Exceptions:

- Exceptions are events that can modify the flow of control through a program. They provide a coherent way to respond to unusual events.

- Examples of types of exceptions that can occur:
    1. **FileNotFoundError** – attempt to open a non-existent file to read with 'r' or 'r+' modes.
    2. **ZeroDivisionError –** attempt to divide by 0.
    3. **IndentationError** – incorrect indentation.

- They are processed by these statements:
    1. ***try/except*** – catch only exceptions raised within the statements in the associated ***try*** block
    2. ***try/finally*** – to guarantee termination actions will fire regardless of exceptions in the ***try*** block's code
    3. ***raise*** – trigger an exception manually for built-in exceptions
    4. ***assert –*** used when debugging code. Test if a condition in your code returns True. If not, the program will raise an AssertionError.
    5. ***with/as –*** allows objects like files to be used to ensure prompt clean-up.

- Most common roles:
    1. Error handling – program can resume execution after ***try.***
    2. Event notification – signal valid conditions without having to test them explicitly.
    3. Special-case handling – clean ups code by making the exception handler at a high level to handle rare cases.
    4. Termination actions – ***try/finally*** allows for a guarantee that closing-time operations will be performed.
    5. Unusual control flows – unwind assignments. A ***raise*** can be used to jump out of multiple loops.