Dictionary Labs

Exercise 1: Using get()
Task: Create a dictionary named person with keys "name", "age", and "city". Then, write code that:

1. Retrieves the value for the key "name" using the get() method.
2. Attempts to retrieve the value for a key "country" using the get() method. If the key does not exist, the program should return the default value "USA". Print both outputs.

Exercise 2: Using pop()
Task: Create a dictionary called fruit_prices with keys "apple", "banana", and "cherry", and assign appropriate price values. Write code that:

1. Removes the key "banana" from the dictionary using pop() and prints its value.
2. Attempts to remove a non-existent key "orange" with a default message such as "Not Found", and prints the message. Finally, print the updated dictionary.

Exercise 3: Using clear()
Task: Create a dictionary named scores with keys for three students (for example, "Alice", "Bob", and "Charlie") with their respective numeric scores. Write code that clears all the items from the dictionary using the clear() method, and then prints the dictionary.

Exercise 4: Using values()
Task: Create a dictionary named book_info that holds information about a book with keys "title", "author", and "published". Write code to extract only the values from the dictionary using the values() method. Convert these values into a list and print them.
Expected Output: Book Information Values: ['1984', 'George Orwell', 1949]

Exercise 5: Using items()
Task: Build a dictionary named inventory with keys representing different fruits (e.g., "apples", "bananas", "oranges") and values representing their quantities. Write code to iterate over the dictionary using the items() method and print each fruit along with its quantity using formatted output. Then, convert the items into a list of tuples and print it.

Solutions

Exercise 1: Using get()

```
# Create the dictionary with basic person details
person = {
   "name": "Alice",
   "age": 30,
   "city": "New York"
}
# Retrieve the value of "name"
name = person.get("name")
print("Name:", name)  # Expected Output: Name: Alice
# Retrieve the value of "country", using "USA" as the default if key not found
country = person.get("country", "USA")
print("Country:", country)  # Expected Output: Country: USA
```

Exercise 2: Using pop()

```
# Create a dictionary with fruit prices
fruit_prices = {
   "apple": 0.99,
   "banana": 0.59,
   "cherry": 2.99
}
# Remove the key "banana" and print its price
price = fruit_prices.pop("banana")
print("Banana price removed:", price)  # Expected Output: Banana price removed: 0.59
# Try removing a non-existent key "orange" with a default value
price = fruit_prices.pop("orange", "Not Found")
print("Orange price:", price)  # Expected Output: Orange price: Not Found
# Print the updated dictionary
print("Updated fruit prices:", fruit_prices)
```

Exercise 3: Using clear()

```
# Create a dictionary with student scores
scores = {
```

```
    "Alice": 85,
    "Bob": 92,
    "Charlie": 78
}
# Clear all items from the dictionary
scores.clear()
# Output the cleared dictionary
print("Scores after clear():", scores)  # Expected Output: Scores after clear(): {}

Exercise 4: Using values()

# Create a dictionary with book details
book_info = {
    "title": "1984",
    "author": "George Orwell",
    "published": 1949
}
# Retrieve only the values in the dictionary
values = book_info.values()
# Convert the view to a list and print the values
print("Book Information Values:", list(values))
# Expected Output: Book Information Values: ['1984', 'George Orwell', 1949]

Exercise 5: Using items()

# Create a dictionary with fruit inventory
inventory = {
    "apples": 10,
    "bananas": 20,
    "oranges": 15
}
# Iterate through key-value pairs using items() and print each pair
for fruit, quantity in inventory.items():
    print(f"{fruit.title()}: {quantity}")
# Convert the items view to a list of tuples and print it
pairs = list(inventory.items())
print("Inventory pairs:", pairs)
# Expected Output: Inventory pairs: [('apples', 10), ('bananas', 20), ('oranges', 15)]
```