**Exercise 1: Simple Payroll Calculator**

**Objectives:**

* Understand basic Python syntax and data types.
* Learn to implement basic arithmetic operations in Python.

**Business Scenario:**

You are tasked with creating a simple payroll calculator for a small business. The calculator should take the number of hours worked and the hourly rate, then calculate the total pay for employees.

**Tasks:**

1. Create a new Python file named **payroll\_calculator.py**.
2. Define variables for the number of hours worked and the hourly rate.
3. Write a function **calculate\_pay(hours, rate)** that calculates the total pay.
4. Use the function to calculate the pay for an example employee who worked 40 hours at a rate of $15 per hour.
5. Print the total pay for the employee.

**Exercise 2: Order Processing System**

**Objectives:**

* Use control structures like if-else statements.
* Implement business logic based on conditions.

**Business Scenario:**

You are developing an order processing system for an online store. The system should apply a discount to orders over $100.

**Tasks:**

1. Create a new Python file named **order\_processing.py**.
2. Define a variable for the order amount.
3. Write a function **apply\_discount(order\_amount)** that applies a 10% discount if the order amount is greater than $100.
4. Calculate the final price for an example order amount of $150.
5. Print the final price after applying the discount.

**Exercise 3: Inventory Management System**

**Objectives:**

* Use loops and data structures to manage data.
* Learn to traverse and manipulate lists.

**Business Scenario:**

You are creating an inventory management system for a retail store. The system should keep track of item quantities and alert when an item is out of stock.

**Tasks:**

1. Create a new Python file named **inventory\_management.py**.
2. Define a list of items with their quantities (e.g., [('item1', 10), ('item2', 0), ('item3', 5)]).
3. Write a function **check\_inventory(inventory)** that prints a message if any item is out of stock.
4. Iterate over the list and use the function to check the inventory.
5. Print the results.

**Exercise 4: Customer Data Analysis**

**Objectives:**

* Use dictionaries to store and manipulate key-value pairs.
* Implement basic data retrieval and update operations.

**Business Scenario:**

You are analyzing customer data for a marketing campaign. The system should store customer names and their purchase amounts, and you need to update and retrieve this data as required.

**Tasks:**

1. Create a new Python file named **customer\_data.py**.
2. Define a dictionary to store customer names and their purchase amounts (e.g., {'Alice': 120, 'Bob': 75, 'Charlie': 90}).
3. Write a function **update\_purchase(customer\_data, name, amount)** that updates the purchase amount for a given customer.
4. Use the function to update the purchase amount for 'Bob' to 100.
5. Print the updated customer data.

**Exercise 5: Sales Report Generator**

**Objectives:**

* Use loops and conditional statements to generate reports.
* Learn to format and display data.

**Business Scenario:**

You are tasked with generating a sales report for the previous month. The report should list the total sales and highlight any sales over $500.

**Tasks:**

1. Create a new Python file named **sales\_report.py**.
2. Define a list of sales amounts (e.g., [200, 600, 150, 800, 300]).
3. Write a function **generate\_report(sales)** that iterates over the sales list and prints each sale amount. Highlight any sale over $500.
4. Call the function to generate a report for the example sales list.
5. Print the total sales for the month.