Raise Exceptions in python

raise Without Specifying Exception Class :

When we use the raise keyword, it's not necessary to provide an exception class along with it. When we don't provide any exception class name with the raise keyword, it re-raises the exception that last occurred.

This is used generally inside an except code block to re-raise an exception which is catched.

For example,

```
a = 10
b = 0
try:
    print(a/b)
except ZeroDivisionError:
    raise
```

```
Output:
Traceback (most recent call last):
  File "main.py", line 4, in
    print(a/b)
ZeroDivisionError: division by zero
```

raise With an Argument :

We can also provide an argument while raising an exception which is displayed along with the exception class name on the console.

We can pass any string for describing the reason for the exception or anything else.

```
raise ValueError("Incorrect Value")
```

```
Output:
Traceback (most recent call last):
  File "main.py", line 1, in <module>
    raise ValueError("Incorrect Value")
ValueError: Incorrect Value
```