SDM College of Engineering and Technology

Dhavalagiri, Dharwad-580 002. Karnataka State. India.
Email: principal@sdmcet.ac.in, cse.sdmcet@gmail.com
Ph: 0836-2447465/ 2448327 Fax: 0836-2464638 Website: sdmcet.ac.in

Department of COMPUTER SCIENCE AND ENGINEERING

MINOR REPORT

[22UHUC500- SOFTWARE ENGINEERING AND PROJECT MANAGEMENT]

Even Semester: Sep-Jan 2024

Course Teacher: Dr. U.P.Kulkarni

2024- 2025

Submitted by
By

Mr. Rohit M Halappanavar
2SD22CS074
5th Semester B division

Table of Contents

A-1:Write a C program to show that C programming language supports only Call By Value.

Business Scenario: **Sales Commission Calculation**.

Theory: Call by Value means that when a function is called, the actual arguments are passed to the function as copies. The function works with the copied values and any modifications made to these copies do not affect the original values.

Here's an explanation of why C only supports Call by Value:

1. Memory and Control:

- **Call by Value** ensures that when a function is called, the actual arguments are copied into the function's parameters. This allows the function to operate on the copies, not on the original variables.
- In C, developers are given direct control over memory, including how data is passed around in a program. By only supporting Call by Value, the language avoids the complexity and hidden side effects that could arise if variables were directly passed by reference.

2. Simplicity of Implementation:

- The mechanism for Call by Value is straightforward: the value is copied into a new memory location for the function to use. This is simple for both the programmer and the underlying system.

3. Safety and Predictability:

- **Call by Value** inherently prevents a function from modifying the caller's data. This makes functions safer because the programmer doesn't have to worry about the internal logic of the function altering external data unintentionally.
- If a function could modify the original variables automatically (as in Call by Reference), it would be harder to track down bugs where data is being changed unexpectedly, especially in large or complex programs.

Design:

I have used a sales commission calculation example to demonstrate why C Language supports only Call By Value, Commission is given only to them who have performed excellent in sales as there are many employees in sales business so calculating commission of each employee is a tedious task and it can be useful if we make a function named calculateCommission, We can call

this function by passing values that are UNITS sold and PRICE PER UNIT as it returns the calculated value. Now after returning calculated value we no more need the basic components for calculation of commission .

Therefore, sending a Copy of Values is way better than Passing the values by References.

Program:

```
#include <stdio.h>
// Function to calculate commission using Call by Value
void calculateCommission(int unitsSold, float pricePerUnit)
{
   float commission = unitsSold * pricePerUnit * 0.05; // 5% commission
   printf("Commission: %.2f\n", commission);
}
void main()
{
   int units = 100;
   float price = 20.50;
   // Call by Value: The values of `units` and `price` are passed as copies
   calculateCommission(units, price);
   // Original values remain unchanged
   printf("Units sold: %d\n", units);   // Still 100
   printf("Price per unit: $%.2f\n", price);  // Still 20.50
}
```

Sample input and output:

Here Input is pre-defined that are Units = 100 and Price = 20.50

Output:

Commission: 102.50

Units sold: 100

Price per unit: $20.50

References:

1) https://www.quora.com
2) Ansi c Textbook