Python Practice Exercises - Chapter 1

Instructions

Complete each exercise by writing Python code. Run your solutions to verify they work correctly.

Exercise 1: Variable Practice

Create a program that collects and displays personal information.

Your Task:

```
# Create variables for:
# - Your name (string)
# - Your age (integer)
# - Your height in meters (float)
# - Whether you're a student (boolean)
# - Your favorite hobbies (list)

# Display all information in a formatted way
```

Expected Output:

```
=== Personal Information ===
Name: [Your Name]
Age: [Your Age] years old
Height: [Your Height] meters
Student Status: [True/False]
Hobbies: [hobby1, hobby2, hobby3]
```

Sample Solution:

```
name = "Alice Johnson"
age = 22
height = 1.68
is_student = True
hobbies = ["reading", "swimming", "coding"]

print("=== Personal Information ===")
print(f"Name: {name}")
print(f"Age: {age} years old")
print(f"Height: {height} meters")
print(f"Student Status: {is_student}")
print(f"Hobbies: {hobbies}")
```

Exercise 2: Basic Calculator

Create a simple calculator that performs basic operations.

Your Task:

```
# Get two numbers from user input
# Perform and display:
# - Addition
# - Subtraction
# - Multiplication
# - Division (handle division by zero)
# - Modulo (remainder)
# - Exponentiation
```

Expected Output:

```
Enter first number: 15
Enter second number: 4

=== Calculator Results ===
15 + 4 = 19
15 - 4 = 11
15 * 4 = 60
15 / 4 = 3.75
15 % 4 = 3
15 ** 4 = 50625
```

Exercise 3: String Manipulation

Practice working with strings and string methods.

Your Task:

```
# Create a program that:
# 1. Gets a sentence from user input
# 2. Displays string information (length, word count)
# 3. Shows the sentence in different formats
# 4. Demonstrates various string methods
```

Expected Features:

- Convert to uppercase and lowercase
- Count specific characters
- Replace words
- Check if sentence contains certain words

Sample Input: "Python programming is fun and powerful"

Expected Output:

```
Original: Python programming is fun and powerful
Length: 35 characters
Words: 6 words
Uppercase: PYTHON PROGRAMMING IS FUN AND POWERFUL
Lowercase: python programming is fun and powerful
Title Case: Python Programming Is Fun And Powerful
Count of 'a': 3
Contains 'Python': True
Replaced 'fun' with 'awesome': Python programming is awesome and powerful
```

Exercise 4: Temperature Converter

Create a temperature conversion program.

Your Task:

```
# Create a program that:
# 1. Gets temperature and unit from user
# 2. Converts between Celsius, Fahrenheit, and Kelvin
# 3. Displays results in all three units
# 4. Includes proper formatting and validation
```

Conversion Formulas:

- Celsius to Fahrenheit: F = (C × 9/5) + 32
- Celsius to Kelvin: K = C + 273.15
- Fahrenheit to Celsius: C = (F - 32) × 5/9
- Kelvin to Celsius: C = K - 273.15

Expected Output:

```
Enter temperature: 25
Enter unit (C/F/K): C

=== Temperature Conversion ===
25.0°C = 77.0°F = 298.15K
```

Exercise 5: Grade Calculator

Create a grade calculation and reporting system.

Your Task:

```
# Create a program that:
# 1. Gets student name and multiple test scores
# 2. Calculates average, highest, and lowest scores
# 3. Determines letter grade
# 4. Provides a formatted report
```

Grading Scale:

- A: 90-100
- B: 80-89
- C: 70-79
- D: 60-69
- F: Below 60

Expected Output:

```
Student Name: Alice Johnson
Enter scores separated by commas: 85, 92, 78, 96, 88

=== Grade Report ===
Student: Alice Johnson
Scores: [85.0, 92.0, 78.0, 96.0, 88.0]
Average: 87.8
Highest: 96.0
Lowest: 78.0
Letter Grade: B
Status: PASSED
```

Exercise 6: Text Analysis Tool

Create a comprehensive text analysis program.

Your Task:

```
# Create a program that analyzes a text and provides:
# 1. Character count (including and excluding spaces)
# 2. Word count
# 3. Sentence count (count periods, exclamation marks, question marks)
# 4. Most common character
# 5. Longest word
```

Sample Input: "Hello world! How are you today? I hope you're doing well."

Expected Output:

```
=== Text Analysis ===
Text: Hello world! How are you today? I hope you're doing well.
Characters (with spaces): 57
Characters (without spaces): 47
Words: 11
Sentences: 3
Most common character: 'o' (appears 6 times)
```

```
Longest word: "doing"
```

Exercise 7: Number Guessing Game Setup

Create the foundation for a number guessing game.

Your Task:

```
# Create a program that:
# 1. Generates a random number between 1 and 100
# 2. Gets user's guess
# 3. Provides feedback (too high, too low, correct)
# 4. Tracks number of attempts
# 5. Calculates score based on attempts
```

Note: For random number generation, you can use:

```
import random
secret_number = random.randint(1, 100)
```

Expected Interaction:

```
Welcome to the Number Guessing Game!
I'm thinking of a number between 1 and 100.

Enter your guess: 50
Too high! Try again.

Enter your guess: 25
Too low! Try again.

Enter your guess: 37
Congratulations! You guessed it in 3 attempts.
Your score: 97 points (100 - 3 attempts)
```

Exercise 8: Data Validation Practice

Create a program that validates different types of user input.

Your Task:

```
# Create functions that validate:
# 1. Email format (contains @ and .)
# 2. Phone number (10 digits)
# 3. Age (positive integer between 0 and 150)
# 4. Password strength (min 8 chars, contains letter and number)
```

Expected Functions:

```
def validate_email(email):
    # Return True if valid, False otherwise
    pass

def validate_phone(phone):
    # Return True if valid, False otherwise
    pass

def validate_age(age):
    # Return True if valid, False otherwise
    pass

def validate_password(password):
    # Return True if valid, False otherwise
    pass
```

Test Cases:

```
# Test your functions
print(validate_email("user@example.com"))    # True
print(validate_email("invalid-email"))       # False
print(validate_phone("1234567890"))          # True
print(validate_phone("123"))                 # False
print(validate_age("25"))                    # True
print(validate_age("200"))                   # False
print(validate_password("password123"))      # True
print(validate_password("weak"))             # False
```

Exercise 9: Shopping Receipt Generator

Create a shopping receipt calculator.

Your Task:

```
# Create a program that:
# 1. Gets item names and prices from user
# 2. Calculates subtotal, tax, and total
# 3. Generates a formatted receipt
# 4. Handles discounts and tips
```

Features to Include:

- Multiple items input
- Tax calculation (8.5%)
- Optional discount percentage
- Formatted receipt output

Expected Output:

```
=== SHOPPING RECEIPT ===
Date: [Current Date]

Items:
1. Apple          $2.50
2. Bread          $3.25
3. Milk           $4.99

Subtotal:         $10.74
Discount (10%):   -$1.07
Tax (8.5%):       $0.82
TOTAL:            $10.49

Thank you for shopping!
```

Exercise 10: Unit Converter

Create a comprehensive unit conversion tool.

Your Task:

```
# Create a program that converts between different units:
# 1. Length: meters, feet, inches, kilometers, miles
# 2. Weight: kilograms, pounds, ounces, grams
# 3. Temperature: Celsius, Fahrenheit, Kelvin
# 4. Time: seconds, minutes, hours, days
```

Conversion Factors:

```
# Length conversions (to meters)
length_to_meters = {
    "m": 1,
    "ft": 0.3048,
    "in": 0.0254,
    "km": 1000,
    "mi": 1609.34
}

# Weight conversions (to grams)
weight_to_grams = {
    "g": 1,
    "kg": 1000,
    "lb": 453.592,
    "oz": 28.3495
}
```

Expected Interface:

```
=== Unit Converter ===
Categories:
1. Length
2. Weight
3. Temperature
```

```
4. Time

Select category: 1
Available units: m, ft, in, km, mi
Enter value: 100
From unit: ft
To unit: m
Result: 100 ft = 30.48 m
```

Solutions Guide

Exercise 1 Solution:

```python
# Personal Information Program
name = "Alex Smith"
age = 20
height = 1.75
is_student = True
hobbies = ["photography", "hiking", "cooking"]

print("=== Personal Information ===")
print(f"Name: {name}")
print(f"Age: {age} years old")
print(f"Height: {height} meters")
print(f"Student Status: {is_student}")
print(f"Hobbies: {', '.join(hobbies)}")
```

Exercise 2 Solution:

```python
# Basic Calculator
try:
    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))

    print(f"\n=== Calculator Results ===")
    print(f"{num1} + {num2} = {num1 + num2}")
    print(f"{num1} - {num2} = {num1 - num2}")
    print(f"{num1} * {num2} = {num1 * num2}")

    if num2 != 0:
        print(f"{num1} / {num2} = {num1 / num2}")
        print(f"{num1} % {num2} = {num1 % num2}")
    else:
        print("Division by zero is not allowed")

    print(f"{num1} ** {num2} = {num1 ** num2}")

except ValueError:
    print("Please enter valid numbers")
```

Exercise 5 Solution:

```python
# Grade Calculator
student_name = input("Student Name: ")
scores_input = input("Enter scores separated by commas: ")

try:
    scores = [float(score.strip()) for score in scores_input.split(",")]

    average = sum(scores) / len(scores)
    highest = max(scores)
    lowest = min(scores)

    # Determine letter grade
    if average >= 90:
        letter_grade = "A"
    elif average >= 80:
        letter_grade = "B"
    elif average >= 70:
        letter_grade = "C"
    elif average >= 60:
        letter_grade = "D"
    else:
        letter_grade = "F"

    status = "PASSED" if average >= 60 else "FAILED"

    print(f"\n=== Grade Report ===")
    print(f"Student: {student_name}")
    print(f"Scores: {scores}")
    print(f"Average: {average:.1f}")
    print(f"Highest: {highest}")
    print(f"Lowest: {lowest}")
    print(f"Letter Grade: {letter_grade}")
    print(f"Status: {status}")

except ValueError:
    print("Please enter valid numeric scores")
```