EXCEPTION_HANDLING ASSIGNMENT

1. An exception is a python object that represents nothing but a runtime error. It basically occurs due to incorrect implementation of logic.
   E.g.- Like when we are using a mobile UPI, but we don’t have fund in our account so it is an exception as we can’t transfer values.

2. The difference between error and exception are:

# Error cannot be handled by the programmer

# Exception can easily be handled by any programmer

# Error cannot be recovered once occurred.

# Exception are generally those problems from which a program can recover.

The difference between error and exception are:

# Error cannot be handled by the programmer

# Exception can easily be handled by any programmer

# Error cannot be recovered once occurred.

# Exception are generally those problems from which a program can recover.

3. To ensure that a certain code block runs no matter whether there’s an exception is called “Finally Block”. Finally block is always executed when an exception is thrown.

4. Some examples of built in exception

ArithmeticError - Raised when an error occurs in numeric calculations

AttributeError - Raised when attribute reference or assignment fails

Exception - Base class for all exceptions

ImportError - Raised when an imported module does not exist

IndentationError - Raised when indentation is not correct

IndexError - Raised when an index of a sequence does not exist

KeyError - Raised when a key does not exist in a dictionary

NameError - Raised when a variable does not exist

OverflowError - Raised when the result of a numeric calculation is too large

SyntaxError - Raised when a syntax error occurs

TypeError - Raised when two different types are combined

UnicodeError - Raised when a unicode problem occurs

ValueError - Raised when there is a wrong value in a specified data type

ZeroDivisionError - Raised when the second operator in a divi

5. The exception handling in python are with the following ways:-
Try- It lets you test a block of code for error.
Except- It lets you handle the errors
Finally- It let you execute the code, regardless of the try and except block.

6. Syntax errors are detected when we have not followed the rules of the particular programming language while writing a program and that may cause of arising an exception, While exception can occur even without any syntax error, for example, exception like trying to open a file which does not exist, division by zero, etc. so this is how Every syntax error is an exception but every exception cannot be a syntax error

7. The raise statement is used to raise an exception. When you raise an exception, you signal to the calling code that something has gone wrong and that the current function is unable to handle the error. The raise statement is typically used in combination with the try and except blocks to handle and propagate exceptions

8. The else block in exception handling is used to specify code that should be executed if no exceptions are raised in the try block. The else block is optional, and it follows all the except blocks.

```
9. try:
 # Code that might raise an exception #
... Except Exception Type:
 # Code that will handle the exception of type Exception Type
```

10. The Purpose of finally block is to do clean up action, every time after completion of the program.

11. Try-except is used for catching and handling exceptions. • Try-finally is used for ensuring that the code is executed, regardless of an exception.