

# Rajalakshmi Engineering College

Name: Mathu Methaa K  
Email: 240701306@rajalakshmi.edu.in  
Roll no: 2116240701306  
Phone: 6381399807  
Branch: REC  
Department: I CSE AH  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 4\_CY

Attempt : 1  
Total Mark : 40  
Marks Obtained : 40

### Section 1 : Coding

#### 1. Problem Statement

You are tasked with designing a shipping cost calculator program that calculates the shipping cost for packages based on their weight and destination. The program utilizes different shipping rates for domestic, international, and remote destinations. The rates for each destination type are provided as global constants.

Constant Values:

DOMESTIC\_RATE = 5.0

INTERNATIONAL\_RATE = 10.0

REMOTE\_RATE = 15.0

Function Signature: calculate\_shipping(weight, destination)

Formula: shipping cost = weight \* destination rate

### ***Input Format***

The first line of the input consists of a float representing the weight of the package.

The second line consists of a string representing the destinations(Domestic or International or Remote).

### ***Output Format***

The program outputs any one of the following:

1. If the input is valid and the destination is recognized, the output should consist of a single line stating the calculated shipping cost for the given weight and destination in the format: "Shipping cost to [destination] for a [weight] kg package: \$[calculated cost]" with two decimal places.
2. If the input weight is not a positive float, print "Invalid weight. Weight must be greater than 0."
3. If the input destination is not one of the valid options, print "Invalid destination."

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: 5.5

Domestic

Output: Shipping cost to Domestic for a 5.5 kg package: \$27.50

### ***Answer***

```
#
```

```
# You are using Python
```

```
# Constants
```

```
DOMESTIC_RATE = 5.0
```

```
INTERNATIONAL_RATE = 10.0
```

```
REMOTE_RATE = 15.0
```

```

# Function to calculate shipping cost
def calculate_shipping(weight, destination):
    if weight <= 0:
        print("Invalid weight. Weight must be greater than 0.")
        return None
    if destination == "Domestic":
        return weight * DOMESTIC_RATE
    elif destination == "International":
        return weight * INTERNATIONAL_RATE
    elif destination == "Remote":
        return weight * REMOTE_RATE
    else:
        print("Invalid destination.")
        return None

# Input
weight = float(input())
destination = input()

# Call function
shipping_cost = calculate_shipping(weight, destination)

if shipping_cost is not None:
    print(f"Shipping cost to {destination} for a {weight} kg package:
    ${shipping_cost:.2f}")

```

**Status :** Correct

**Marks :** 10/10

## 2. Problem Statement

Develop a text analysis tool that needs to count the occurrences of a specific substring within a given text string.

Write a function `count_substrings(text, substring)` that takes two inputs: the text string and the substring to be counted. The function should count how many times the substring appears in the text string and return the count.

Function Signature: `count_substrings(text, substring)`

### ***Input Format***

The first line of the input consists of a string representing the text.

The second line consists of a string representing the substring.

### ***Output Format***

The output should display a single line of output containing the count of occurrences of the substring in the text string.

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: programming is fun and programming is cool  
programming

Output: The substring 'programming' appears 2 times in the text.

### ***Answer***

# You are using Python

def count\_substrings(text, substring):

count = text.count(substring)

print(f"The substring '{substring}' appears {count} times in the text.")

text = input()

substring = input()

count\_substrings(text, substring)

**Status :** Correct

**Marks :** 10/10

### **3. Problem Statement**

Imagine you are tasked with developing a function for calculating the total

cost of an item after applying a sales tax. The sales tax rate is equal to 0.08 and it is defined as a global variable.

The function should accept the cost of the item as a parameter, calculate the tax amount, and return the total cost.

Additionally, the program should display the item cost, sales tax rate, and total cost to the user.

Function Signature: `total_cost(item_cost)`

### ***Input Format***

The input consists of a single line containing a positive floating-point number representing the cost of the item.

### ***Output Format***

The output consists of three lines:

"Item Cost:" followed by the cost of the item formatted to two decimal places.

"Sales Tax Rate:" followed by the sales tax rate in percentage.

"Total Cost:" followed by the calculated total cost after applying the sales tax, formatted to two decimal places.

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 50.00

Output: Item Cost: \$50.00

Sales Tax Rate: 8.0%

Total Cost: \$54.00

### ***Answer***

#

# Global constant

SALES\_TAX\_RATE = 0.08

```
def total_cost(item_cost):  
    # Calculate the tax amount  
    tax = item_cost * SALES_TAX_RATE  
    # Return the total cost  
    return item_cost + tax
```

```
# Header Snippet  
item_cost = float(input())
```

```
# Footer Snippet
```

```
total_cost = total_cost(item_cost)  
print(f"Item Cost: ${item_cost:.2f}")  
print(f"Sales Tax Rate: {SALES_TAX_RATE * 100}%")  
print(f"Total Cost: ${total_cost:.2f}")
```

**Status :** Correct

**Marks :** 10/10

#### 4. Problem Statement

Amrita is developing a password strength checker for her website. She wants the checker to consider the length and the diversity of characters used in the password. A strong password should be long and include a mix of character types: uppercase, lowercase, digits, and special symbols.

She also wants the feedback to be user-friendly, so she wants to include the actual password in the output. Help Amrita finish this password checker using Python's built-in string methods.

Character Types Considered:

Lowercase letters (a-z) Uppercase letters (A-Z) Digits (0-9) Special characters (from string.punctuation, e.g. @, !, #, \$)

##### ***Input Format***

The input consists of a single string representing the user's password.

##### ***Output Format***

The program prints the strength of the password in this format:

If the password length < 6 characters or fewer than 2 of the 4 character types, the output prints "<password> is Weak"

If password length  $\geq 6$  and at least 2 different character types, the output prints "<password> is Moderate"

If Password length  $\geq 10$  and all 4 character types present, the output prints "<password> is Strong"

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: password123

Output: password123 is Moderate

### **Answer**

```
# You are using Python
password = input()
```

```
has_lower = False
has_upper = False
has_digit = False
has_special = False
```

```
for char in password:
    if char.islower():
        has_lower = True
    elif char.isupper():
        has_upper = True
    elif char.isdigit():
        has_digit = True
    else:
        has_special = True
```

```
types_count = has_lower + has_upper + has_digit + has_special
length = len(password)
```

```
if length >= 10 and types_count == 4:
```

```
print(f"{password} is Strong")  
elif length >= 6 and types_count >= 2:  
    print(f"{password} is Moderate")  
else:  
    print(f"{password} is Weak")
```

**Status :** Correct

**Marks :** 10/10