Exceptions in Python

Exception causes the code to stop if there is an error, for example if you input a string for what is supposed to be an integer, it will show “ValueError” as followed.

```
a = int(input())
print(a)

# HelloWorld
# Traceback (most recent call last):
#   File "d:\LSC\Teams\Computer Team\testing.py", line 1, in <module>
#     a = int(input())
# ValueError: invalid literal for int() with base 10: 'HelloWorld'
```

Prevention:

Try and except branch can help

```
try:
    a = int(input())
    print(a)
except:
    print("this is not an integer")
```

Once an error occurred in the try branch, it will go to the except branch, the terminal doesn’t show any error.

Either of those is executed successfully, the program continues.

We can also specify the error

```
try:
    a = float(input())
    print(1 / a)
except ValueError:
    print("Please input a number")
except ZeroDivisionError:
    print("Please input a non-zero number")
```

Still like if-elif branch you can add except at the end

```
try:
    a = float(input())
    print(1 / a)
except ValueError:
    print("Please input a number")
except ZeroDivisionError:
    print("Please input a non-zero number")
except:
    print("Something unexpected happened")
```

Normal errors

```
short_list = [1]
short_list = [1]
short_list.append(2)
short_list.depend(3) # AttributeError

short_list = [1]
one_value = short_list[0.5] # TypeError

print(1 / 0) # ZeroDivisionError

s = "abc"
a = int(s) # ValueError

# SyntaxError
```