Exception Hierarchy

In Python, exceptions are organized in a hierarchy, with the base class being `BaseException`. All other built-in exceptions inherit from this base class. Here's a brief overview of the Python exception hierarchy with examples:

BaseException:

- **Example:**
  ```python
  try:
      # Some code that might raise an exception
      result = 10 / 0
  except BaseException as e:
      print(f"Caught an exception: {e}")
  ```

Exception:

- All built-in exceptions directly or indirectly inherit from the `Exception` class.

- **Example:**
  ```python
  try:
      # Some code that might raise an exception
      value = int("abc")
  except Exception as e:
      print(f"Caught an exception: {e}")
  ```

StandardError (deprecated):

- In Python 2, exceptions were categorized into `StandardError`, but this category has been removed in Python 3. All exceptions now directly inherit from `BaseException` or `Exception`.

ArithmeticError:

- Base class for arithmetic errors.

- **Example:**
  ```python
  try:
      result = 10 / 0
  except ArithmeticError as e:
      print(f"ArithmeticError: {e}")
  ```

LookupError:

- Base class for lookup errors.

- **Example:**
  ```python
  try:
      my_list = [1, 2, 3]
      value = my_list[10]
  except LookupError as e:
      print(f"LookupError: {e}")
  ```

ValueError:

- Raised when a built-in operation receives an argument of the correct type but with an invalid value.

- **Example:**
  ```python
  try:
      value = int("abc")
  except ValueError as e:
      print(f"ValueError: {e}")
  ```

TypeError:

- Raised when an operation or function is applied to an object of an inappropriate type.

- **Example:**
  ```python
  try:
      result = 10 + "5"
  except TypeError as e:
      print(f"TypeError: {e}")
  ```

ZeroDivisionError:

- Raised when the second argument of a division or modulo operation is zero.

- **Example:**
  ```python
  try:
      result = 10 / 0
  except ZeroDivisionError as e:
      print(f"ZeroDivisionError: {e}")
  ```

IndexError:

- Raised when a sequence subscript is out of range.

- **Example:**
  ```python
  try:
      my_list = [1, 2, 3]
      value = my_list[10]
  except IndexError as e:
      print(f"IndexError: {e}")
  ```

These are just a few examples, and the Python exception hierarchy includes many more specific exceptions for various error types. Understanding this hierarchy can help in choosing the right exceptions to catch in your code and handling them appropriately.