**Topic 76: Planning for Things to Go Wrong**

**What**  
This topic explains how to handle errors or exceptions in your Python program using try and except blocks. It demonstrates how to deal with situations where something unexpected goes wrong, like when a user provides an incorrect file name, ensuring your program doesn't crash but instead handles the error gracefully.

**Why**

* **Graceful Error Handling**: When an error occurs, instead of letting the program stop abruptly with an unclear message, you can catch the error and provide a helpful message to the user.
* **Improved User Experience**: Clear error messages help the user understand what went wrong and how to fix it without terminating the program.
* **Prevent Program Crashes**: Without error handling, errors could cause your program to stop unexpectedly. Using try and except ensures your program continues running even when some part fails.

**How**

1. **Using try block**:  
   Wrap the code that might produce an error in a try block. In this case, the code that asks for a file name, opens the file, and reads it:

python

Copy code

try:

filename = input("What text file to open? ")

with open(filename) as f:

print(f.read())

1. **Catching specific exceptions with except**:  
   If the code in the try block causes an error, use the except block to catch the specific exception. For example, if the file doesn’t exist, a FileNotFoundError will be raised, and you can handle it:

python

Copy code

except FileNotFoundError:

print("Sorry, " + filename + " not found.")

1. **Indentation**:  
   The code within the try and except blocks must be indented. This is required in Python to differentiate the code within these blocks:

python

Copy code

try:

# Code that might cause an error

except ErrorType:

# Code that handles the error

1. **Handling Multiple Errors**:  
   Python allows you to handle multiple types of errors. You can include multiple except blocks for different errors:

python

Copy code

try:

# Some risky code

except FileNotFoundError:

print("File not found")

except ValueError:

print("Value error occurred")

1. **Generic Exception Handling**:  
   If you don’t know which specific error might occur, you can catch all exceptions with a general except:

python

Copy code

except Exception as e:

print("An error occurred:", e)

**Things to Remember**

* **try block**: This is where you put the code that might cause an error.
* **except block**: If an error occurs in the try block, the program will jump to the except block and handle the error.
* **Specific Errors**: You should try to catch specific errors (like FileNotFoundError) to give users clear messages about what went wrong.
* **Indentation**: Code inside try and except blocks must be indented.
* **Error Types**: You can catch multiple error types by having multiple except blocks or use a generic except to catch all errors.