

LIST COMPREHENSION

List Comprehension is a shorthand to define and create lists using loops and conditions.

Example: Using list comprehension, create a list that contain values from 1 to 10.

```
L=[x for x in range(1,11)]
```

This statement is a shorthand for :

```
L=[]
```

```
for x in range (1,11):
```

```
    L.append(x)
```

Q: Using LC, create a list that contain the values of the table of 5.

```
L=[x*5 for x in range(1,11)]
```

Output: 5,10,15,20,25,30,35,40,45,50

List comprehension using nested loop:

```
L1=[1,2,3]
```

```
L2=[4,5,6]
```

```
L3=[x*y for x in L1 for y in L2]
```

Output:[4,5,6,8,10,12,12,15,18]

Q: Write LC for producing a list of numbers between 1 and 1000 that are divisible by 3.

```
L=[x for x in range(1000) if x%3==0]
```

Q: What is the LC equivalent for:

{x:x is a whole number less than 20,x is even} (including zero)

```
L=[x for x in range(20) if x%2==0]
```

Output: [0,2,4,6,8,10,12,14,16,18]

MCQs

1. Write the list comprehension to pick out only negative integers from a given list 'l'.

- a) `[x<0 in l]`
- b) `[x for x<0 in l]`
- c) `[x in l for x<0]`
- d) `[x for x in l if x<0]`

2. What will be the output of the following Python code?

```
s=["pune", "mumbai", "delhi"]  
[(w.upper(), len(w)) for w in s]
```

- a) Error
- b) `['PUNE', 4, 'MUMBAI', 6, 'DELHI', 5]`
- c) `[PUNE, 4, MUMBAI, 6, DELHI, 5]`
- d) `[('PUNE', 4), ('MUMBAI', 6), ('DELHI', 5)]`

3. What will be the output of the following Python code?

```
l1=[2,4,6]
```

```
l2=[-2,-4,-6]
```

```
# The purpose of zip() is to map the similar index of multiple containers so that they can  
# be used just using
```

```
# as a single entity.
```

```
for i in zip(l1, l2):
```

```
    print(i)
```

- a)
 - 2, -2
 - 4, -4
 - 6, -6
- b) `[(2, -2), (4, -4), (6, -6)]`
- c)
 - (2, -2)
 - (4, -4)
 - (6, -6)
- d) `[-4, -16, -36]`

4. Write a list comprehension for number and its cube for `l=[1, 2, 3, 4, 5, 6, 7, 8, 9]`.

- a) `[x**3 for x in l]`
- b) `[x^3 for x in l]`

c) `[x**3 in l]`

d) `[x^3 in l]`

5. What will be the output of the following Python code?

```
l=[[1 ,2, 3], [4, 5, 6], [7, 8, 9]]
```

```
[[row[i] for row in l] for i in range(3)]
```

a) Error

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

c)

1 4 7

2 5 8

3 6 9

d)

(1 4 7)

(2 5 8)

(3 6 9)

6. Read the information given below carefully and write a list comprehension such that the output is: `['e', 'o']`

```
w="hello"
```

```
v=('a', 'e', 'i', 'o', 'u')
```

a) `[x for w in v if x in w]`

b) `[x for x in w if x in v]`

c) `[x for x in v if w in v]`

d) `[x for v in w for x in w]`

7. What will be the output of the following Python code?

```
[ord(ch) for ch in 'abc']
```

a) `[97, 98, 99]`

b) `['97', '98', '99']`

c) `[65, 66, 67]`

d) Error

8. Write a list comprehension equivalent for the Python code shown below.

```
for i in range(1, 101):
```

```
if int(i*0.5)==i*0.5:  
    print(i)
```

- a) [i for i in range(1, 100) if int(i*0.5)==(i*0.5)]
- b) [i for i in range(1, 101) if int(i*0.5)==(i*0.5)]
- c) [i for i in range(1, 101) if int(i*0.5)=(i*0.5)]
- d) [i for i in range(1, 100) if int(i*0.5)=(i*0.5)]

9. What will be the output of the following Python list comprehension?

```
[j for i in range(2,8) for j in range(i*2, 50, i)]
```

- a) A list of prime numbers up to 50
- b) A list of numbers divisible by 2, up to 50
- c) A list of non prime numbers, up to 50
- d) Error

10. What will be the output of the following Python code?

```
l=["good", "oh!", "excellent!", "#450"]  
[n for n in l if n.isalpha() or n.isdigit()]
```

- a) ['good', 'oh', 'excellent', '450']
- b) ['good']
- c) ['good', '#450']
- d) ['oh!', 'excellent!', '#450']