Rajalakshmi Engineering College

Name: MITHULESH J
Email: 240701313@rajalakshmi.edu.in
Roll no: 240701313
Phone: 8056467713
Branch: REC
Department: I CSE FC
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

#

```python
# You are using Python
def calculate_shipping(weight, destination):

    if weight <= 0:
        return

    d={"domestic":5.0,"international":10.0,"remote":15.0}
```

```
    if(destination.lower() not in d.keys()):
        return
    return weight*d[destination.lower()]
weight = float(input())
destination = input().strip()
shipping_cost=calculate_shipping(weight, destination)
if shipping_cost is None:
    if weight <=0:
        print("Invalid weight. Weight must be greater than 0.")
    else:
        print("Invalid 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

Arjun is working on a mathematical tool to manipulate lists of numbers. He
needs a program that reads a list of integers and generates two lists: one
containing the squares of the input numbers, and another containing the
cubes. Arjun wants to use lambda functions for both tasks.

Write a program that computes the square and cube of each number in the
input list using lambda functions.

Input Format

The input consists of a single line of space-separated integers representing the
list of input numbers.

Output Format

The first line contains a list of the squared values of the input numbers.

The second line contains a list of the cubed values of the input numbers.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 2 3
Output: [1, 4, 9]
[1, 8, 27]

Answer

```python
# You are using Python
def compute_squares_and_cubes(numbers):
    squares = list(map(lambda x: x**2, numbers))
    cubes = list(map(lambda x: x**3, numbers))
    print(squares)
    print(cubes)
numbers = list(map(int, input().split()))
compute_squares_and_cubes(numbers)
```

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

```
#

SALES_TAX_RATE = 0.08

def total_cost(item_cost):
    tax_amount = item_cost * SALES_TAX_RATE
    total_price = item_cost + tax_amount
    return total_price
item_cost = float(input())

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

Implement a program for a retail store that needs to find the highest even price in a list of product prices. Your goal is to efficiently determine the maximum even price from a series of product prices. Utilize the max() inbuilt function in the program.

For example, if the prices are 10 15 24 8 37 16, the even prices are 10 24 8 16. So, the maximum even price is 24.

Input Format

The input consists of a series of product prices separated by a space.

The prices should be entered as a space-separated string of numbers.

Output Format

If there are even prices in the input, the output prints "The maximum even price is: " followed by the maximum even price.

If there are no even prices in the input, the output prints "No even prices were found".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10 15 24 8 37 16
Output: The maximum even price is: 24

Answer

```python
# You are using Python
def find_max_even_price(prices):
    even_prices = list(filter(lambda x: x % 2 == 0, prices))
    if even_prices:
        print(f"The maximum even price is: {max(even_prices)}")
    else:
        print("No even prices were found")
```

```python
prices = list(map(int, input().split()))
find_max_even_price(prices)
```

Status : Correct   *Marks : 10/10*