Exception Handling

Session Objectives

- Concept
- Exception Handling
- try and except Block
- Try with Else Clause
- Try with finally Clause
- Raise an exception
- Summary

- **Concept**

  - Error in Python can be of two types i.e. **Syntax errors and Exceptions**.
  - **Syntax Errors** are the problems in a program due to which the program will stop the execution. Mostly done by programer.
    - mystr=”hello
    - if i>10
  - **Exceptions** are raised when some internal events occur which changes the normal flow of the program.
    - a=10000/0
    - “ZeroDivisionError” exception as we are trying to divide a number by 0

–

–

– Syntax:

- Exception Handling
    - Try and except statements are used to catch and handle exceptions in Python.
    - Statements that can **raise exceptions** are kept inside the **try clause** and the statements that **handle the exception** are written inside **except clause.**
    - Syntax:

```
try:
    # Some Code....
except:
    # optional block
    # Handling of exception (if required)
else:
    # execute if no exception
finally:
    # Some code .....(always executed)
```

- Example

  -
    ```
    try:
      print(x)
    except:
       print("An exception occurred")
    ```

Output:An exception occurred #variarble x not defined

```
try:
  print(x)
except NameError:
  print("Variable x is not defined")
except:
  print("Something else went wrong")
```

Output: Variable x is not defined

Try with Else Clause

- You can also use the else clause on the try-except block which must be present after all the except clauses.
- The code enters the else block only if the **try clause does not raise an exception**.
- Example:

```
try:
  print("Hello")
except:
  print("Something went wrong")
else:
  print("Nothing went wrong")
```

Output:

```
Hello
Nothing went wrong
```

Try with finally Clause

- You can also use the finally clause on the try-except block which must be present after all the except clauses.
- The finally block will be executed **regardless if the try block raises an error or not.**.
- Example:

```
try:
  print("Hello")
except:
  print("Something went wrong")
finally:
  print("The 'try except' is finished")
```

Output:

```
Hello
The 'try except' is finished
```

- # Raise an exception
    - As a Python developer you can choose to throw an exception if a condition occurs.
    - To throw (or raise) an exception, use the raise keyword.
    - Example

```
x = -1


if x < 0:
  raise Exception("Sorry, no numbers below zero")
```

Output:

```
Exception: Sorry, no numbers below zero
```

Summary

- Concept
- Exception Handling
- try and except Block
- Try with Else Clause
- Try with finally Clause
- Raise an exception
- Demo