

## Multiple Exceptions

- If the Index is not given properly it'll give **index error**
- You can handle a particular exception as well by simply defining except with that error, then this except block will not handle any other except block other than what is defined
- If exception block is not defined it'll handle all type of exceptions.

```
l = [10, 20, 30, 40, 50]
try:
    index = int(input('enter index')) # take int as I/P from user
    print(l[index]) # printing the result
    print('end of try block') # indicting end of try block

except: # if problem occur it enters except block
    print('invalid index') # prints an exception has occurred

print('terminate gracefully') # prints that a program has ended gracefully
```

output 1

```
enter index9
invalid index
terminate gracefully
```

output 2

```
enter indexxyz
invalid index
terminate gracefully
```

- You can write multiple except block to handle different type of exception

```
l = [10, 20, 30, 40, 50]
try:
    index = int(input('enter index')) # take int as I/P from user
    print(l[index]) # printing the result
    print('end of try block') # indicting end of try block

except IndexError: # when wrong/no index this exception is raised
    print('invalid index')
except ValueError: # when proper value is not given this exception is raised
    print('enter only integer value')

print('terminate gracefully') # prints that a program has ended gracefully
```

Output 1

```
enter index9
invalid index
terminate gracefully
```

output 2

```
enter indexxyz
enter only integer value
terminate gracefully
```

- Instead of writing two except block we can write a single except block that can handle both the exception

```
l = [10, 20, 30, 40, 50]
try:
    index = int(input('enter index')) # take int as I/P from user
    print(l[index]) # printing the result
    print('end of try block') # indicting end of try block

except (IndexError, ValueError) as msg: # writing multiple exceptions
    # in single block
    print(msg)

print('terminate gracefully') # prints that a program has ended gracefully
```

Output 1

```
enter index9
list index out of range
terminate gracefully
```

output 2

```
enter indexxyz
invalid literal for int() with base 10: 'xyz'
terminate gracefully
```