Python Fundamentals for Data Engineers

1. Variables and Data Types

Variables store data, and Python supports multiple data types like integers, floats, strings, lists, tuples, dictionaries, and sets.

Example:

```
# Variables and data types

name = "Madhu"         # String

age = 25               # Integer

height = 5.9           # Float

skills = ["Python", "SQL", "ETL"]  # List

print(f"My name is {name}, I am {age} years old, and I know {skills}.")
```

Case Scenario:

Store employee details (name, age, department) and retrieve them when needed.

2. Control Flow (if-else, loops)

Decision-making (if-else) and iteration (for, while) structures allow dynamic and repetitive task handling.

Example:

```
# Control flow

score = 85
```

```
if score >= 90:
    print("Grade: A")
elif score >= 75:
    print("Grade: B")
else:
    print("Grade: C")
```

Case Scenario:

Evaluate the performance of students and assign grades.

3. Functions

Reusable blocks of code that perform specific tasks, promoting modularity and reducing redundancy.

Example:

```
# Function example
def calculate_area(length, width):
    return length * width

area = calculate_area(5, 3)
print(f"The area of the rectangle is {area} square units.")
```

Case Scenario:

A function to calculate and return the total price of items in a shopping cart.

4. Error Handling

Mechanism to catch and handle exceptions, ensuring graceful handling of unexpected inputs.

Example:

```
try:
    num = int(input("Enter a number: "))
    print(f"Square of {num} is {num**2}.")
except ValueError:
    print("Please enter a valid integer.")
```

Case Scenario:

Validate user inputs in a form, such as ensuring age is numeric.

5. Basic Input and Output

Methods to get input from the user and display output.

Example:

```
# Input and output
name = input("Enter your name: ")
print(f"Hello, {name}!")
```

Case Scenario:

Ask a user for their credentials and confirm login.

6. Lists, Tuples, and Dictionaries

Data structures for organizing and processing collections of data efficiently.

Example:

```
# List
fruits = ["apple", "banana", "cherry"]
fruits.append("date")

# Dictionary
employee = {"name": "Madhu", "age": 25, "dept": "Engineering"}
print(f"Employee Name: {employee['name']}")
```

Case Scenario:

Store and update customer details in an e-commerce platform.

7. Loops (For and While)

Iterative structures to execute a block of code multiple times.

Example:

```
# Loop example
for i in range(1, 6):
    print(f"Table of 2: {2} x {i} = {2*i}")
```

Case Scenario:

Print a multiplication table or generate sequential file names.

8. String Operations

Operations like concatenation, slicing, and formatting for data manipulation.

Example:

```
# String operations

text = "Madhu learning Python"

print(text.upper())

print(text.split(" "))
```

Case Scenario:

Extract domain names from email addresses.