**Experiment: 8**

**Name**: Sahil Ashok Jagdale  
**PRN**: 23410005  
**Experiment No**: 8  
**Branch**: Electronics (EN-1)  
**Software Used**: Thonny

**Aim**: Programs to study Python Exceptions, Python Exception Handling and Python Custom Exceptions

**Theory:**

**Python Exceptions**

An **exception** in Python is an error that occurs during the execution of a program, which disrupts the normal flow of instructions. Exceptions occur due to various reasons such as invalid input, division by zero, accessing a non-existent file, etc. When an exception occurs, Python stops the execution and displays an error message unless it is properly handled. Common types of exceptions include ZeroDivisionError, ValueError, IndexError, FileNotFoundError, and many more.

**Python Exception Handling**

**Exception handling** is a mechanism in Python to gracefully manage and respond to runtime errors without crashing the program. It is done using the try, except, else, and finally blocks:

* **try block** contains the code that might raise an exception.
* **except block** is used to handle the exception if it occurs.
* **else block** runs if no exception occurs in the try block.
* **finally block** executes code regardless of whether an exception occurred or not.

Exception handling ensures the program remains robust and continues running even when errors occur.

**Python Custom Exceptions**

**Custom exceptions** are user-defined error types in Python. They are created by defining a new class that inherits from the built-in Exception class. Custom exceptions allow programmers to define their own error types specific to their application, making the code more readable and meaningful. By raising custom exceptions, developers can better indicate specific problem situations in their program logic.

Program:

Output: