

IIT Ropar

CSL 201 Data Structures

Semester 1, AY 2017/18

Lab Assignment1 - 40 marks

Due on 10th September, 11:59 PM

Objective

To understand and use various advanced features of C++ programming language and OOP concepts.

Instructions

1. You are to use C++ programming language to complete the assignment.
2. Provide a Makefile to compile your final code.
3. All function and class declarations should be in “.h” files while the definitions should be in “.cpp” files. It is recommended that you have one header file and one cpp file for each class.
4. This is an individual assignment. You can have high level discussions with other students, though.
5. Include a “Readme.txt” file on how to compile and run the code.
6. Upload your submission to moodle by the due date and time.
7. There will be 10% penalty for late submission for first three days. After that, your assignment will not be evaluated.
8. If any student asks for deadline extension, he or she will get 5% penalty.

Program Description

In this assignment you need to implement a vehicle management system for a transport company. The company has three types of vehicles: cars, trucks, and buses. A car has the following basic attributes: color, mileage, top speed, and the number of gears. All attributes are given as integers.

1. Write a program for an online vehicle management system. The program should have capability to add new vehicles to the system.

2. Read the car details from the file "car.txt" and add to the system using the add functionality. Maintain the vehicle details in a linked list. Do not use STL, implement your own linked list.
3. Extend the system to add trucks, which have the following attributes: color, mileage, top speed, and the number of gears, loading capacity in weight (kg) and loading capacity in volume (liters). Read the truck details from the file "truck.txt" and add to the system.
4. Further extend the system to have bus with the following attributes: color, mileage, top speed, the number of gears, and number of seats. Read the details from the file "bus.txt" and add to the system.

Functionalities

The developed system should implement the following functionalities.

1. The program should be able to add cars, trucks, or busses to the system dynamically. User will enter the type of vehicle followed by its details. If the vehicle with the same name already exists, you program should give an error.
2. The program should be able to remove a car, truck, or bus from the system. User will enter the name of the vehicle to be removed.
3. A watermelon needs 30 units of the area in a truck. The program should give the minimum number of trucks required to accommodate given number of watermelons. User will enter the number of watermelon and output should be the minimum number of trucks required along with the list of trucks.
4. The program should be able to find the smallest number of vehicles (buses and cars) needed to accommodate the given number of people. Use seats information of buses and assume each car can accommodate 5 passengers. User will enter the number of person and output should be the list of vehicles required.
5. Mechanism to book or un-book a car, truck, and bus. User will enter the type of vehicle. The program should display the details of the booked vehicle.
6. Program should be able to print all details of the given car, truck, and bus. User will enter the name of the vehicle.

No need to implement a separate GUI. You can display these choices in terminal itself and let the user make a choice. The user can be asked to enter subsequent details depending on the choice.

Overall, the students should use the following C++ features at least once in the code with proper justification:

- Function overloading

- Operator overloading
- Constructors and destructors
- Inheritance
- Virtual functions
- Dynamic memory allocation
- Function templates
- Access specifiers

You are allowed to add new functionalities to the system to use these features. Please mention new additions during the assignment demo.

Assessment

- The marks division is as follows: 15 marks for features, 15 marks for functionalities, 10 marks for code organization and clarity.
- You will be assessed individually in the lab session. You'll have approximately 10 minutes Q&A session with Dr. Mukesh or one of the TAs (Amit, Rohit, Pratibha).
- You will have to run the code directly from your Moodle submission, you won't be allowed to bring an updated version to the lab.
- After you run the code, we will ask you questions about the code. Be prepared to answer the questions.