**Training Report Day-9**

**17 June 2024**

**Exception in Python:**

Exception handling in Python is a mechanism to respond to runtime errors, preventing the program from crashing and allowing the program to handle errors gracefully. It helps in debugging, maintaining clean code, and providing user-friendly error messages.

**Key Concepts**

**1. Exception:** An exception is an error that occurs during the execution of a program. When an exception is raised, the normal flow of the program is interrupted.

**2. Try Block**: The code that might raise an exception is placed inside a try block.

**3. Except Block**: The code that handles the exception is placed inside an except block.

**4. Else Block:** The code inside the else block is executed if no exceptions are raised.

**5. Finally Block:** The code inside the finally block is executed regardless of whether an exception is raised or not.

**6. Raise:** Used to raise an exception manually.

Some common built in exceptions:

* IndexError
* KeyError
* ValueError
* TypeError
* ZeroDivisionError
* FileNotFoundError
* IOError
* ImportError
* AttributeError
* RuntimeError

**Handling divide by zero exception:**

def divide(a, b):

    try:

        result = a / b

    except ZeroDivisionError:

        return "Cannot divide by zero!"

    else:

        return result

    finally:

        print("Execution of divide function complete.")

print(divide(10, 2))  # Output: 5.0

print(divide(10, 0))  # Output: Cannot divide by zero!