**1) Explain why we have to use the Exception class while creating a Custom Exception.**

Ans- Custom exceptions provide you the flexibility to add attributes and methods that are not part of a standard Java exception. These can store additional information, like an application-specific error code, or provide utility methods that can be used to handle or present the exception to a user.

**2) Write a python program to print Python Exception Hierarchy.**

Ans- import inspect

print "The class hierarchy for built-in exceptions is:"

inspect.getclasstree(inspect.getmro(BaseException))

def classtree(cls, indent=0):

print '.' \* indent, cls.\_\_name\_\_

for subcls in cls.\_\_subclasses\_\_():

classtree(subcls, indent + 3)

classtree(BaseException)

**3) What errors are defined in the Arithmetic Error class? Explain any two with an example.**

Ans- ArithmeticError is thrown when an error occurs while performing mathematical operations. These errors include attempting to perform a bitshift by a negative amount, and any call to intdiv() that would result in a value outside the possible bounds of an int.

ArithmeticError is simply an error that occurs during numeric calculations.

ArithmeticError types in Python include:

* OverFlowError
* ZeroDivisionError
* FloatingPointError

arithmetic = 5/0

print(arithmetic)

try:

arithmetic = 5/0

print(arithmetic)

except ArithmeticError:

print('You have just made an Arithmetic error')

**4) Why LookupError class is used? Explain with an example KeyError and IndexError.**

Ans- The LookupError exception in Python forms the base class for all exceptions that are raised when an index or a key is not found for a sequence or dictionary respectively. You can use LookupError exception class to handle both IndexError and KeyError exception classes.

Containers like lists and dictionaries will generate errors if you try to access items in them that do not exist. For lists, this type of error is called an IndexError ; for dictionaries, it is called a KeyError

if you have a list with three items and you try to access the fourth item, you will get an IndexError. This can happen with strings, tuples, lists, and generally any object that is indexable.

The Python KeyError is an exception that occurs when an attempt is made to access an item in a dictionary that does not exist. The key used to access the item is not found in the dictionary, which leads to the KeyError

**5) Explain ImportError. What is ModuleNotFoundError?**

Ans- Python's ImportError ( ModuleNotFoundError ) indicates that you tried to import a module that Python doesn't find. It can usually be eliminated by adding a file named \_\_init\_\_.py to the directory and then adding this directory to $PYTHONPATH .

Sometimes, Python throws the ModuleNotFoundError afterward. What does this error mean in Python? As the name implies, this error occurs when you're trying to access or use a module that cannot be found. In the case of the title, the "module named Python" cannot be found. Python here can be any module

**6) List down some best practices for exception handling in python.**

Ans- Use try/catch/finally blocks to recover from errors or release resources.

Handle common conditions without throwing exceptions.

Design classes so that exceptions can be avoided.

Throw exceptions instead of returning an error code.

Use the predefined .NET exception types.