

Chapter 3: Model Solutions

Below, you'll find sample solutions to the lab exercises in the book.

Lab Exercises 3.1

1. Program to sum all elements in a given list:

```
my_list = [1, 2, 3, 4, 5]
total = sum(my_list)
print("Sum of elements:", total)
```

2. Program to count the number of even and odd numbers in a given list:

```
my_list = [1, 2, 3, 4, 5]
count_even = sum(1 for num in my_list if num % 2 == 0)
count_odd = len(my_list) - count_even
print("Even count:", count_even)
print("Odd count:", count_odd)
```

3. Program to check if a given list is sorted in ascending order:

```
my_list = [1, 2, 3, 4, 5]
sorted_check = my_list == sorted(my_list)
if sorted_check:
    print("The list is sorted in ascending order.")
else:
    print("The list is not sorted in ascending order.")
```

4. Program to reverse a given list:

```
my_list = [1, 2, 3, 4, 5]
reversed_list = my_list[::-1]
print("Reversed list:", reversed_list)
```

5. Program to find the frequency of each element in a given list:

```
my_list = [1, 2, 1, 3, 2, 4, 2, 5]
freq_dict = {}
```

```
for element in my_list:
    freq_dict[element] = freq_dict.get(element, 0) + 1
print("Frequency of elements:", freq_dict)
```

Lab Exercises 3.2

1. Program to perform set operations on two sets:

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}

union_set = set1.union(set2)
intersection_set = set1.intersection(set2)
difference_set = set1.difference(set2)

print("Union:", union_set)
print("Intersection:", intersection_set)
print("Difference:", difference_set)
```

2. Program to modify a tuple of integers:

```
my_tuple = (1, 2, 3, 4, 5)

modified_tuple = tuple(num * 2 if num % 2 == 0 else num + 1
for num in my_tuple)

print("Modified tuple:", modified_tuple)
```

3. Program to print names in alphabetical order:

```
my_tuple = ('Alice', 'Bob', 'Charlie', 'David')

sorted_names = tuple(sorted(my_tuple))

print("Names in alphabetical order:", sorted_names)
```

4. Program to create a set of unique characters from a string:

```
my_string = "hello world"

unique_chars = set(my_string)

print("Unique characters:", unique_chars)
```

5. Program to perform set operations on two sets:

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}

union_set = set1.union(set2)
intersection_set = set1.intersection(set2)
symmetric_diff_set = set1.symmetric_difference(set2)

print("Union:", union_set)
print("Intersection:", intersection_set)
print("Symmetric Difference:", symmetric_diff_set)
```

6. A Python program that takes a tuple of numbers and calculates the sum and average

```
my_tuple = (10, 20, 30, 40, 50)

# Calculate the sum of the numbers in the tuple
total = sum(my_tuple)

# Calculate the average of the numbers in the tuple
average = total / len(my_tuple)

print("Sum:", total)
print("Average:", average)
```

Lab Exercises 3.3

1. Program to create a dictionary representing a person's contact information:

```
contact_info = {}

contact_info['name'] = input("Enter name: ")
contact_info['email'] = input("Enter email: ")
contact_info['phone'] = input("Enter phone number: ")

print("Contact Information:", contact_info)
```

2. Program to simulate an English to Spanish dictionary:

```
dictionary = {
    'hello': 'hola',
    'goodbye': 'adiós',
    'thank you': 'gracias',
    'yes': 'sí',
    'no': 'no'
}

word = input("Enter an English word: ")

if word in dictionary:
    translation = dictionary[word]
    print("Spanish translation:", translation)
else:
    print("Word not found in dictionary.")
```

3. Track the sales of products in a store.

```
sales = {}

def add_sale():
    product = input("Enter the name of the product: ")
    price = float(input("Enter the price of the product: "))
```

```

    if product in sales:
        sales[product] += price
    else:
        sales[product] = price

    print("Sale added.")

def display_sales():
    print("Sales Information:")
    for product, amount in sales.items():
        print(product + ": $" + str(amount))

while True:
    print("\n=== Sales Tracking System ===")
    print("1. Add Sale")
    print("2. Display Sales")
    print("3. Exit")

    choice = int(input("Enter your choice (1-3): "))

    if choice == 1:
        add_sale()
    elif choice == 2:
        display_sales()
    elif choice == 3:
        break
    else:
        print("Invalid choice. Please try again.")

```

4. To merge two dictionaries, you can use the update() method.

```

dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd': 4}

```

```
dict1.update(dict2)
print(dict1)
```

5. Define a dictionary

```
my_dict = {'key1': 'value1',
           'key2': 'value2',
           'key3': 'value3'}
```

6. Access an item in a dictionary

```
my_dict = {'name': 'John', 'age': 30, 'city': 'New York'}
print(my_dict['name'])
print(my_dict['age'])
print(my_dict['city'])
```

7. Use the get() method

```
my_dict = {'name': 'John', 'age': 30}
print(my_dict.get('name'))
print(my_dict.get('city', 'Unknown'))
```

8. Use the pop() method

```
my_dict = {'name': 'John', 'age': 30}
age = my_dict.pop('age')
print(age)
print(my_dict)
```

9. use the update() method

```
dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd': 4}
dict1.update(dict2)
print(dict1)
```