

# Object Oriented Programming (BSDSF24)

## Programming Assignment 03

**Date:** 10-may-2025

**Due on:** 14-05-25

03:00 PM

---

### Task 1

You are provided with a cm.cpp file with code for Cash Memo case study. The code has several classes and main logic in the same file. You are required to transform the single file coding to multiple file coding. The other requirement is to demonstrate the aggregation of the Date object in the Cashmemo object, and composition of the Saleline objects in the Cashmemo. Lastly, you have to create a make script to build the multifile file coding. For submission, you have to submit all .cpp file, all .h files and makefile, in a zip file (don't submit .rar or .7z or other archives).

### Task 2

Create an class employee with data members as name (string), department (string), scale (int), another class officer inheriting employee class with its data member monthly\_salary (int), and a class daily\_wager inheriting employee class with its data members daily\_wage (int) and absent\_count. With other required functions and operations, add a member function take\_home\_income in both inherited classes, and as a virtual function in employee class (whose dummy implementation is just throw 0 and return 0). For officers take home income is rounded to integer and 90% of the monthly salary (10% is fixed tax) and for daily wagers, it is according to the days they worked, with no tax. Demonstrate the working of classes by creating array of pointers to several employee objects, pointing to some employee objects, some officer objects and some daily\_wager objects. Note: you must use only parameterized constructors in all three classes and avoid dangling pointers and memory leaks.

### Task 3

Transform the task 2 code to create multiple file coding, also its makefile. Also, add a payroll class that is responsible for managing the list of employees instead of main logic, obviously through employee pointers. The main logic in the case is like:

```
Payroll p;           // create a payroll object with no employees
p.loadEmployees();    // somehow load employees into array of pointers
p.printSalaries();     // print the names and take home income of all employees
```

In the 3<sup>rd</sup> task, for payroll class, the Big 5 should be implemented, as it hold pointers as data member.