Lesson Plan:

List Of General Use Exception

PW SKILLS

In Python, there is a built-in hierarchy of exception classes that can be used for handling various types of errors. Here is a list of general-use exception classes in Python:

- **Exception**: This is the base class for all built-in exceptions. It catches all exceptions that inherit from it.
- **ArithmeticError**: This is the base class for arithmetic errors, and it includes exceptions like ZeroDivisionError and OverflowError.
- **LookupError**: This is the base class for lookup errors, and it includes exceptions like IndexError and KeyError.
- **AssertionError**: Raised when an assert statement fails.
- **AttributeError**: Raised when an attribute reference or assignment fails.
- **EOFError**: Raised when the input() function hits an end-of-file condition without reading any data.
- **FileNotFoundError**: Raised when a file or directory is requested but cannot be found.
- **ImportError**: Raised when an import statement fails to find the module.
- **ModuleNotFoundError**: A subclass of ImportError, raised when a specific module is not found.
- **OSError**: This is the base class for I/O and system-related errors. It includes exceptions like FileExistsError, FileNotFoundError, and PermissionError.
- **TypeError**: Raised when an operation or function is applied to an object of an inappropriate type.
- **ValueError**: Raised when a built-in operation or function receives an argument with the right type but an inappropriate value.
- **ZeroDivisionError**: Raised when the second operand of a division or modulo operation is zero.
- **KeyError**: Raised when a dictionary key is not found.
- **IndexError**: Raised when a sequence subscript is out of range.
- **NameError**: Raised when a local or global name is not found.
- **NotImplementedError**: Raised when an abstract method that needs to be implemented in an inherited class is not actually implemented.