** and 1 from the **blue box**, what are all the possible outcomes I can get?

[*expression* for *val1* in *collection1* ***and*** *val2* in *collection2*]



```python
red = [1,2,3,4,5]
blue = [6,7,8,9,10]

# If we are selecting 1 ball from the red box and 1 from the blue, what are the combinations I would get?
outcomes = [(r,b) for r in red for b in blue]

print(outcomes)
```

```
[(1, 6), (1, 7), (1, 8), (1, 9), (1, 10), (2, 6), (2, 7), (2, 8), (2, 9), (2, 10), (3, 6), (3, 7), (3, 8), (3, 9), (3, 10), (4,
6), (4, 7), (4, 8), (4, 9), (4, 10), (5, 6), (5, 7), (5, 8), (5, 9), (5, 10)]
```

Practice: Matching students

Class A (4 students)

A111: 70 marks

A112: 25 marks

A113: 11 marks

A114: 80 marks

Class B (4 students)

B121: 90 marks

B122: 13 marks

B123: 66 marks

B124: 55 marks

Expected Outcome:

(‘A111’, ’B122’, 57)

(‘A111’, ’B123’, 4)

(‘A111’, ’B124’, 15)

(‘A112’, ’B122’, 12)

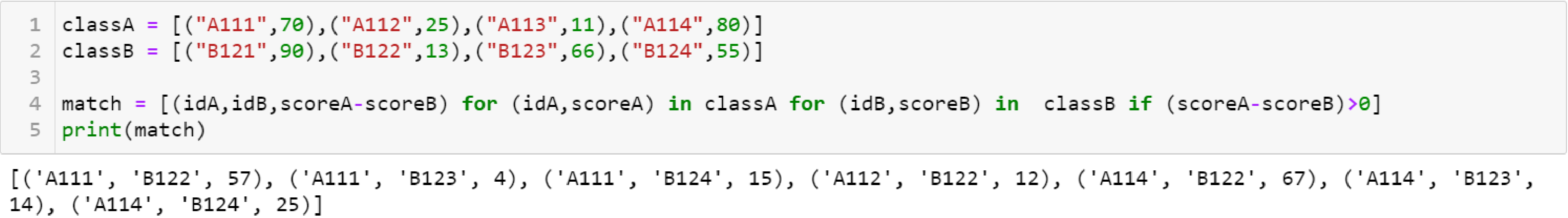
(‘A114’, ’B122’, 67)

(‘A114’, ’B123’, 14)

(‘A114’, ’B124’, 25)

You want to match the students in a competition so that Class A has the higher chance to win.
To do that, you want to show all combinations where students in
Class A has **higher score** than students in **Class B**.

Answer: Matching students



```python
classA = [("A111",70),("A112",25),("A113",11),("A114",80)]
classB = [("B121",90),("B122",13),("B123",66),("B124",55)]

match = [(idA,idB,scoreA-scoreB) for (idA,scoreA) in classA for (idB,scoreB) in  classB if (scoreA-scoreB)>0]
print(match)
```

```
[('A111', 'B122', 57), ('A111', 'B123', 4), ('A111', 'B124', 15), ('A112', 'B122', 12), ('A114', 'B122', 67), ('A114', 'B123',
14), ('A114', 'B124', 25)]
```

Thank you