

Assignment no 4

Deadline : 5th April, 2018

Problem 1:

(Tic-Tac-Toe) Write a program that allows a player to play the tic-tac-toe game with computer. Your program must contain the class `ticTacToe` to implement a `ticTacToe` object. Include a 3-by-3 two-dimensional array, as a private member variable, to create the board. If needed, include additional member variables. Some of the operations on a `ticTacToe` object are printing the current board, getting a move, checking if a move is valid, and determining the winner after each move. Add additional operations as needed.

Problem 2:

Let's assume you make a team of developers and that team is going to make a new social network.

Every user has a profile. That profile includes the following:

1. User Name
 2. User Age
 3. User ID (Unique for each user on the network)
 4. Friends List (containing the User IDs of the other users of the same social network which are friends with this user)
- User IDs are predefined, whenever you create a new user, assign any of those available IDs and keep record which IDs are taken so that next time you do not assign taken ID to new user.
 - Friend list of a user can only contain the user IDs which are taken at the creation of his/her profile.
 - Write a function to take inputs for a user profile.
 - Write another function that takes two different users as argument and counts the number of mutual friends between them using the search techniques you have studied in your CP course. This function should return how many mutual friends were found.

Problem 3:

Suppose you want to create a class called employee with the following data:

- Id (assigned when employee joins the organization and not changed afterwards)
- Name (has maximum 20 characters)
- Total hours (Total hours spent at work during the week.)
- Salary (Can change with time)
- Organization code (fixed and same for all employees and can never change)
- Home address (can change)
- Organization address (same for all employees but can change with time if the office moves elsewhere for expansion)
- Max hours (should store the maximum hours spent at work during the week by any employee who ever existed. Same for all employees but can change with time)
- Organization city code (same for all employees and fixed and never changes)

Write the class declaration for employee after deciding which members are const, static or static const.

Create a function to create file against the name of each employee and store the corresponding data in it. Eg: Ali.txt.

Create a function in which the user can then search the record of any employee by its ID.

Add the constructor and copy constructor to the employee class. Make sure you initialize the const members properly. If needed add private helper members to the class to copy const data members. Also, make sure that max hours are maintained properly when new employee is created.

Problem 4:

In ocean navigation, locations are measured in degrees and minutes of latitude and longitude. Thus if you're lying off the mouth of Papeete Harbor in Tahiti, your location is 149 degrees 34.8 minutes west longitude, and 17 degrees 31.5 minutes south latitude. This is written as 149°34.8' W, 17°31.5' S. There are 60 minutes in a degree. (An older system also divided a minute into 60 seconds, but the modern approach is to use decimal minutes instead.) Longitude is measured from 0 to 180 degrees, east or west from Greenwich, England, to the international dateline in the Pacific. Latitude is measured from 0 to 90 degrees, north or south from the equator to the poles.

Create a class angle that includes three member variables: an int for degrees, a float for minutes, and a char for the direction letter (N, S, E, or W). This class can hold either a latitude variable or a longitude variable. Write one member function to obtain an angle value (in degrees and minutes) and a direction from the user, and a second to display the angle value in 179°59.9' E format. Also write a three-argument constructor. Write a main() program that displays an angle initialized with the constructor, and then, within a loop, allows the user to input any angle value, and then displays the value. You can use the hex character constant '\xF8', which usually prints a degree (°) symbol.

Problem 5:

A matrix is a two-dimensional array. Create a class matrix that provides the same safety feature as the array class that is, it checks to be sure no array index is out of bounds. Make the member data in the matrix class of size given by user. A constructor should allow the programmer to specify the actual dimensions of the matrix. The member functions that access data in the matrix will now need two index numbers: one for each dimension of the array. Here's what a fragment of a main() program that operates on such a class might look like:

```
matrix m1(3, 4);    // define a matrix object

int temp = 12345;    // define an int value

m1.putel(7, 4, temp); // insert value of temp into matrix at 7,4

temp = m1.getel(7, 4); // obtain value from matrix at 7,4
```

Problem 6: (Public, Protected, Private Inheritance)

Write a class Student having Name, Roll no, CNIC as its attributes and name being a public data member. Inherit Undergraduate class from students having Semester no, CGPA and StudentID as its attributes, with Roll no. not being inherited from Student class and Name not being a public data member any more. Inherit Freshman, Sophomore, Junior and Senior classes from Undergraduate class in a manner that the attributes of Undergraduate class is not inherited further if we inherit further classes from Sophomore and Junior class.

Make default and parametrized constructors of all classes and implement all your work in main() function.

Problem 7: (Constructors, Destructors, Overriding)

Write a class Employee having the following attributes:

1. Employee name
2. Date of birth
3. Date of joining
4. CNIC

Inherit the following classes from Employee:

Admin:

This class has the following attributes:

1. Years of experience
2. Current Pay Grade

Accounts:

This class has the following attributes:

1. Salary
2. Office no

Faculty:

This class has the following attributes:

1. Highest academic degree achieved
2. Number of published research papers
3. Field of specialization

Make default constructor and parametrized constructors of all classes. Also, make destructors of all classes, which print the message "I am the destructor of <Class Name> class".

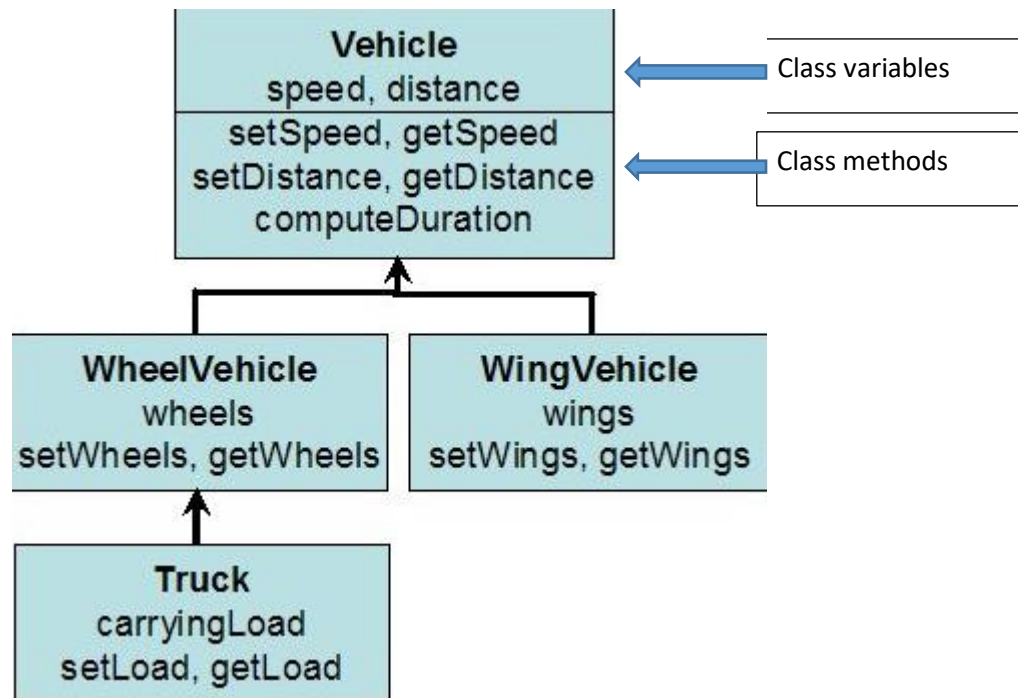
Write setter and display functions for all classes. You have to make the same function name in all classes. Explain in comments the following:

- Why cannot we call the functions of base class directly through derive class object?
- Is there any method to call functions of base class from derived class object?

Implement your work in main function by creating instances of all classes and by calling functions.

Problem 8: (Multilevel Inheritance)

Design the following classes with given variables and methods using inheritance.



Note: In Main, use the object of truck class to access all the data and functions of all inherited classes .And Create the object of wing Vehicle to access all the functions of Vehicle. Explain your work with proper output.

Problem 9: (Diamond Ambiguity)

Write a class Person having Name, Age, Gender as its attributes. It has a member function Func() which prints "I am function of Person class". Derive classes Businessman having companyName, earning as its attributes, and Teacher having experience, salary as its attributes, from Person. Derive a class Employed from both Businessman and Teacher, which cannot further inherit any of its attributes. Make default constructors of all classes which tells to whom class do they belongs to.

In main function, create an object of Employed class and try to call Func() from it. An error will occur. Explain the reason of error in comments. You can eliminate this error by creating Func() in Employed class but since we want to call the function of Person's class, our requirements will

not be fulfilled and it will be a wrong practice. Solve this ambiguity with an appropriate way and comment on how you solved it.

Problem 10:

Design a class named Customer with the following member variables:

- CustomerNumber (holds unique number for each customer)
- Gender (remains constant)
- Name
- CNIC
- City
- State
- PhoneNo

Customer class will have InputCustomerData member function, which will Input all the data for customer. DisplayCustomerData member function, which will display all the data for customer.

Next, design a class name SpecialCustomer derived from Customer class having following member variables:

- Discount (in %age)
- No. of visits

Write appropriate Accessor(getter) and Mutator(setter) functions for these members.

Demonstrate an object of Customer and SpecialCustomer class.