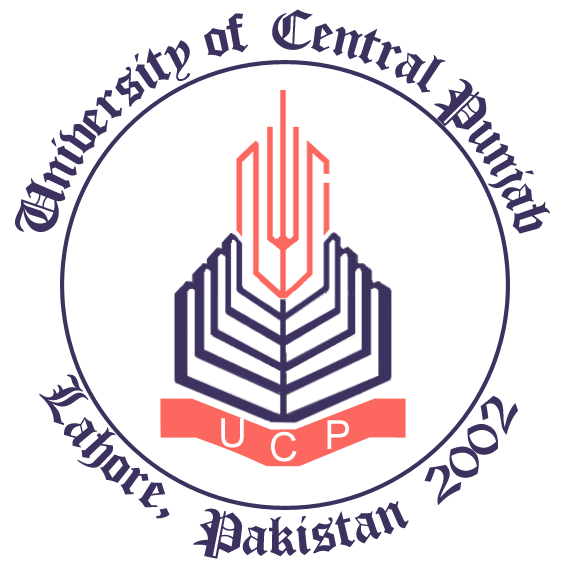
DSA Revision Lab 1



Topic: Composition

Session: Spring 2023

Faculty of Information Technology

UCP Lahore Pakistan

Table of Contents

Topic: Constructors, Setters and Getters ................................................................................................. 1

[LabObjectives 3](#_Toc5070)

[Instructions 3](#_Toc5071)

[Sample task: 3](#_Toc5072)

[Lab Tasks: 5](#_Toc5073)

[Task 1: 5](#_Toc5074)

[Task 2: 5](#_Toc5075)

Task 3: ............................................................................................... **Error! Bookmark not defined.** Task 4: ............................................................................................... **Error! Bookmark not defined.**

# LabObjectives

Lab Objectives:

• The purpose of this activity is to clear the concept of relationship among objects i.e composition. At the end of this lab you will be able to implement composition if needed in the given scenario.

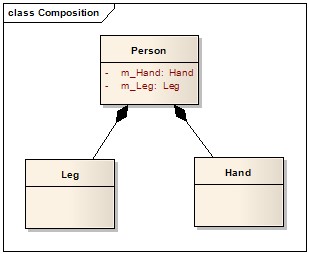
# Instructions

* Indent your code
* Comment your code
* Use meaningful variable names
* Plan your code carefully on a piece of paper before you implement it.
* Name of the program should be same as the task name. i.e. the first program should be

Task\_1\_01.cpp

# Sample task:

In cases where in addition to the part-of relationship between ClassA and ClassB - there’s a strong life cycle dependency between the two, meaning that when ClassA is deleted then ClassB is also deleted as a result, we should be more specific and use the composition link instead of the aggregation link or the association link.



Solution:

**class Date** #include<iostream> #include<string.h> using namespace std; class Date{ private: int Date; int Month; int Year;

|  |
| --- |
| public:  Date(int d = 1,int m = 1,int y = 2013)  { cout<<"Date Constructor "<<endl; Day = d;  Month = m;  Year = y;  }  void PrintDate() const  { cout<<Day<<"/"<<Month<<"/"<<Year<<endl;  }    ~Date()  { cout << "Date Destructor"<<endl; }  };  **class Person**    class Person { private:    char Name[25];    const Date BirthDate; public:    Person(const char \* const n, const Date &bd):BirthDate(bd)    {    cout<<"Person Constructor."<<endl; strcpy(Name,n);    }  void PrintPerson() const  { cout<<"Name: "<<Name<<endl; cout<<"Birth Date: "; BirthDate.PrintDate();  }    ~Person()  { cout << "Person Destructor"<<endl; } };  int main()  {  Date D1(13,10,1989);  Person P1("Bilal Hassan",D1);  P1.PrintPerson();  } |

# Lab Tasks:

## Task 1:

Make a University class with objects of following classes:

1. ClassRoom
2. ComputerLab

Class must have following member variables

1. String RoomNumber
2. int area
3. int NoOfChairs
4. bool projector

ComputerLab must have following member variables

1. String RoomNumber
2. int NoOfComputer
3. bool projector

## Task 2:

Make a Computer System class with objects of following classes

1. Monitor
2. CPU
3. keyboard

Monitor must have following member variables

1. String CompanyName
2. int Size
3. float price

CPU must have following member variables

1. String CompanyName
2. int Speed
3. float price

Key Board must have following member variables

1. String CompanyName
2. int NumOfKeys
3. float price

In main user should see any information about his/her system.

NOTE:

Also write the constructor & destructor sequence for every class.