Python, exceptions are a powerful mechanism for handling errors and other exceptional events that may occur during the execution of a program. Here are some common exceptions you might encounter:

1. **SyntaxError**: Raised when the parser encounters a syntax error.
2. **IndentationError**: Raised when there are issues with indentation.
3. **NameError**: Raised when a local or global name is not found.
4. **TypeError**: Raised when an operation or function is applied to an object of inappropriate type.
5. **IndexError**: Raised when a sequence subscript is out of range.
6. **ValueError**: Raised when a function receives an argument of the right type but an inappropriate value.
7. **KeyError**: Raised when a dictionary key is not found.

**8. AttributeError**: Raised when an attribute reference or assignment fails.Attempting to call a method that does not exist

**9.ImportError**: Raised when an import statement fails.

**10.IOError**: Raised when an I/O operation fails.

**11.ZeroDivisionError**: Raised when division by zero occurs.

# Example code that causes a Zerodivision error

try:

x = 10 / 0

except ZeroDivisionError:

print("You can't divide by zero!")

# Example code that causes a index and value error

try:

# code that may raise different types of exceptions

x = int("not a number")

y = [1, 2, 3]

print(y[10])

except (ValueError, IndexError) as e:

print(f"An error occurred: {e}")

# Example code that causes a TypeError

def add\_numbers(a, b):

return a + b

result = add\_numbers(5, "3")

print(result)