

C++ DAY 4 Programs

1) Student Mark Manipulation

The screenshot shows a code editor with a sidebar containing file icons for main.cpp, JS, TS, and GO. The main area displays the following C++ code:

```
main.cpp
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 class Student
6 {
7     int rno;
8     char name[30];
9     int m1, m2;
10    int total;
11    float avg;
12    char result[5];
13
14 public:
15     void getdata();
16     void process();
17     void putdata();
18 };
19
20 void Student::getdata()
21 {
22     cout << "Enter Roll No: ";
23     cin >> rno;
24
25     cin.ignore(); // ⚡ VERY IMPORTANT (clears buffer)
}
```

The output window shows the execution of the program:

```
Enter number of students: 2
Enter details of Student 1
Enter Roll No: 1
Enter Name: Manoj
Enter Marks 1: 98
Enter Marks 2: 90

Enter details of Student 2
Enter Roll No: 2
Enter Name: MohanRaj
Enter Marks 1: 98
Enter Marks 2: 99

===== STUDENT DETAILS =====

Student 1
Roll No : 1
Name   : Manoj
Marks 1 : 98
Marks 2 : 90
Total  : 188
Average : 94
Result  : Pass
```

2) Case Study 2: Employee Salary Management:

An organization needs a program to manage salary details of its employees.

Requirements / Questions:

Create an Employee class with empld, name, basic pay, HRA, and DA.

Use an array of Employee objects to store salary details.

Implement member functions to:

Calculate gross salary

Display pay slip

Display employees whose gross salary is above a given value.

Find the total salary expenditure of the company.

The screenshot shows a code editor with a sidebar containing file icons for main.cpp, JS, TS, and GO. The main area displays the following C++ code:

```
main.cpp
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 class Employee
6 {
7     int empId;
8     char name[30];
9     float basic, hra, da;
10    float gross;
11
12 public:
13     void getdata();
14     void calculate();
15     void display();
16     float getGross();
17 };
18
19 void Employee::getdata()
20 {
21     cout << "Enter Employee ID: ";
22     cin >> empId;
23     cin.ignore();
24     cout << "Enter Employee Name: ";
25     cin.getline(name, 30);
}
```

The output window shows the execution of the program:

```
Enter DA: 887.989
Enter details of Employee 3
Enter Employee ID: Vishwa
Enter Employee Name: Enter Basic Pay: Enter HRA: Enter DA:
Enter details of Employee 4
Enter Employee ID: Enter Employee Name: Enter Basic Pay: Enter HRA:
Enter DA:
===== PAY SLIPS =====

Employee 1
Employee ID : 77767
Name       : Manoj
Basic Pay  : 4e+08
HRA        : 908979
DA          : 653534
Gross Salary : 4.01563e+08

Employee 2
Employee ID : 363753
Name       : Mohan
Basic Pay  : 45555.9
HRA        : 8676.06
DA          : 887.989
Gross Salary : 55120
```

3) Case Study 3: Library Book Record System

A library wants to maintain book information.

Requirements / Questions:

Create a Book class with bookId, title, author, and price.

Store records using an array of Book objects.

Write member functions to:

Accept book details

Display all books

Search for a book by title.

Display books whose price is below a specified amount.

The screenshot shows a code editor interface with a sidebar of file icons. The main area displays the code for `main.cpp`, which includes the definition of a `Book` class with methods for getting data, displaying it, searching by title, and getting price. It also includes a `getdata()` method for the `Book` class and a `getdata()` method for the global scope. The output window shows the execution of the program, prompting for the number of books (3), then for details of three books, and finally displaying the details of the first book.

```
main.cpp
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 class Book
6 {
7     int bookId;
8     char title[30];
9     char author[30];
10    float price;
11
12 public:
13    void getdata();
14    void display();
15    int searchByTitle(char t[]);
16    float getPrice();
17 };
18
19 void Book::getdata()
20 {
21     cout << "Enter Book ID: ";
22     cin >> bookId;
23
24     cin.ignore();
25 }
```

Output

```
Enter number of books: 3
Enter details of Book 1
Enter Book ID: 1
Enter Book Title: StrangerThings
Enter Author Name: Duffer Brothers
Enter Price: 4354546467656.877867

Enter details of Book 2
Enter Book ID: Spider Man
Enter Book Title: Enter Author Name: Enter Price:
Enter details of Book 3
Enter Book ID: Enter Book Title: Enter Author Name: Enter Price:
===== BOOK DETAILS =====

Book 1
Book ID : 1
Title   : StrangerThings
Author  : Duffer Brothers
Price   : 4.35455e+12

Book 2
Book ID : 0
Title   :
Author  :
```