Computing Fundamentals using Python

SUBJECT CODE : UQ25CA151A

Samyukta D Kumta

Computer Applications

Example : Try with else and finally

```
try:
    num = int(input("Enter a number: "))
    result = 10 / num
except ZeroDivisionError:
    print("Cannot divide by zero!")
except ValueError:
    print("Invalid input, please enter a number.")
else:
    print("Result is:", result)
finally:
    print("Program execution completed.")
```

Raising an Exception

Syntax:

raise ExceptionType("Custom error message")

ExceptionType : the type of exception (e.g., ValueError, TypeError, RuntimeError, etc.)“

Custom error message" : message shown when the exception is raised

Raising an Exception

Raise

- Used to **manually raise exceptions** in Python.
- Can raise built-in exceptions (ValueError, TypeError, etc.) or custom exceptions.
- Usually combined with try-except to handle gracefully.
- **Example:**

```
if age < 0:
    raise ValueError("Age cannot be negative")
```

Raising an Exception

Example: Value Error

```
age = -5
if age < 0:
    raise ValueError("Age cannot be negative")
```

Raising an Exception

Example: Value Error

```
x = "10"
y = 20

try:
    if type(x) != int or type(y) != int:
        raise TypeError("Both x and y must be integers")
    result = x + y
    print("Result:", result)
except TypeError as e:
    print("Handled Exception:", e)
```

Multiple choice questions

1. What happens if an exception is not handled in Python?

A) The program continues execution normally
B) The program stops immediately with an error message
C) The program skips the faulty code and continues
D) The exception is ignored by Python

Answer: B

Multiple choice questions

2. Which block is always executed whether an exception occurs or not?

A) try
B) except
C) else
D) finally

Answer: D

3. What will be the output of this code?

```
try:
    print(10 / 0)
except ZeroDivisionError:
    print("Divided by zero")
else:
    print("No error")
finally:
    print("Done")
```

A) Divided by zero
B) No error
C) Divided by zero \n Done
D) No error \n Done

Answer: C

4. Which of the following is used to raise exceptions manually?

A) throw
B) raise
C) except
D) assert

Answer: B

5. What will happen here?

```
try:
    x = int("abc")
except (ValueError, TypeError):
    print("Error occurred")
```

A) Program crashes
B) Prints "Error occurred"
C) Prints nothing
D) Syntax error

Answer: B

6. What will be the output?

```
try:
    a = [1, 2, 3]
    print(a[5])
except IndexError as e:
    print("Error:", e)
```

A) Program crashes
B) Error: list index out of range
C) Error: invalid index
D) Nothing happens

Answer: B

Practice programs

1. Write a Python program to input two numbers and handle the following exceptions:(i) Division by zero(ii) Invalid input
2. Write a Python program to access the 5th element of a list. Handle the IndexError if the index is out of range.
3. Write a Python program that only accepts integers. If the input is not an integer, raise and handle a TypeError.
4. Write a Python program that asks for a person's age. Raise and handle a ValueError if the age entered is negative.
5. Write a Python program to convert a string into an integer. Handle the ValueError if the string is not a valid number.
6. Write a Python program to simulate a simple ATM withdrawal. If the withdrawal amount is greater than the balance, raise and handle an Exception saying "Insufficient Balance".