

CSE 108 (Object Oriented Programming Sessional)

July 2025 Term

Offline 2 on C++: Code before you Vote



shutterstock.com - 2389787513

Objective

The objective of this lab assignment is to assess students' knowledge of Object-Oriented Programming in C++, with emphasis on Dynamic allocation of objects, Function Overloading: Overloading Constructor functions, Copy Constructors, passing objects to and returning objects from functions. Students are required to complete a partially implemented program so that its output exactly matches the given reference solution.

Problem Specification

As the upcoming national election in Bangladesh approaches, political parties must manage their members, nominate candidates for constituencies, cancel nominations when necessary, and sometimes even form new parties. In this lab, you will simulate such a fictional election management system using C++ classes.

You are given a skeletal file named *your_student_no.cpp*. This file contains two classes.

1. *PartyMember* – represents an individual party member
2. *PoliticalParty* – represents a political party that manages its members

The functions in these two classes are left unimplemented. You are required to perform the tasks below.

- Complete these two classes such that the output generated by your code matches the expected output for the given main function. The expected output is appended at the end of the above file in a comment block.

- Specific tasks to be performed are described in the comments of the given file. Follow those instructions during your implementation.
- Observe carefully the given ‘main’ function and the ‘expected’ output to determine the behavior of the functions.
- Add input validations in functions with arguments in all applicable cases.
- Optimize memory usage. Allocate objects only when required and free them when those are out of scope.
- Reuse code as much as possible.

Additionally, you must adhere to the restrictions below.

- Do not use STL containers (vector, map, etc.)
- Do not modify/delete existing data members for the given classes. However, you can add new members to the class PoliticalParty as required.
- Do not change the names, return types, or parameters of the given function.
- Do not modify *main()*. You may add to/delete from code the main function for your test purpose. But make sure all your modifications in the ‘main’ function are removed before submission.
- Do not add extra output statements.
- Follow the given class structure strictly.

Submission Guideline

Rename your file with your **7-digit** student number and upload this single .cpp file to the assigned Moodle link. Do not upload any zip file. The submission deadline is **Friday, 09 January 2026, 11:59 PM.**

For any further relevant query, engage in the relevant Moodle discussion thread.