1. The Smart Attendance Logger

> ***Key Skill:*** *Using* `set` *for automatic deduplication in a state-based loop.*

```
def mark_attendance():
    # Initialize an empty set - order doesn't matter, uniqueness does
    attendance_set = set()

    while True:
        name = input("Enter student name (or type 'STOP' to finish): ").strip()

        if name.upper() == "STOP":
            break

        if name: # Ensures empty inputs aren't added
            attendance_set.add(name.title())

    print(f"\n--- Attendance Report ---")
    print(f"Total Unique Students: {len(attendance_set)}")
    print(f"Attendees: {', '.join(attendance_set)}")

# mark_attendance()
```

2. The Restaurant Bill Splitter

> ***Key Skill:*** *Managing a list of floats and using default parameter values.*

```
def split_bill(items_prices, people, tax=0.05, tip=0.10):
    # Calculate totals
    subtotal = sum(items_prices)
    total_with_tax_tip = subtotal * (1 + tax + tip)

    # Calculate share per person
    share = total_with_tax_tip / people
    return round(share, 2)

# Example: bill = [25.50, 40.00, 15.75]
# print(f"Each person owes: ${split_bill(bill, 3)}")
```

3. The Digital Wardrobe Organizer

> ***Key Skill:*** *Independent checks using multiple* `if` *statements.*

```
wardrobe = ["Blue Jeans", "White Shirt", "Black Jacket"]

# Collecting 3 separate inputs
item1 = input("Outfit Item 1: ").title()
item2 = input("Outfit Item 2: ").title()
item3 = input("Outfit Item 3: ").title()
```

```
print("\n--- Availability Check ---")
# Three INDEPENDENT checks
if item1 not in wardrobe:
    print(f"❌ Warning: {item1} is missing!")

if item2 not in wardrobe:
    print(f"❌ Warning: {item2} is missing!")

if item3 not in wardrobe:
    print(f"❌ Warning: {item3} is missing!")
```

4. The E-Commerce Discount Engine

> ***Key Skill:*** *Mutually exclusive logic using* `if-elif-else`.

```
def apply_discount(total):
    if total > 500:
        discount_rate = 0.20 # 20%
    elif total > 200:
        discount_rate = 0.10 # 10%
    else:
        discount_rate = 0.0 # 0%

    discount_amount = total * discount_rate
    return total - discount_amount

# print(f"Final Total: ${apply_discount(350)}")
```

5. The Contact Book Search

> ***Key Skill:*** *Key-based lookup and handling "KeyError" with conditional checks.*

```
phonebook = {
    "Alice": "555-0101",
    "Bob": "555-0102",
    "Charlie": "555-0103",
    "David": "555-0104",
    "Eve": "555-0105"
}

name_query = input("Search for name: ").title()

if name_query in phonebook:
    print(f"✅ Found: {name_query}'s number is {phonebook[name_query]}")
else:
    print("⚠️ Error: Contact not found in phonebook.")
```

6. The Travel Itinerary Builder

> ***Key Skill:*** *Iteration with indexed tracking using* `enumerate`.

```
cities = ["Tokyo", "Paris", "New York", "London"]

print("\n--- 2026 Travel Itinerary ---")
for count, city in enumerate(cities, start=1):
    print(f"Stop {count}: {city}")
```

7. The ATM Pin Validator

> ***Key Skill:*** *While-loop control with attempt counters and early exit (* `break` *).*

```
def validate_pin():
    correct_pin = "1234"
    attempts = 0

    while attempts < 3:
        user_pin = input(f"Enter PIN ({3 - attempts} tries left): ")

        if user_pin == correct_pin:
            print("🔓 Access Granted. Welcome back!")
            return # Exit function entirely on success

        attempts += 1
        print("❌ Incorrect PIN.")

    print("🚫 Card Blocked. Please contact your bank.")

# validate_pin()
```

8. The Movie Ticket Age Checker

> ***Key Skill:*** *Retrieving data from an immutable* `tuple` *based on conditional indices.*

```
PRICES = (0, 10, 15) # Index: 0=Toddler, 1=Child, 2=Adult

age = int(input("Enter Age: "))

if age < 5:
    price = PRICES[0]
elif age < 18:
    price = PRICES[1]
else:
    price = PRICES[2]

print(f"Your ticket price is: ${price}")
```

9. The Grocery Inventory Update

> ***Key Skill:*** *Batch-modifying dictionary values in a loop.*

```python
inventory = {"Apples": 50, "Bananas": 20, "Oranges": 35}

# Shipment arrives: Add 10 to every stock level
for item in inventory:
    inventory[item] += 10

print("--- Inventory Updated (New Shipment) ---")
for fruit, stock in inventory.items():
    print(f"{fruit}: {stock} units")
```

10. The Flexible Event Invitation

> ***Key Skill:*** *Unpacking variable positional (`*args`) and keyword (`**kwargs`) arguments.*

```python
def create_invite(event_name, *guest_names, **details):
    print(f"\n--- {event_name.upper()} ---")

    print("Guest List:")
    for guest in guest_names:
        print(f" • {guest}")

    print("\nEvent Details:")
    for key, value in details.items():
        # Format "start_time" to "Start Time"
        formatted_key = key.replace('_', ' ').title()
        print(f" {formatted_key}: {value}")

# create_invite("Python Workshop", "Alice", "Bob",
#               Date="Jan 24, 2026", Location="Room 404", Time="10 AM")
```
