



ಭಾರತೀಯ ಮಾಹಿತಿ ತಂತ್ರಜ್ಞಾನ ಸಂಸ್ಥೆ ರಾಯಚೂರು
भारतीय सूचना प्रौद्योगिकी संस्थान रायचूर
Indian Institute of Information Technology Raichur

LAB - 3

Introduction to OOP

Submission Guidelines

- Ensure your system is in 'No Aeroplane Mode'.
- No Taskbar should be open.
- Create a new folder named **LAB-2**.
- Inside **LAB-2**, create two question files named in the following format:
[RollNumber]_[LabName]_[QuestionNumber]
(e.g., **12345_MatrixLab_Q1.cpp** and **12345_MatrixLab_Q2.cpp**)

Lab Timing and Submission

- Lab Time: **6:00 PM - 8:00 PM**
- Submission Deadline: **8:00 PM - 8:05 PM** (Submit on Classroom)
- No Extensions: Late submissions will not be accepted.
- Viva: **8:05 PM - 8:30 PM** (Marks will be assigned based on viva performance)

Question 1:- (100 points)

Videha's Ice Cream Adventure

Problem Description

Videha is a young ice cream fan who visits a trendy parlor where customers can set their own prices for ice cream flavors and toppings. The parlor offers three flavors and three toppings, but the cost for each is not fixed—it is provided by the user at runtime. To build the perfect dessert, the parlor uses an object-oriented system with both **single** and **multiple** inheritance.

Class Structure :--

1. IceCream (Base Class):

This class represents a basic ice cream and serves as the foundation.

2. Flavor Classes (Derived via Single Inheritance):

- The available flavors are:
 - Vanilla
 - ButterScotch
 - Chocolate

3. The cost for each flavor is provided by the user through a dictionary

(e.g., {"Vanilla":50, "ButterScotch": 60, "Chocolate": 70}).

4. ToppingMixin (Mixin Class):

- This mixin adds the functionality for toppings. The available toppings are:
 - Nuts
 - Sprinkles
 - Cherries

5. Their costs are given by the user

(e.g., {"Nuts": 10, "Sprinkles": 5, "Cherries": 15}).

6. Order Class (Multiple Inheritance):

This class combines a flavor (from the single inheritance hierarchy) with the **ToppingMixin** to create the final ice cream order. It calculates the total cost as the sum of the flavor's price and the prices of any added toppings.

Constraints:

$1 \leq |\text{order}| \leq 50$

(The order string has at least one character and at most 50 characters.)

`order` contains space-separated words representing a flavor followed by zero or more toppings.

The number of words in `order` is between 1 and 4.

$1 \leq |\text{flavor_costs}| \leq 3$

(There are exactly 3 predefined flavors: "Vanilla", "ButterScotch", and "Chocolate".)

$1 \leq |\text{topping_costs}| \leq 3$

(There are exactly 3 predefined toppings: "Nuts", "Sprinkles", and "Cherries".)

$1 \leq \text{cost} \leq 1000$

(Each flavor or topping cost is between 1 and 1000, inclusive.)

All words in `order` are case-sensitive. "Vanilla" is valid, but "vanilla" is invalid.

The order must contain exactly **one flavor** (first word).

If an invalid flavor or topping is encountered, return -1.

Input Constraints:

User enters prices for 3 flavors and 3 toppings.

The user selects one flavor and optionally one topping (or "None" for no topping).

If an invalid flavor or topping is entered, the program should display an error and exit.

Question 2:-(100 points)

Performing arithmetic operations on matrices

Perform the basic operations(+, -, *) on **matrices** given by the user.

You need to perform the above operations in both **public** and **private inheritance** methods and the results must be shown in different classes by printing each class name.

Input : Input of matrices from the user

Output : arithmetic operations(+,-,*) on those two matrices.

Example input :-

Input :

A = 1 2 B = 5 6
 3 4 7 8

Output :

Public inheritance

addition

A + B = 6 8
 10 12



Subtraction

$$A - B = -4 -4$$

Multiplication

$$A * B = 19 \ 22$$
$$43 \ 50$$

Private inheritance

addition

$$A + B = 6 \ 8$$
$$10 \ 12$$

Subtraction

$$A - B = -4 -4$$

Multiplication

$$A * B = 19 \ 22$$
$$43 \ 50$$

Note :- Here results from addition, subtraction and multiplication are from **different derived classes** not from the **same derived class**.

■ ■ ■