OBJECT ORIENTED PROGRAMMING LAB

Submitted By: Tanmay Vig

Roll No.: 19BCS061

Batch: B. Tech 3rd year

Submitted to: -

Dr. Mumtaz Ahmed

Mr. Shamim Ahmad

Department Of Computer Engineering

S no.	Program title	Date Submitted
1	Enter some Students data like Name, Roll no, Class, subject code, sub name, marks obtained, max marks etc.	10/08/21
2	Implement Program p1 using Default Constructor and Parameterized Constructor	17/08/21
3	Implement Array of Objects in given problem	24/08/21
4	Implement Bank Account	31/08/21
5	Implement Friend Function	07/9/21
6	Implement Book Store	14/9/21
7	Implement Employee	21/9/21
8	Implement Operator Overloading: String Operations	5/10/21
9	Implement Stack using classes and objects	26/10/21
10	Implement Inheritance in given problem	16/11/21
11	Implement Inheritance in given problem	23/11/21
12	Templates-Write a program in C++ to implement a generic Stack.	30/11/21
13	Templates-Write a program in C++ to implement a generic Stack.	7/12/21

P1: Enter some Students data like Name, roll no, Class, subject code, subname, marks obtained, max marks etc. Print their Grades according to following rule Grade A if marks obtained are in range 75-100. Grade B if marks obtained are in range 60-74. Grade C if marks obtained are in range 45-59 Grade D if marks obtained are in range 35-44 Grade F if marks obtained are in range 0-34.

Source Code:

```
#include<iostream>
#include<vector>
#include<string>
#include <iomanip>
#include<vector>

using namespace std;

class Student{
   private:
      string name;
      string _class;
      string roll_num;
      int num;
      string *subject_code;
      string *subject_name;
```

```
    float *marks;
    float max_marks;
  public:

    Student() {}

    Student(string na,string cl,string r_n,int n, string s_c[],string s_n[], float m[]){
      name=na;
      _class=cl;
      roll_num=r_n;
      num=n;
      subject_code=s_c;
      subject_name=s_n;
      marks=m;
      max_marks=-1;
      for(int i=0;i<num;i++){
        if(max_marks<marks[i]) max_marks=marks[i];
      }
    }
    void marksheet(){
      cout<<"Name: "<<name<<"\nRoll Number: "<<roll_num<<"\nClass: "<<_class<<endl;
      cout<<"-----------------------------------------\n";
      cout<<"| Subject code| Subject Name| Marks| Grade|\n";
      cout<<"-----------------------------------------\n";
      for(int i=0;i<num;i++){
        cout<<"| "<<setw(12)<<subject_code[i]<<"| "<<setw(12)<<subject_name[i]<<"|
"<<setw(5)<<marks[i]<<"| "<<setw(5)<<grade(marks[i])<<"|\n";
```

```
            }
            cout<<"----------------------------------------\n";
            cout<<"Max_marks: "<<max_marks<<endl;
            cout<<"Net Grade: "<<grade(total(marks,num)/num)<<endl;
        }

        float total(float m[],int n){
            float sum=0;
            for(int i=0;i<n;i++) sum+=m[i];
            return sum;
        }

        string grade(float m){
            if(m>=75) return "A";
            else if(m<75 && m>=60) return "B";
            else if(m<60 && m>=45) return "C";
            else if(m<45 && m>=35) return "D";
            return "F";
        }
};

int main(){
    int n;
    cout<<"Enter Number of students:\t";
    cin>>n;
    Student arr[100];
    for(int i=0;i<n;i++){
```

```
        string name,clas,r_n,s_c[20],s_n[20]; int num; float m[20];
        cout<<"Enter Name and Roll Number of student:\t";
        cin>>name>>r_n;
        cout<<"Enter Class of student:\t";
        cin>>clas;
        cout<<"Enter Number of subjects:\t";
        cin>>num;
        cout<<"Enter the Subject codes, Subject Name, marks obtained\n";
        for(int j=0;j<num;j++) cin>>s_c[j]>>s_n[j]>>m[j];
        Student s(name,clas,r_n,num,s_c,s_n,m);
        cout<<endl;
        s.marksheet();
    }
    return 0;
}
```

Output:

```
D:\sem-5\opp_Lab\1.exe
Enter Number of students:       2
Enter Name and Roll Number of student:  Ruben 19BCS061
Enter Class of student: 3rd
Enter Number of subjects:       4
Enter the Subject codes, Subject Name, marks obtained
CEN501 SE 80
CEN502 CI 23
CEN503 CN 60
CEN504 ML 54

Name: Ruben
Roll Number: 19BCS061
Class: 3rd
-----------------------------------------
| Subject code| Subject Name| Marks| Grade|
-----------------------------------------
|       CEN501|           SE|    80|     A|
|       CEN502|           CI|    23|     F|
|       CEN503|           CN|    60|     B|
|       CEN504|           ML|    54|     C|
-----------------------------------------
Max_marks: 80
Net Grade: C
Enter Name and Roll Number of student:  Bedi 20BCS010
Enter Class of student: 2nd
Enter Number of subjects:       3
Enter the Subject codes, Subject Name, marks obtained
CEN301 DS 90
CEN302 DLD 51
CEN OOP 80

Name: Bedi
Roll Number: 20BCS010
Class: 2nd
-----------------------------------------
| Subject code| Subject Name| Marks| Grade|
-----------------------------------------
|       CEN301|           DS|    90|     A|
|       CEN302|          DLD|    51|     C|
|          CEN|          OOP|    80|     A|
-----------------------------------------
Max_marks: 90
Net Grade: B

--------------------------------
Process exited after 168.8 seconds with return value 0
Press any key to continue . . .
```

P2: implement Program P1 using Default Constructor and Parameterized constructor.

Source Code:

```
#include <bits/stdc++.h>

using namespace std;

class Student {

   public:

      string name;

      int _class, rollNum, totalSubjects;

      string subjectNames[100], subjectCodes[100];

      int marks[100], maxMarks[100];

// Default Constructor

   Student() {

   this->name="";

   this->_class=0;

   this->rollNum=0;

   }

// Parameterized Constructor

   Student(string name, int _class, int rollNum) {

      this->name=name;

      this->_class=_class;

      this->rollNum=rollNum;
```

```
    }
// Student data setter, in case of default constructor
    void setStudentData(string name, int _class, int rollNum) {
        this->name=name;
        this->_class=_class;
        this->rollNum=rollNum;
    }
    void setData() {
        int n;
        cout << "Enter the number of subjects" << endl;
        cin >> n;
        this->totalSubjects = n;
        for(int i=0;i<n;i++) {
            string subName, subCode;
            int marks, maxMarks;
            cout << "Enter the name, code, marks and max marks for subject " << i+1 << endl;
            cin >> subName;
            cin >> subCode;
            cin >> marks;
            cin >> maxMarks;
            this->subjectNames[i] = subName;
            this->subjectCodes[i] = subCode;
            this->marks[i] = marks;
```

```
            this->maxMarks[i] = maxMarks;
        }
    }
    void getData() {
        cout << "Subject wise marks of " << this->name << " are" <<endl;
        cout << "Name\t\t" << "Code\t\t" << "Max Marks\t\t" << "Marks Obtained" << endl;
        int totalMarks = 0, totalMaxMarks = 0;
        double avg;
        for(int i=0;i<this->totalSubjects;i++) {
            cout << this->subjectNames[i] << "\t\t" << this->subjectCodes[i] << "\t\t" << this-
            >maxMarks[i] << "\t\t\t" << this->marks[i] << endl;
            totalMarks += this->marks[i];
            totalMaxMarks += this->maxMarks[i];
        }
        avg = ((double)totalMarks/(double)totalMaxMarks)*100.0;
        cout << endl << "Grade of student " << this->name << " is: ";
        if(avg >= 75.0) cout << "A" << endl << endl;
        else if(avg >= 60.0) cout << "B" << endl << endl;
        else if(avg >= 45.0) cout << "C" << endl << endl;
        else if(avg >= 35.0) cout << "D" << endl << endl;
        else cout << "F" << endl << endl;
    }
};
```

```
int main() {
    cout << "Enter the number of students" << endl;
    int n;
    cin >> n;
    for(int i=1;i<=n;i++) {
        string name;
        int _class, rollNum;
        cout << "Enter the name, class and roll number of student " << i << endl;
        cin >> name;
        cin >> _class;
        cin >> rollNum;
        int choice;
        cout << "Enter 0 to use default constructor, 1 to use parameterized constructor" << endl;
        cin >> choice;
        if(choice == 0) {
            Student* student = new Student();
            student->setStudentData(name, _class, rollNum);
            student->setData();
            student->getData();
        } else {
            Student* student = new Student(name, _class, rollNum);
            student->setData();
            student->getData();
```

```
    }

  }

return 0;

}
```

Output:

D:\sem-5\oop_Lab\2.exe

```
Enter the name, code, marks and max marks for subject 2
Sub2 102 96 100
Subject wise marks of Tanmay are
Name            Code            Max Marks               Marks Obtained
Sub1            101             89                      89
Sub2            102             100                     96

Grade of student Tanmay is: A

Enter the name, class and roll number of student 2
Ruben 12 34
Enter 0 to use default constructor, 1 to use parameterized constructor
1
Enter the number of subjects
2
Enter the name, code, marks and max marks for subject 1
Sub1 121 20 60
Enter the name, code, marks and max marks for subject 2
Sub2 122 89 90
Subject wise marks of Ruben are
Name            Code            Max Marks               Marks Obtained
Sub1            121             60                      20
Sub2            122             90                      89

Grade of student Ruben is: B


--------------------------------
Process exited after 138.4 seconds with return value 0
Press any key to continue . . .
```

P3 a: Create a class called Employee that contains employee number (type long), name (an object of class string) and salary (integer). Include a member function called getData() to get data from the user for insertion into the object, and another function called putData() to display the data. Define both the functions outside the class. Assume the name has embedded blanks (like Rakesh Kumar Mehta). Write a main() program to execute this class. It should create an array of type Employee, and then invite the user to input data for up to 10 employees. Finally, it should print out the data for all the employees in proper format.

Source code:

```
#include<iostream>
#include<string>
#include <iomanip>

using namespace std;

class Employee{
   public:
      long long int employee_number;
      string name;
      int salary;
      void getData();
      void putData();
```

```
};

void Employee::getData(){
   cout<<"Enter employee number\t";
   cin>>employee_number;
   cin.clear();
   cin.sync();
   cout<<"Enter Name of employee"<<endl;
   getline(cin,name);
   cout<<"Enter Salary\t";
   cin>>salary;
}

void Employee::putData(){

cout<<left<<"|"<<setw(15)<<employee_number<<"|"<<setw(15)<<name<<"|"<<setw(15)<<sal
ary<<endl;
}

int main(){
   Employee arr[10];
   cout<<"Enter details:\n";
   for(int i=0;i<10;i++){
      cout<<"Enter data for employee "<<i+1<<endl;
      arr[i].getData();
   }
   cout<<"Employee Details:\n"<<
```

```
        "_____\n|Employee Number|Name
    |Salary       |\n_____\n";
      for(int i=0;i<10;i++){
        arr[i].putData();
        cout<<"_____\n";
      }
      return 0;
    }
```

Output:

```
D:\sem-5\oop_Lab\3a.exe
Enter data for employee 9
Enter employee number   9
Enter Name of employee
Garvit Ahuja
Enter Salary    88000
Enter data for employee 10
Enter employee number   10
Enter Name of employee
Bhargav Bajaj
Enter Salary    60000
Employee Details:
_____________________________________
|Employee Number|Name           |Salary        |
_____________________________________
|1              |Paras Dang     |80000
_____________________________________
|2              |Prabhmeet Singh|75000
_____________________________________
|3              |Parmeesh Singh |89000
_____________________________________
|4              |Ruben Singh    |89000
_____________________________________
|5              |Naman Sachdeva |60000
_____________________________________
|6              |Mridul Bhatia  |50000
_____________________________________
|7              |Aryan Bedi     |89000
_____________________________________
|8              |Tanmay Vig     |1000000
_____________________________________
|9              |Garvit Ahuja   |88000
_____________________________________
|10             |Bhargav Bajaj  |60000
_____________________________________

--------------------------------
Process exited after 229.1 seconds with return value 0
Press any key to continue . . .
```

```
D:\sem-5\oop_Lab\3a.exe
Enter details:
Enter data for employee 1
Enter employee number   1
Enter Name of employee
Paras Dang
Enter Salary    80000
Enter data for employee 2
Enter employee number   2
Enter Name of employee
Prabhmeet Singh
Enter Salary    75000
Enter data for employee 3
Enter employee number   3
Enter Name of employee
Parmeesh Singh
Enter Salary    89000
Enter data for employee 4
Enter employee number   4
Enter Name of employee
Ruben Singh
Enter Salary    89000
Enter data for employee 5
Enter employee number   5
Enter Name of employee
Naman Sachdeva
Enter Salary    60000
Enter data for employee 6
Enter employee number   6
Enter Name of employee
Mridul Bhatia
Enter Salary    50000
Enter data for employee 7
Enter employee number   7
Enter Name of employee
Aryan Bedi
Enter Salary    89000
Enter data for employee 8
Enter employee number   8
Enter Name of employee
Tanmay Vig
Enter Salary    1000000
Enter data for employee 9
Enter employee number   9
Enter Name of employee
Garvit Ahuja
Enter Salary    88000
Enter data for employee 10
Enter employee number   10
Enter Name of employee
Bhargav Bajaj
```

P3 b: Create a class called Student that contains a Student id (type int), name (char array) and marks (integer). Include a member function called getStudent() to get data from the user for insertion into the object, and another function called printStudent() to display the data. Define both the functions inside the class. Assume the name has no embedded blanks. Write a main() program to execute this class. It should create an array of type Student, and then invite the user to input data for up to 10 students. Finally, it should print out the data for all the students in proper format.

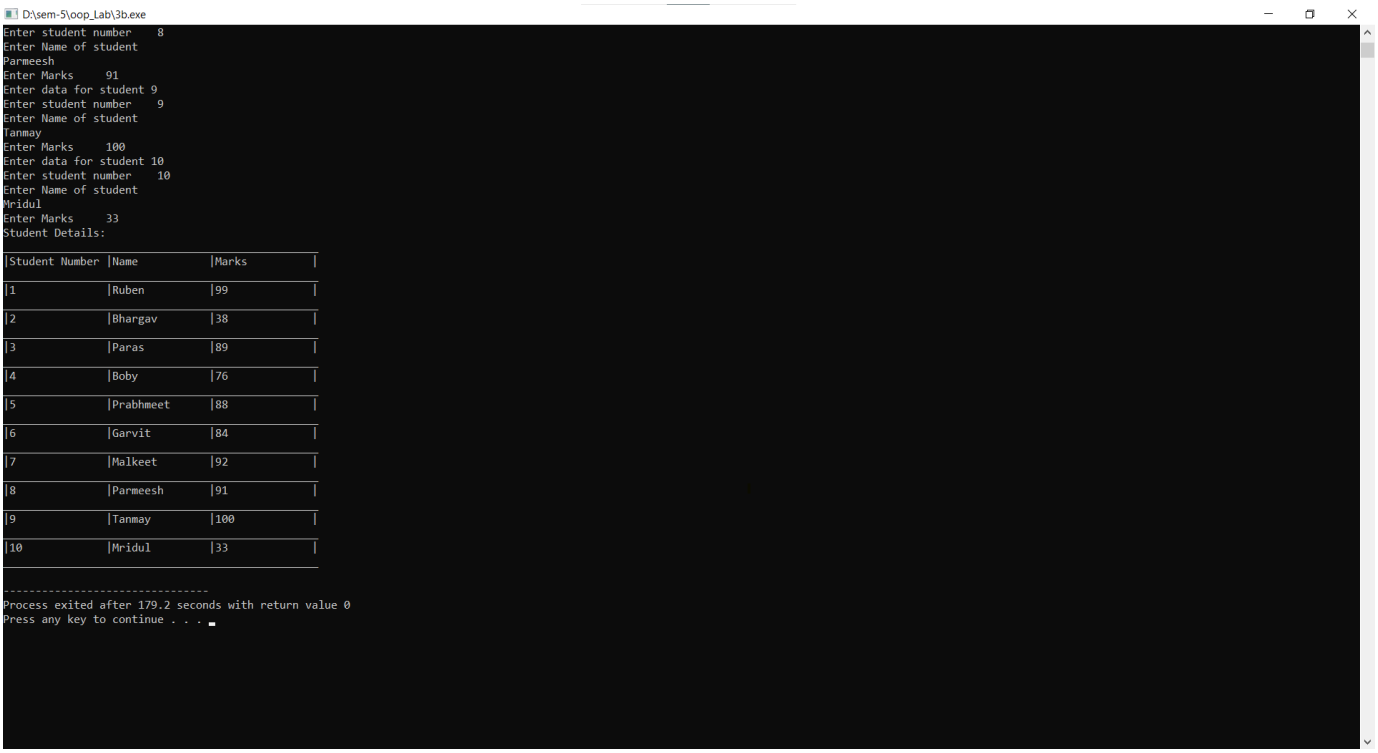
Source Code:

```
#include<iostream>
#include<string>
#include <iomanip>
#define M 10

using namespace std;

class Student{
   public:
      int student_id;
```

```
        char name[30];
        int marks;
        void getData(){
           cout<<"Enter student number\t";
           cin>>student_id;
           cin.clear();
           cin.sync();
           cout<<"Enter Name of student"<<endl;
           cin.getline(name,30);
           cout<<"Enter Marks\t";
           cin>>marks;
        }
        void putData(){

cout<<left<<"|"<<setw(15)<<student_id<<"|"<<setw(15)<<name<<"|"<<setw(15)<<marks<<"|
"<<endl;
        }
};

int main(){
   Student arr[M];
   cout<<"Enter details:\n";
   for(int i=0;i<M;i++){
      cout<<"Enter data for student "<<i+1<<endl;
      arr[i].getData();
   }
   cout<<"Student Details:\n"<<
```

```
    "_____\n|Student Number |Name
|Marks        |\n_____\n";
   for(int i=0;i<M;i++){
      arr[i].putData();
      cout<<"_____\n";
   }
   return 0;
}
```

output:


D:\sem-5\oop_Lab\3b.exe
Enter student number 8
Enter Name of student
Parmeesh
Enter Marks 91
Enter data for student 9
Enter student number 9
Enter Name of student
Tanmay
Enter Marks 100
Enter data for student 10
Enter student number 10
Enter Name of student
Mridul
Enter Marks 33
Student Details:
Student Number	Name	Marks
1	Ruben	99
2	Bhargav	38
3	Paras	89
4	Boby	76
5	Prabhmeet	88
6	Garvit	84
7	Malkeet	92
8	Parmeesh	91
9	Tanmay	100
10	Mridul	33
Process exited after 179.2 seconds with return value 0
Press any key to continue . . .

P4: Write a program in C++ to represent a Bank Account.

Data members:

- Account number
- Name of the depositor
- Type of account
- Balance amount in the account

Member functions:

- To assign initial values
- To deposit an amount
- To withdraw an amount after checking the balance
- To display name and balance of a single customer if account number is entered through
- the keyboard
- To display the details of all customers
- Sort the customers according to their balance

Write a main program to test the class. Your program must be able to handle at least 10 customers.

Source Code:

```
#include<iostream>
#include<string>
#define M 10
```

```
using namespace std;



class Account{
   private:
      long long int account_number;
      string name;
      string account_type;
      int balance;

   public:
      void putDetails(){
         cout<<"Enter Account Number:\t";
         cin>>account_number;
         cin.ignore();
         cout<<"Enter Name of depositor:\t";
         getline(cin, name);
         cout<<"Enter Type of Account(Current, Saving, Salary):\t";
         getline(cin, account_type);
         cout<<"Enter Balance amount in the account:\t";
         cin>>balance;
      }

      void deposite(int amt){
         balance=balance+amt;
         cout<<"Amount deposited: "<<amt<<". Total balance: "<<balance<<endl;
      }
```

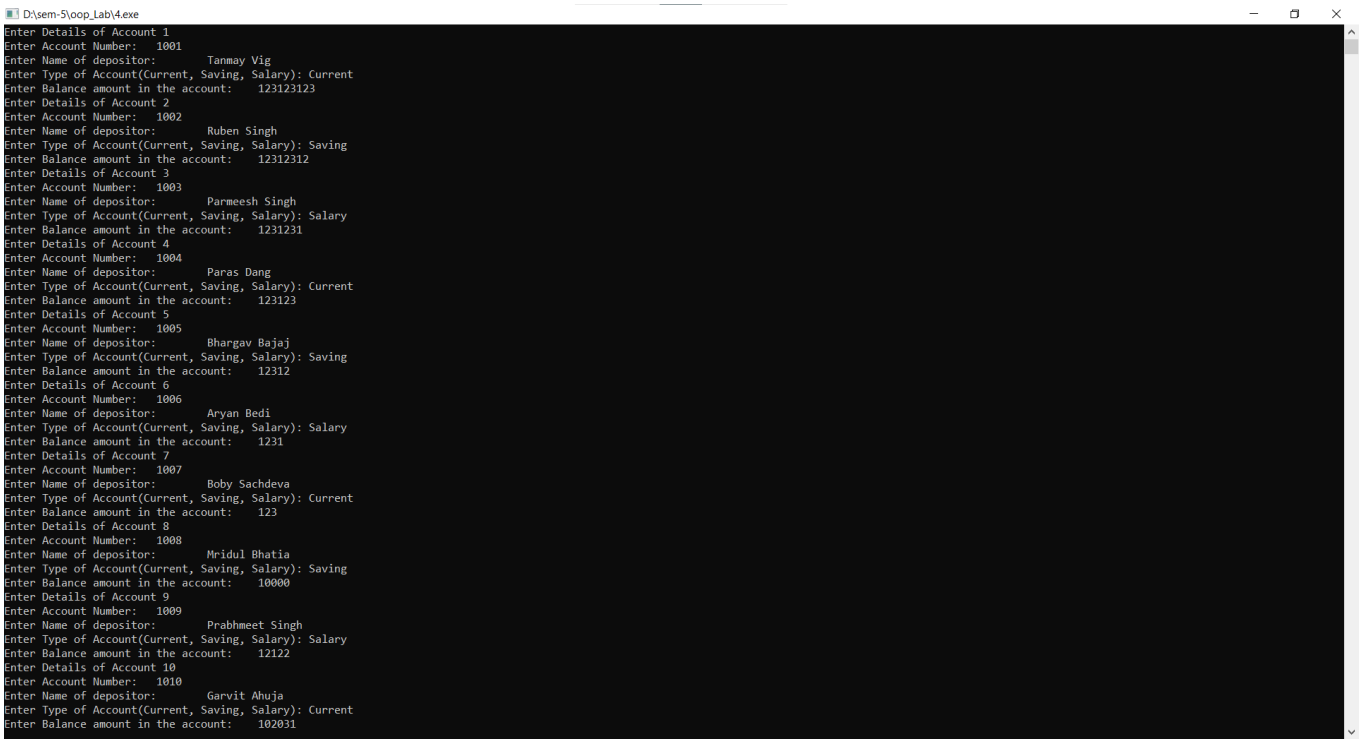
```
    void withdraw(int amt){
      if(balance<amt) cout<<"Error: please input Valid amount.\nYour current balance is: "<<balance<<endl;
      else{
        balance=balance-amt;
        cout<<"Amount withdraw: "<<amt<<". Total balance: "<<balance<<endl;
      }
    }
    void display(){
      cout<<"Name: "<<name<<".\nAccount Type: "<<account_type<<".\nBalance: "<<balance<<endl;
    }
    int getbalance(){
      return balance;
    }
    long long int getAccNum(){
      return account_number;
    }
};

int findAcc(Account arr[],long long int num) {
  for(int i=0;i<M;i++) {
    if(arr[i].getAccNum()==num) return i;
  }
  return -1;
}
```

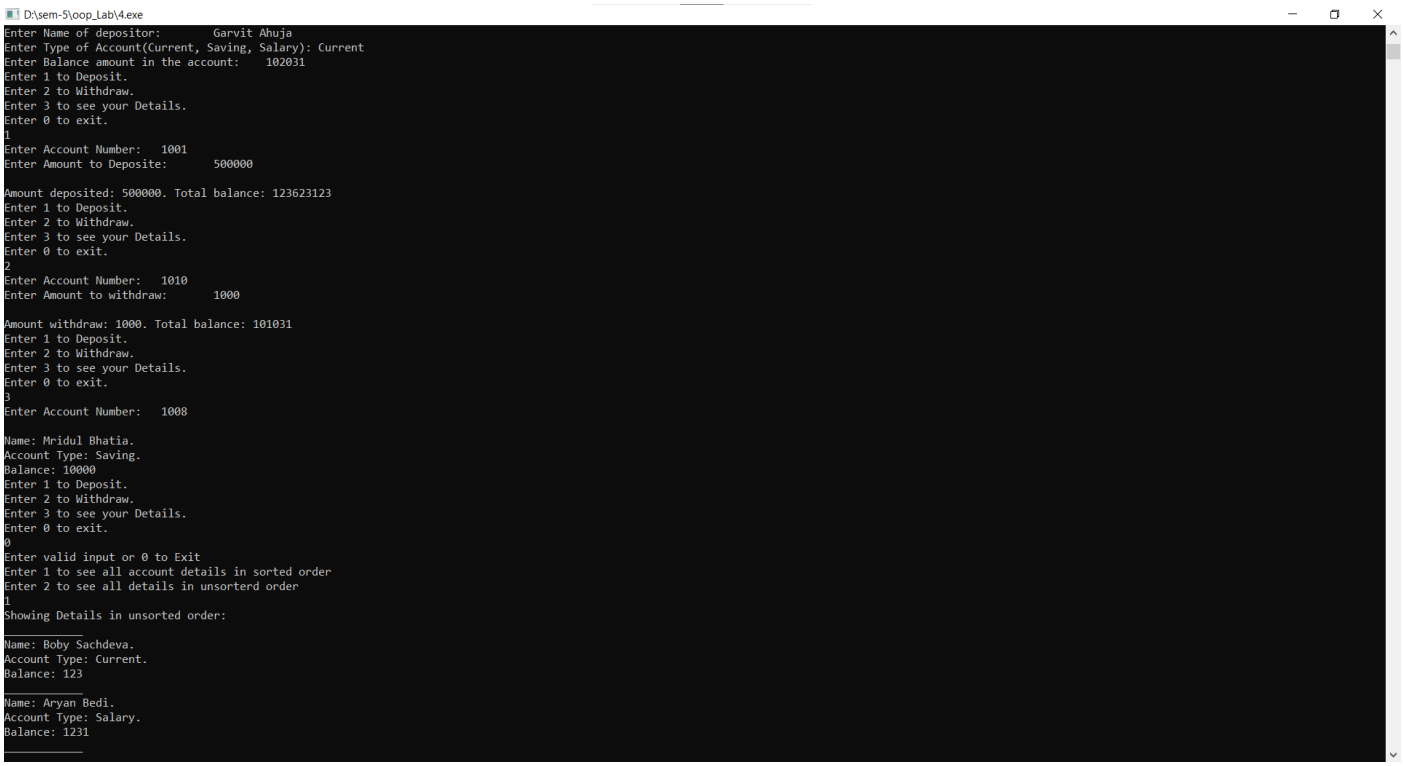
```
int main(){
   Account arr[M];
   int n;
   long long int accNum;
   for(int i=0;i<M;i++){
      cout<<"Enter Details of Account "<<i+1<<endl;
      arr[i].putDetails();
   }
   do{
      cout<<"Enter 1 to Deposit. \nEnter 2 to Withdraw. \nEnter 3 to see your Details.\nEnter 0 to exit.\n";
      cin>>n;
      if(n){
         cout<<"Enter Account Number:\t";
         cin>>accNum;
      }
      int index=findAcc(arr,accNum),amt;
      if(index!=-1){
         switch (n)
         {
         case 1:
            cout<<"Enter Amount to Deposite:\t";
            cin>>amt;
            cout<<endl;
            arr[index].deposite(amt);
            break;
         case 2:
```

```
            cout<<"Enter Amount to withdraw:\t";
            cin>>amt;
            cout<<endl;
            arr[index].withdraw(amt);
            break;
        case 3:
            cout<<endl;
            arr[index].display();
            break;
        default:
            cout<<"Enter valid input or 0 to Exit\n";
            break;
        }
    }else if(index==-1){
        cout<<"\nError: no account found.\n";
    }
  }while(n!=0);
  cout<<"Enter 1 to see all account details in sorted order\nEnter 2 to see all details in
unsorterd order\n";
  cin>>n;
  if(n==1){
    int i,j;
    for(i=0;i<M-1;i++){
      for(j=0;j<M-i-1;j++){
        if(arr[j].getbalance()>arr[j+1].getbalance()){
          Account temp=arr[j];
          arr[j]=arr[j+1];
```

```
                    arr[j+1]=temp;
                }
            }
        }
        cout<<"Showing Details in unsorted order:\n";
        cout<<"_____________"<<endl;
        for(int i=0;i<M;i++){
            arr[i].display();
            cout<<"_____________"<<endl;
        }
    }else{
        cout<<"Showing Details in unsorted order:\n";
        cout<<"_____________"<<endl;
        for(int i=0;i<M;i++){
            arr[i].display();
            cout<<"_____________"<<endl;
        }
    }
    return 0;
}
```

Output:


D:\sem-5\oop_Lab\4.exe
Enter Details of Account 1
Enter Account Number: 1001
Enter Name of depositor: Tanmay Vig
Enter Type of Account(Current, Saving, Salary): Current
Enter Balance amount in the account: 123123123
Enter Details of Account 2
Enter Account Number: 1002
Enter Name of depositor: Ruben Singh
Enter Type of Account(Current, Saving, Salary): Saving
Enter Balance amount in the account: 12312312
Enter Details of Account 3
Enter Account Number: 1003
Enter Name of depositor: Parmeesh Singh
Enter Type of Account(Current, Saving, Salary): Salary
Enter Balance amount in the account: 1231231
Enter Details of Account 4
Enter Account Number: 1004
Enter Name of depositor: Paras Dang
Enter Type of Account(Current, Saving, Salary): Current
Enter Balance amount in the account: 123123
Enter Details of Account 5
Enter Account Number: 1005
Enter Name of depositor: Bhargav Bajaj
Enter Type of Account(Current, Saving, Salary): Saving
Enter Balance amount in the account: 12312
Enter Details of Account 6
Enter Account Number: 1006
Enter Name of depositor: Aryan Bedi
Enter Type of Account(Current, Saving, Salary): Salary
Enter Balance amount in the account: 1231
Enter Details of Account 7
Enter Account Number: 1007
Enter Name of depositor: Boby Sachdeva
Enter Type of Account(Current, Saving, Salary): Current
Enter Balance amount in the account: 123
Enter Details of Account 8
Enter Account Number: 1008
Enter Name of depositor: Mridul Bhatia
Enter Type of Account(Current, Saving, Salary): Saving
Enter Balance amount in the account: 10000
Enter Details of Account 9
Enter Account Number: 1009
Enter Name of depositor: Prabhmeet Singh
Enter Type of Account(Current, Saving, Salary): Salary
Enter Balance amount in the account: 12122
Enter Details of Account 10
Enter Account Number: 1010
Enter Name of depositor: Garvit Ahuja
Enter Type of Account(Current, Saving, Salary): Current
Enter Balance amount in the account: 102031


D:\sem-5\oop_Lab\4.exe
Enter Name of depositor: Garvit Ahuja
Enter Type of Account(Current, Saving, Salary): Current
Enter Balance amount in the account: 102031
Enter 1 to Deposit.
Enter 2 to Withdraw.
Enter 3 to see your Details.
Enter 0 to exit.
1
Enter Account Number: 1001
Enter Amount to Deposite: 500000
Amount deposited: 500000. Total balance: 123623123
Enter 1 to Deposit.
Enter 2 to Withdraw.
Enter 3 to see your Details.
Enter 0 to exit.
2
Enter Account Number: 1010
Enter Amount to withdraw: 1000
Amount withdraw: 1000. Total balance: 101031
Enter 1 to Deposit.
Enter 2 to Withdraw.
Enter 3 to see your Details.
Enter 0 to exit.
3
Enter Account Number: 1008
Name: Mridul Bhatia.
Account Type: Saving.
Balance: 10000
Enter 1 to Deposit.
Enter 2 to Withdraw.
Enter 3 to see your Details.
Enter 0 to exit.
0
Enter valid input or 0 to Exit
Enter 1 to see all account details in sorted order
Enter 2 to see all details in unsorterd order
1
Showing Details in unsorted order:
Name: Boby Sachdeva.
Account Type: Current.
Balance: 123
Name: Aryan Bedi.
Account Type: Salary.
Balance: 1231


D:\sem-5\oop_Lab\4.exe
Showing Details in unsorted order:
Name: Boby Sachdeva.
Account Type: Current.
Balance: 123
Name: Aryan Bedi.
Account Type: Salary.
Balance: 1231
Name: Mridul Bhatia.
Account Type: Saving.
Balance: 10000
Name: Prabhmeet Singh.
Account Type: Salary.
Balance: 12122
Name: Bhargav Bajaj.
Account Type: Saving.
Balance: 12312
Name: Garvit Ahuja.
Account Type: Current.
Balance: 101031
Name: Paras Dang.
Account Type: Current.
Balance: 123123
Name: Parmeesh Singh.
Account Type: Salary.
Balance: 1231231
Name: Ruben Singh.
Account Type: Saving.
Balance: 12312312
Name: Tanmay Vig.
Account Type: Current.
Balance: 123623123
Process exited after 359.6 seconds with return value 0
Press any key to continue . . .

P5 a: Write the definition for a class called Rectangle that has floating point data members length and width. The class has the following member functions:

void setlength(float) to set the length data member

void setwidth(float) to set the width data member

float perimeter() to calculate and return the perimeter of the rectangle

float area() to calculate and return the area of the rectangle

void show() to display the length and width of the rectangle

int sameArea(Rectangle) that has one parameter of type Rectangle. sameArea() returns 1 if the two Rectangles have the same area, and returns 0 if they don't.

Write the definitions for each of the above member functions.

Write main function to create two rectangle objects. Set the length and width of the first rectangle to 5 and 2.5. Set the length and width of the second rectangle to 5 and 18.9. Display each rectangle and its area and perimeter.

3. Check whether the two Rectangles have the same area and print a message indicating the result. Set the length and width of the first rectangle to 15 and 6.3. Display each Rectangle and its area and perimeter again. Again, check whether the two Rectangles have the same area and print a message indicating the result.

Source Code:

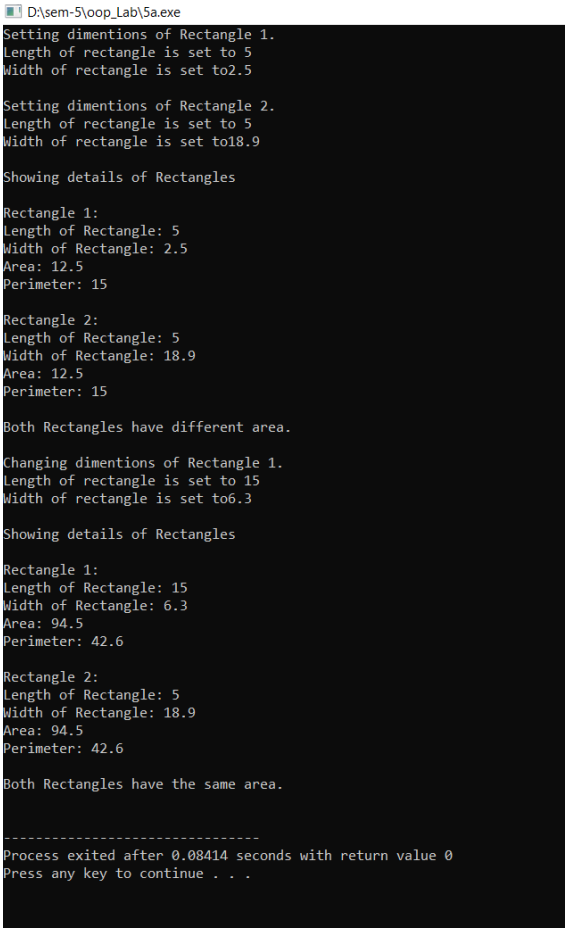
```
#include<iostream>


using namespace std;


class Rectangle{
   private:
      float length;
      float width;
   public:
      void setLength(float);
      void setWidth(float);
      float perimeter();
      float area();
      void show();
      int sameArea(Rectangle);
};


void Rectangle::setLength(float l){
   length=l;
```

```
    cout<<"Length of rectangle is set to "<<l<<endl;
}

void Rectangle::setWidth(float w){
    width=w;
    cout<<"Width of rectangle is set to"<<w<<endl;
}

float Rectangle::perimeter(){
    return 2*(length+width);
}

float Rectangle::area(){
    return length*width;
}

void Rectangle::show(){
    cout<<"Length of Rectangle: "<<length<<"\nWidth of Rectangle: "<<width<<endl;
}

int Rectangle::sameArea(Rectangle r){
    return area()==r.area() ? 1 : 0;
}

int main(){
    Rectangle r1,r2;
    cout<<"Setting dimentions of Rectangle 1.\n";
```

```
r1.setLength(5); r1.setWidth(2.5);
cout<<endl;

cout<<"Setting dimentions of Rectangle 2.\n";
r2.setLength(5); r2.setWidth(18.9);
cout<<endl;

cout<<"Showing details of Rectangles\n\n";
cout<<"Rectangle 1:\n";
r1.show();
cout<<"Area: "<<r1.area()<<endl;
cout<<"Perimeter: "<<r1.perimeter()<<endl;
cout<<endl;
cout<<"Rectangle 2:\n";
r2.show();
cout<<"Area: "<<r1.area()<<endl;
cout<<"Perimeter: "<<r1.perimeter()<<endl;
cout<<endl;

if(r1.sameArea(r2)) cout<<"Both Rectangles have the same area.\n\n";
else cout<<"Both Rectangles have different area.\n\n";

cout<<"Changing dimentions of Rectangle 1.\n";
r1.setLength(15);
r1.setWidth(6.3);
cout<<endl;
```

```
    cout<<"Showing details of Rectangles\n\n";
    cout<<"Rectangle 1:\n";
    r1.show();
    cout<<"Area: "<<r1.area()<<endl;
    cout<<"Perimeter: "<<r1.perimeter()<<endl;
    cout<<endl;
    cout<<"Rectangle 2:\n";
    r2.show();
    cout<<"Area: "<<r1.area()<<endl;
    cout<<"Perimeter: "<<r1.perimeter()<<endl;
    cout<<endl;

    if(r1.sameArea(r2)) cout<<"Both Rectangles have the same area.\n\n";
    else cout<<"Both Rectangles have different area.\n\n";

    return 0;
}
```

Output:



P5 b: Write the definition for a class called complex that has floating point data members for storing

real and imaginary parts. The class has the following member functions:

void set(float, float) to set the specified value in object

void disp() to display complex number object

complex sum(complex) to sum two complex numbers & return complex number

1. Write the definitions for each of the above member functions.

2. Write main function to create three complex number objects. Set the value in two objects and

call sum() to calculate sum and assign it in third object. Display all complex numbers.

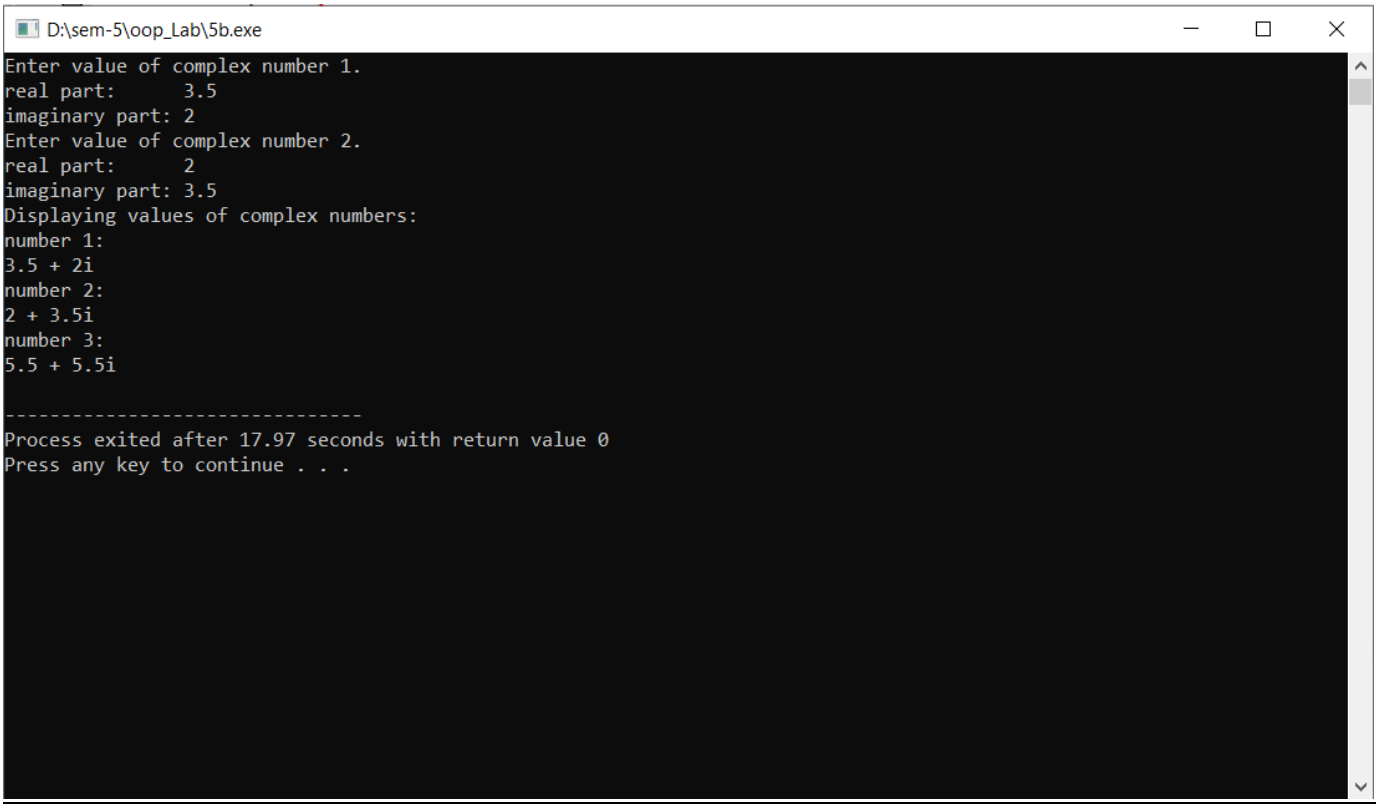
Source code:

```
#include<iostream>


using namespace std;


class Complex{
   private:
      float real;
      float imag;
      Complex friend sum(Complex,Complex);
   public:
      void set(float r,float i){
         real = r; imag = i;
      }
      void display(){
         cout<<real<<" + "<<imag<<"i\n";
      }
};
```

```
Complex sum(Complex x,Complex y){
   Complex c;
   c.real = x.real+y.real; c.imag = x.imag+y.imag;
   return c;
}

int main(){
   Complex a,b,c;
   float r,i;
   cout<<"Enter value of complex number 1.\nreal part:\t";
   cin>>r;
   cout<<"imaginary part:\t"; cin>>i;
   a.set(r,i);

   cout<<"Enter value of complex number 2.\nreal part:\t";
   cin>>r;
   cout<<"imaginary part:\t"; cin>>i;
   b.set(r,i);

   c = sum(a,b);
   cout<<"Displaying values of complex numbers:\n";
   cout<<"number 1:\n";
   a.display();
   cout<<"number 2:\n";
   b.display();
   cout<<"number 3:\n";
   c.display();
```

```
    return 0;
}
```

Output:



P5 c: Write the definition for a class called Distance that has data member feet as integer and inches as float. The class has the following member functions:

void set(int, float) to give value to object

void disp() to display distance in feet and inches

Distance add(Distance) to sum two distances & return distance

1. Write the definitions for each of the above member functions.

2. Write main function to create three Distance objects. Set the value in two objects and call add() to calculate sum and assign it in third object. Display all distances.

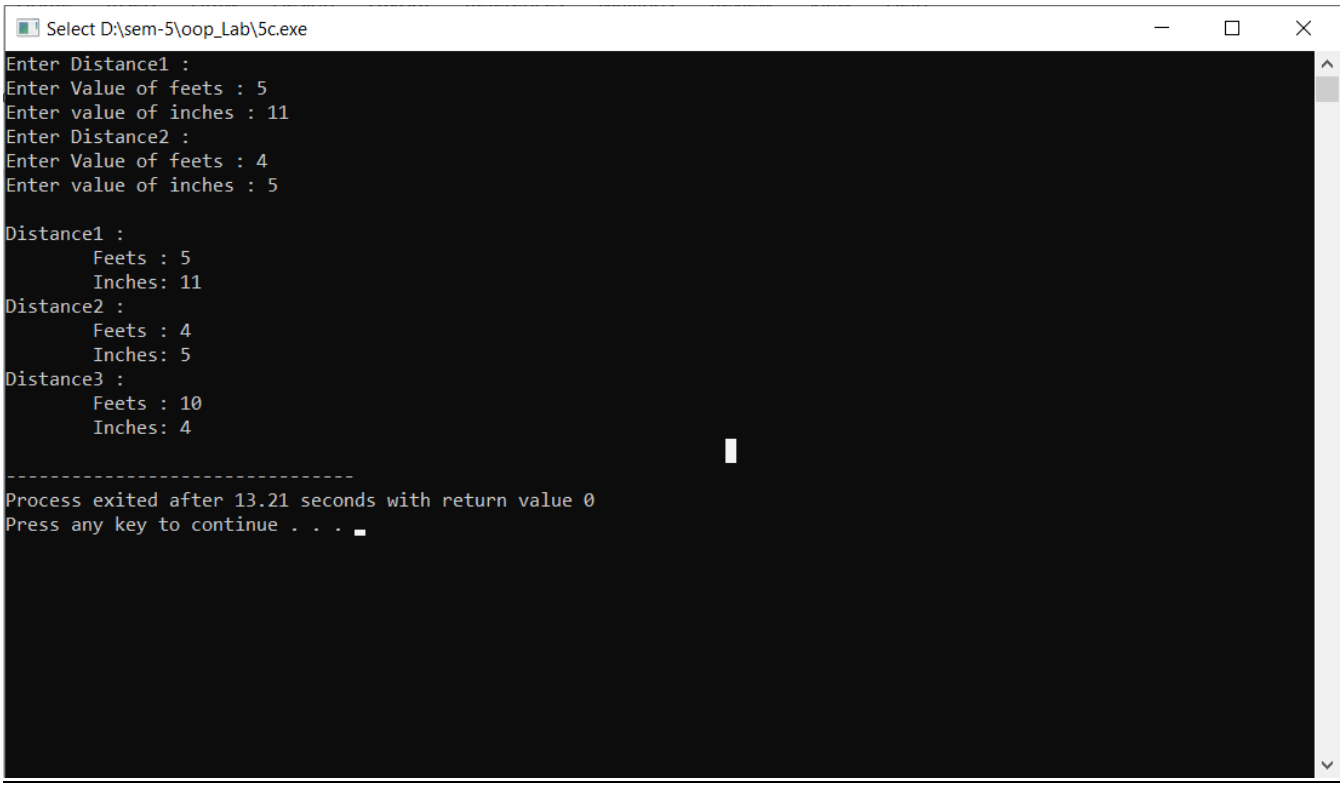
Source Code:

```
#include <iostream>
#include <cmath>
using namespace std;

class Distance
{
   private:
      int feet;
      int inch;
      Distance friend add(Distance,Distance);
   public:
      void set  ();
      void disp ();
};

void Distance:: set()
{
   cout << "Enter Value of feets : "; cin >> feet;
   cout << "Enter value of inches : "; cin >> inch;
```

```
}

void Distance:: disp()
{
   cout << endl << "\tFeets : " << feet;
   cout << endl << "\tInches: " << inch;
}

Distance add(Distance d1,Distance d2){
   Distance temp;
   int totdist = d2.inch+ d1.inch;
   temp.feet = totdist/12+ d2.feet + d1.feet;
   temp.inch = totdist%12;
   return temp;
}


int main()
{
   Distance d1;
   Distance d2;
   Distance d3;
   Distance d4;

   cout << "Enter Distance1 : " << endl;
```

```
    d1.set();

    cout << "Enter Distance2 : " << endl;
    d2.set();

    d3 = add(d1,d2);

    cout << endl << "Distance1 : " ;
    d1.disp();

    cout << endl << "Distance2 : " ;
    d2.disp();

    cout << endl << "Distance3 : " ;
    d3.disp();

    cout << endl;
    return 0;
}
```

Output:



P5 d. Write the definition for a class called time that has hours and minutes as integer. The class has the following member functions:

void settime(int, int) to set the specified value in object

void showtime() to display time object

time sum(time &,time &) to sum two time object & return time

1. Write the definitions for each of the above member functions.

2. Write main function to create three time objects. Set the value in two objects and call sum() to

calculate sum and assign it in third object. Display all time objects.

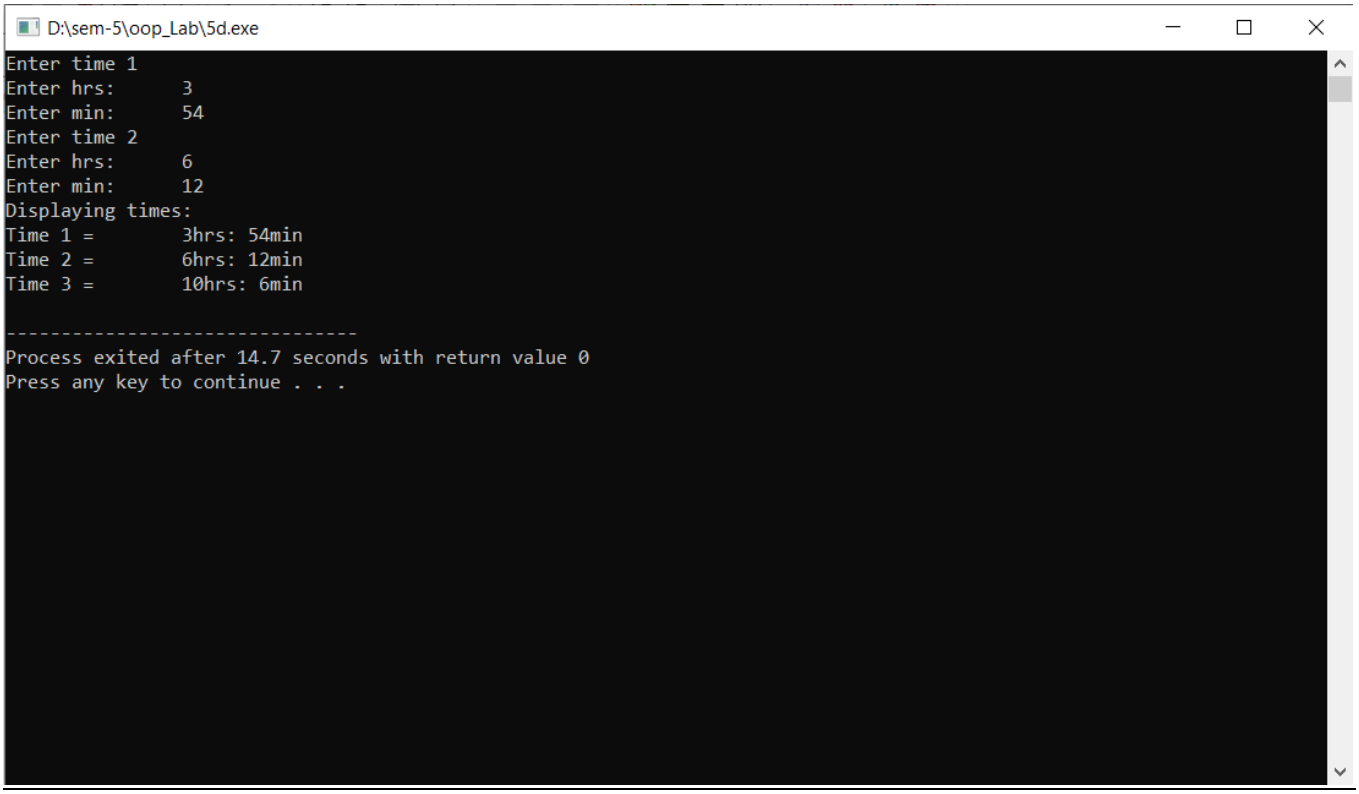
Source Code:

```
#include<iostream>


using namespace std;


class Time{
   private:
      int hrs,min;
      Time friend sum(Time&,Time&);
   public:
      void setTime(int h,int m){
         hrs=h%24;min=m;
      }
      void showTime(){
         cout<<"hrs: "<<hrs<<"min: "<<min<<endl;
      }
};


Time sum(Time &t1, Time &t2){
   Time t;
   t.hrs = (t1.hrs + t2.hrs + (t1.min + t2.min)/60)%24;
   t.min = (t1.min + t2.min)%60;
```

```
    return t;
}

int main(){
    Time t1,t2,t3;
    int hrs,min;
    cout<<"Enter time 1\nEnter hrs:\t";
    cin>>hrs;
    cout<<"Enter min:\t";
    cin>>min;
    t1.setTime(hrs,min);

    cout<<"Enter time 2\nEnter hrs:\t";
    cin>>hrs;
    cout<<"Enter min:\t";
    cin>>min;
    t2.setTime(hrs,min);

    t3 = sum(t1,t2);

    cout<<"Displaying times:\n";
    cout<<"Time 1 = \t";
    t1.showTime();

    cout<<"Time 2 = \t";
    t2.showTime();
```

```
    cout<<"Time 3 = \t";
    t3.showTime();
    return 0;
}
```

Output:



P6: A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise the message "Required copies not in stock" is displayed. Design a system using a class called Books with suitable member functions and Constructors. Implement C++ program for the system.

Source Code:

```
#include<iostream>
#include<string>
#include<iomanip>
#define M 5
using namespace std;

class Books{
   private:
```

```
        string author;
        string title;
        float price;
        string publisher;
        int stock_position;
        void friend add_to_array(Books &,Books,int);
        void friend bill(Books arr[],int);
        void friend search_by_author(Books arr[],string);
        void friend search_by_title(Books arr[],string);
        void friend show_all_books(Books arr[]);
    public:
        Books(){}
        Books(string a, string t, float pr, string pub, int st){
            author = a;
            title=t;
            price=pr;
            publisher=pub;
            stock_position = st;
        }
        bool match(string,string);
        void details();
        float query(int);
        void update_price(float);
};

bool Books::match(string t, string auth){
    if(title==t && auth==author) return true;
```

```
    return false;
}

void Books::details(){
    cout<<"Title: "<<title<<endl<<"Author: "<<author<<endl;
    cout<<"Publisher: "<<publisher<<endl<<"Price: "<<price<<endl;
}

float Books::query(int q){
    if(q<=stock_position){
        stock_position-=q;
        return q*price;
    }
    return 0.0;
}

void Books::update_price( float p){
    price=p;
}

void add_to_array(Books &a,Books b,int q){
    a=b;
    a.stock_position = q;
}

void bill(Books arr[],int n){
    float total=0;
```

```
    for(int i=0;i<n;i++){
        total+=arr[i].price * arr[i].stock_position;
        cout<<"Item: "<<i+1<<endl;
        cout<<"Title: "<<arr[i].title<<endl;
        cout<<"Author: "<<arr[i].author<<endl;
        cout<<"Price: "<<arr[i].price<<endl;
        cout<<"Quantity: "<<arr[i].stock_position<<endl;
    }
    cout<<"---------------------------------\nTotal: "<<total<<endl;
}

void show_all_books(Books arr[]){

cout<<"_____
_____\n";
    cout<<"|Title          |Author         |Publisher       |Price    |Stock Position|\n";

cout<<"_____
_____\n";
    for(int i=0;i<M;i++){

cout<<left<<"|"<<setw(18)<<arr[i].title<<"|"<<setw(17)<<arr[i].author<<"|"<<setw(18)<<arr[i].
publisher<<"|"<<setw(10)<<arr[i].price<<"|"<<setw(14)<<arr[i].stock_position<<"|\n";

cout<<"_____
_____\n";
    }
}

void search_by_author(Books arr[],string a){
```

```
    cout<<"_____\n";
    cout<<"|Title          |Publisher       |Price    |Stock Position|\n";
    cout<<"_____\n";
    for(int i=0;i<M;i++){
        if(arr[i].author==a){

cout<<left<<"|"<<setw(18)<<arr[i].title<<"|"<<setw(18)<<arr[i].publisher<<"|"<<setw(10)<<arr
[i].price<<"|"<<setw(14)<<arr[i].stock_position<<"|\n";

cout<<"_____\n";
        }
    }
}



void search_by_title(Books arr[],string a){
    cout<<"_____\n";
    cout<<"|Author          |Publisher       |Price    |Stock Position|\n";
    cout<<"_____\n";
    for(int i=0;i<M;i++){
        if(arr[i].title==a){

cout<<left<<"|"<<setw(18)<<arr[i].title<<"|"<<setw(18)<<arr[i].publisher<<"|"<<setw(10)<<arr
[i].price<<"|"<<setw(14)<<arr[i].stock_position<<"|\n";

cout<<"_____\n";
        }
    }
}
```

```
int main(){
    Books arr[M],bought[M];
    string author,title,publisher;
    float price;
    int stock,n=0;
    bool flag;
    cout<<"Eneter Details of books in stock.\n";
    for(int i=0;i<M;i++){
        cout<<"Details of book "<<i+1<<endl;
        cout<<"Enter Title\n";
        cin.clear();
        cin.sync();
        getline(cin,title);
        cout<<"Enter Author's name\n";
        getline(cin,author);
        cout<<"Enter Publisher\n";
        getline(cin,publisher);
        cout<<"Enter Price\n";
        cin>>price;
        cout<<"Enter Stock Position\n";
        cin>>stock;
        arr[i]=Books(author,title,price,publisher,stock);
    }
    while(1){
        int choice=0;
```

```
cout<<"Enter 1 to buy Book\nEnter 2 to see all book details\nEnter 3 to update price of a book\nEnter 4 to search all books by an author\nEnter 5 to search books by Title\nEnter 0 to exit.\n";
cin>>choice;
switch(choice){
  case 1:
    cout<<"Enter the Title and Author of the book\n";
    cin.clear();
    cin.sync();
    getline(cin,title);
    getline(cin,author);
    for(int i=0;i<M;i++){
      flag = arr[i].match(title,author);
      if(flag){
        int q,amt;
        cout<<"Congratulations!! We have the required book in stock.\n";
        arr[i].details();
        cout<<"Enter the Number of Books you want? \t";
        cin>>q;
        amt=arr[i].query(q);
        if(amt!=0){
          add_to_array(bought[n],arr[i],q);
          cout<<"Cost of books: "<<amt<<endl;
          n++;
        }else{
          cout<<"Stock Unavailable.\n";
          flag=false;
        }
```

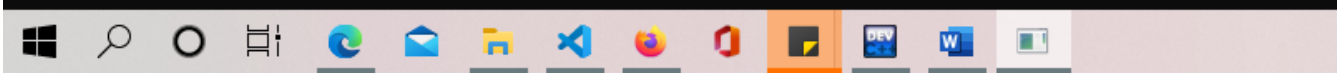
```
                break;
            }
        }
        if(!flag) cout<<"We Do not have the book you require\n";
        break;
    case 2:
        show_all_books(arr);
        break;
    case 3:
        cout<<"Enter the Title and Author of Book\n";
        cin.clear();
        cin.sync();
        getline(cin,title);
        getline(cin,author);
        flag=false;
        for(int i=0;i<M;i++){
            if(arr[i].match(title,author)){
                float price=0;
                flag=true;
                cout<<"Enter new price: ";
                cin>>price;
                arr[i].update_price(price);
                break;
            }
        }
        if(!flag){
            cout<<"Book not found"<<endl;
```

```
                }
                break;
            case 4:
                cout<<"Enter Name of Author: ";
                cin.clear();
                cin.sync();
                getline(cin,author);
                search_by_author(arr,author);
                break;
            case 5:
                cout<<"Enter Title of Book: ";
                cin.clear();
                cin.sync();
                getline(cin,title);
                search_by_title(arr,title);
                break;
            default:
                cout<<"Enter valid input\n";
                break;
        }
        if(choice==0) break;
    }
    bill(bought,n);
    return 0;
}
```

Output:

D:\sem-5\oop_Lab\6.exe

```
Eneter Details of books in stock.
Details of book 1
Enter Title
DS and Algo
Enter Author's name
Tanmay Vig
Enter Publisher
TV Corp
Enter Price
1000
Enter Stock Position
30
Details of book 2
Enter Title
Basics of ML
Enter Author's name
Tanmay Vig
Enter Publisher
bmp
Enter Price
550
Enter Stock Position
15
Details of book 3
Enter Title
DS and Algo
Enter Author's name
Chandu
Enter Publisher
Carl
Enter Price
300
Enter Stock Position
20
Details of book 4
Enter Title
Atomic Habits
Enter Author's name
James Clear
Enter Publisher
rh
Enter Price
500
Enter Stock Position
10
Details of book 5
Enter Title
Harry Potter
Enter Author's name
JK
```



D:\sem-5\oop_Lab\6.exe

```
Enter Stock Position
10
Details of book 5
Enter Title
Harry Potter
Enter Author's name
JK
Enter Publisher
Rowling
Enter Price
250
Enter Stock Position
9
Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title
Enter 0 to exit.
2
```

Title	Author	Publisher	Price	Stock Position
DS and Algo	Tanmay Vig	TV Corp	1000	30
Basics of ML	Tanmay Vig	bmp	550	15
DS and Algo	Chandu	Carl	300	20
Atomic Habits	James Clear	rh	500	10
Harry Potter	JK	Rowling	250	9

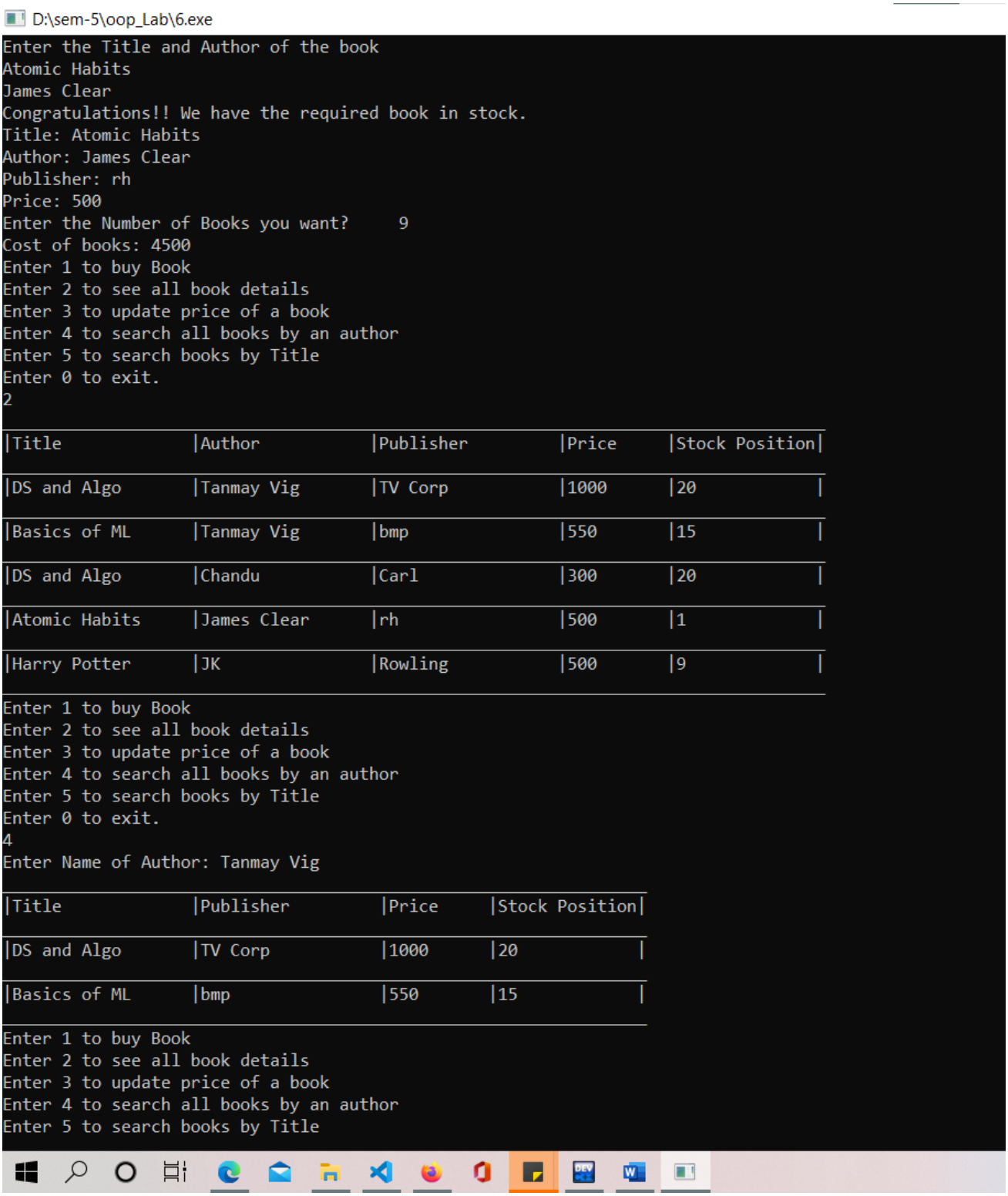
```
Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title
Enter 0 to exit.
3
Enter the Title and Author of Book
Harry Potter
JK
Enter new price: 500
Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title
Enter 0 to exit.
```

D:\sem-5\oop_Lab\6.exe

```
Enter the Title and Author of Book
Harry Potter
JK
Enter new price: 500
Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title
Enter 0 to exit.
2
```

Title	Author	Publisher	Price	Stock Position
DS and Algo	Tanmay Vig	TV Corp	1000	30
Basics of ML	Tanmay Vig	bmp	550	15
DS and Algo	Chandu	Carl	300	20
Atomic Habits	James Clear	rh	500	10
Harry Potter	JK	Rowling	500	9

```
Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title
Enter 0 to exit.
1
Enter the Title and Author of the book
DS and Algo
Tanmay Vig
Congratulations!! We have the required book in stock.
Title: DS and Algo
Author: Tanmay Vig
Publisher: TV Corp
Price: 1000
Enter the Number of Books you want?     10
Cost of books: 10000
Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title
Enter 0 to exit.
1
Enter the Title and Author of the book
Atomic Habits
```


D:\sem-5\oop_Lab\6.exe
Enter the Title and Author of the book
Atomic Habits
James Clear
Congratulations!! We have the required book in stock.
Title: Atomic Habits
Author: James Clear
Publisher: rh
Price: 500
Enter the Number of Books you want? 9
Cost of books: 4500
Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title
Enter 0 to exit.
2
Title	Author	Publisher	Price	Stock Position
DS and Algo	Tanmay Vig	TV Corp	1000	20
Basics of ML	Tanmay Vig	bmp	550	15
DS and Algo	Chandu	Carl	300	20
Atomic Habits	James Clear	rh	500	1
Harry Potter	JK	Rowling	500	9
Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title
Enter 0 to exit.
4
Enter Name of Author: Tanmay Vig
Title	Publisher	Price	Stock Position
DS and Algo	TV Corp	1000	20
Basics of ML	bmp	550	15
Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title

D:\sem-5\oop_Lab\6.exe

```
|DS and Algo       |TV Corp           |1000      |20            |
_____________________________________________________________
|Basics of ML      |bmp               |550       |15            |
_____________________________________________________________
Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title
Enter 0 to exit.
5
Enter Title of Book: DS and Algo
_____________________________________________________________
|Author             |Publisher         |Price     |Stock Position|
_____________________________________________________________
|DS and Algo       |TV Corp           |1000      |20            |
_____________________________________________________________
|DS and Algo       |Carl              |300       |20            |
_____________________________________________________________
Enter 1 to buy Book
Enter 2 to see all book details
Enter 3 to update price of a book
Enter 4 to search all books by an author
Enter 5 to search books by Title
Enter 0 to exit.
0
Enter valid input
Item: 1
Title: DS and Algo
Author: Tanmay Vig
Price: 1000
Quantity: 10
Item: 2
Title: Atomic Habits
Author: James Clear
Price: 500
Quantity: 9
--------------------------------
Total: 14500

--------------------------------
Process exited after 339.5 seconds with return value 0
Press any key to continue . . .
```

P7: Write a program in C++ to implement a class Employee using following concepts:

Pointer to Objects, Dynamic Memory Allocation using new and delete, Dynamic constructors, Destructors

Source Code:

```
#include<iostream>
#include<string>
#include<iomanip>
using namespace std;


class Employee{
   private:
      // int holidays;
      long long int id,salary;
      string name;
      int friend find(Employee *arr[], int n, string name,long long int id, bool mode_id, bool print)
;
      void friend display(Employee *arr[],int , bool );
      void friend sort(Employee *arr[],int n,int form);
      void friend del(Employee *arr[], int n,int pos);
   public:
      Employee(){}
      Employee(int i, string n,long long int s){
         id = i;
         name = n;
         salary = s;
```

```
            // holidays = h;
        }
};

int find(Employee *arr[],int n,string name,long long int id,bool mode_id, bool print){
    bool found=false;
    for(int i=0;i<n;i++){
        if((!mode_id && arr[i]->name==name) || (mode_id && arr[i]->id == id)){
            found = true;
            if(print)display(arr,i,found);
            return i;
            break;
        }
    }
    if(!found) cout<<"No related data found.\n";
    return -1;
}

void display(Employee *arr[],int n, bool single){
    cout<<"---------------------------------------\n";
    cout<<"|id       |Name          |Salary    |\n";
    cout<<"---------------------------------------\n";
    if(single){
        cout<<left<<"|"<<setw(10)<<(*arr[n]).id<<"|"<<setw(15)<<(*arr[n]).name<<"|"<<setw(10)
<<(*arr[n]).salary<<"|\n";
        cout<<"---------------------------------------\n";
    }else{
```

```
        for(int i=0;i<n;i++){
            cout<<left<<"|"<<setw(10)<<(*arr[i]).id<<"|"<<setw(15)<<(*arr[i]).name<<"|"<<setw(10
)<<(*arr[i]).salary<<"|\n";
            cout<<"---------------------------------------\n";
        }
    }
}

void sort(Employee *arr[],int n,int form){
    for(int i=0;i<n-1;i++){
        for(int j=0;j<n-i-1;j++){
            if((form == 1 && arr[j]->id>arr[j+1]->id) || (form==2 && arr[j]->name.compare(arr[j+1]-
>name)>0) || (form==3 && arr[j]->salary>arr[j+1]->salary)){
                Employee temp = *arr[j];
                *arr[j] = *arr[j+1];
                *arr[j+1] = temp;
            }

        }
    }
    display(arr,n,false);

}

void del(Employee *arr[],int n,int pos){
    for(int i=pos;i<n-1;i++){
        Employee temp = *arr[i];
        *arr[i] = *arr[i+1];
```

```
            *arr[i+1] = temp;
        }
        cout<<"Deleted\n";
    }

    int main(){
        int n=5;
        Employee *arr[100];
        cout<<"Enter the employee Details:\n";
        for(int i=0;i<n;i++){
            long long int id, salary; string name;
            cout<<"Enter the details of employee "<<(i+1)<<endl;
            cout<<"Enter id, name salary\n";
            cin>>id;
            cin.ignore();
            getline(cin,name);
            cin>>salary;
            arr[i] = new Employee(id,name,salary);
        }
        while(true){
            int choice=0,s=0;
            long long int id,salary;
            string name;
            cout<<"to display data of all employees Enter 1\nto display data of one employee Enter 2\
nto sort data Enter 3\nto delete Enter 4\nto insert new employee Enter 5\nto exit Enter 0\n";
            cin>>choice;
            switch (choice)
```

```
        {
        case 2:
           cout<<"to search by name Enter 1\nto search by id Enter 2\n";
           cin>>s;
           if(s==1){
              cout<<"Enter name: ";
              cin.clear();
              cin.sync();
              getline(cin,name);
              find(arr,n,name,-1,false,true);
           }else{
              cout<<"Enter id: ";
              cin>>s;
              find(arr,n,"",s,true,true);
           }
           break;
        case 1:
           display(arr,n,false);
           break;
        case 3:
           cout<<"to sort by id Enter 1\nto sort by name Enter 2\nto sort by salary Enter 3\n";
           cin>>s;
           sort(arr,n,s);
           break;
        case 4:
           cout<<"to find by id Enter 1\nto find by name Enter 2\n";
           cin>>s;
```

```
            if(s==1){
                cout<<"Enter id: ";cin>>s;
                s=find(arr,n,"",s,true,false);
            }else{
                cin.clear();
                cin.sync();
                getline(cin,name);
                s=find(arr,n,name,-1,false,false);
            }
            if(s!=-1){
                del(arr,n,s);
                n--;
            }
            break;
        case 5:
            cout<<"Enter id, name, salary\n";
            cin>>id;
            cin.ignore();
            getline(cin,name);
            cin>>salary;
            arr[n++] = new Employee(id,name,salary);
            break;
        default:
            choice =0;
            break;
        }
        if(!choice) break;
```

```
    }

    return 0;

}
```

Output:

```
123112
Enter the details of employee 5
Enter id, name salary
104
faisal ahmed
11121
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
1
---------------------------------------
|id        |Name           |Salary    |
---------------------------------------
|100       |tanmay vig     |100000    |
---------------------------------------
|101       |almas          |1020203   |
---------------------------------------
|102       |amir javed     |120030    |
---------------------------------------
|103       |surya          |123112    |
---------------------------------------
```

```
PS D:\sem-5\oop_Lab> cd "d:\sem-5\oop_Lab\" ; if ($?) { g++ 7.cpp -o 7 } ; if ($?) { .\7 }
Enter the employee Details:
Enter the details of employee 1
Enter id, name salary
100
tanmay vig
100000
Enter the details of employee 2
Enter id, name salary
101
almas
1020203
Enter the details of employee 3
Enter id, name salary
102
amir javed
120030
Enter the details of employee 4
Enter id, name salary
103
surya
123112
Enter the details of employee 5
Enter id, name salary
```

```
---------------------------------------
|id        |Name           |Salary    |
---------------------------------------
|100       |tanmay vig     |100000    |
---------------------------------------
|101       |almas          |1020203   |
---------------------------------------
|102       |amir javed     |120030    |
---------------------------------------
|103       |surya          |123112    |
---------------------------------------
|104       |faisal ahmed   |11121     |
---------------------------------------
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
2
to search by name Enter 1
to search by id Enter 2
1
Enter name: almas
```

```
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
2
to search by name Enter 1
to search by id Enter 2
1
Enter name: almas
---------------------------------------
|id        |Name           |Salary    |
---------------------------------------
|101       |almas          |1020203   |
---------------------------------------
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
2
to search by name Enter 1
```

```
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
2
to search by name Enter 1
to search by id Enter 2
103
Enter id: 103
---------------------------------------
|id        |Name           |Salary    |
---------------------------------------
|103       |surya          |123112    |
---------------------------------------
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
3
to sort by id Enter 1
to sort by name Enter 2
to sort by salary Enter 3
```

```
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
3
to sort by id Enter 1
to sort by name Enter 2
to sort by salary Enter 3
2
--------------------------------------
|id        |Name           |Salary    |
--------------------------------------
|101       |almas          |1020203   |
--------------------------------------
|102       |amir javed     |120030    |
--------------------------------------
|104       |faisal ahmed   |11121     |
--------------------------------------
|103       |surya          |123112    |
--------------------------------------
|100       |tanmay vig     |100000    |
--------------------------------------
to display data of all employees Enter 1
```

```
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
3
to sort by id Enter 1
to sort by name Enter 2
to sort by salary Enter 3
3
--------------------------------------
|id        |Name           |Salary    |
--------------------------------------
|104       |faisal ahmed   |11121     |
--------------------------------------
|100       |tanmay vig     |100000    |
--------------------------------------
|102       |amir javed     |120030    |
--------------------------------------
|103       |surya          |123112    |
--------------------------------------
|101       |almas          |1020203   |
--------------------------------------
```

```
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
3
to sort by id Enter 1
to sort by name Enter 2
to sort by salary Enter 3
1
--------------------------------------
|id        |Name           |Salary    |
--------------------------------------
|100       |tanmay vig     |100000    |
--------------------------------------
|101       |almas          |1020203   |
--------------------------------------
|102       |amir javed     |120030    |
--------------------------------------
|103       |surya          |123112    |
--------------------------------------
|104       |faisal ahmed   |11121     |
--------------------------------------
```

```
--------------------------------------
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
4
to find by id Enter 1
to find by name Enter 2
1
Enter id: 103
Deleted
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
1
--------------------------------------
|id        |Name           |Salary    |
--------------------------------------
|100       |tanmay vig     |100000    |
```

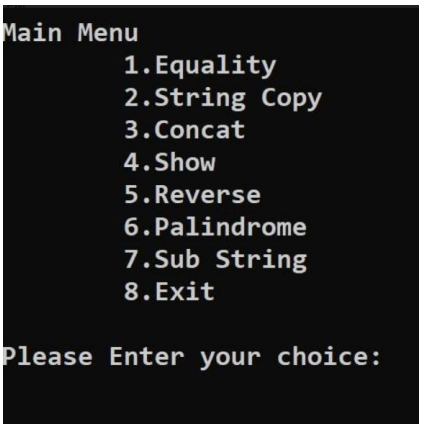
```
--------------------------------------
|id        |Name           |Salary    |
--------------------------------------
|100       |tanmay vig     |100000    |
--------------------------------------
|101       |almas          |1020203   |
--------------------------------------
|102       |amir javed     |120030    |
--------------------------------------
|104       |faisal ahmed   |11121     |
--------------------------------------
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
4
to find by id Enter 1
to find by name Enter 2
2
almas
Deleted
to display data of all employees Enter 1
```

```
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
1
--------------------------------------
|id        |Name           |Salary    |
--------------------------------------
|100       |tanmay vig     |100000    |
--------------------------------------
|102       |amir javed     |120030    |
--------------------------------------
|104       |faisal ahmed   |11121     |
--------------------------------------
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
5
Enter id, name, salary
```

```
5
Enter id, name, salary
105
almas
10020
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
1
---------------------------------------
|id        |Name           |Salary    |
---------------------------------------
|100       |tanmay vig     |100000    |
---------------------------------------
|102       |amir javed     |120030    |
---------------------------------------
|104       |faisal ahmed   |11121     |
---------------------------------------
|105       |almas          |10020     |
---------------------------------------
to display data of all employees Enter 1
```

```
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
1
---------------------------------------
|id        |Name           |Salary    |
---------------------------------------
|100       |tanmay vig     |100000    |
---------------------------------------
|102       |amir javed     |120030    |
---------------------------------------
|104       |faisal ahmed   |11121     |
---------------------------------------
|105       |almas          |10020     |
---------------------------------------
to display data of all employees Enter 1
to display data of one employee Enter 2
to sort data Enter 3
to delete Enter 4
to insert new employee Enter 5
to exit Enter 0
0
```

P8: Use operator overloading to do complete the following functions:



Source Code:

```
#include<iostream>
#include<cstring>
using namespace std;

class String{
   private:
      char str[30];
   public:
      void input();
      void show();
      void operator==(String s);
      void operator=(String s);
      void operator+(String s);
      void operator<<(String s);
      void operator>>(String s);
      void operator~();
```

```
        int operator/(String s);
};

void String::input(){
   cout<<"Enter String: ";
   cin.clear();
   cin.sync();
   cin.getline(str,30);
}

void String::show(){
   for(int i=0;str[i]!='\0';i++){
      cout<<str[i];
   }
}

void String::operator==(String s){
   int i=0;
   bool equal=true;
   while(*(str+i)!='\0' && *(s.str + i)!='\0' ){
      if(*(str+i)!=*(s.str+i)){
         equal=false;
         break;
      }
      i++;
   }
   if(equal && *(str+i)!='\0' && *(s.str+i)!='\0') cout<<"Both String are equal\n";
```

```
    else cout<<"Strings are not equal\n";
}

void String::operator=(String s){
    int i=0,len= sizeof(s.str)/sizeof(char);
    for(int i=0;i<len;i++){
        *(str+i)= *(s.str+i);
        // str[i]=s.str[i];
    }
    *(str+len )='\0';
}

void String::operator+(String s){
    int len=0;
    while(this->str[len]!='\0') len++;
    for(int i=0;s.str[i]!='\0';++i,++len){
        this->str[len] = s.str[i];
    }
    this->str[len] = '\0';
}

void String::operator<<(String s){
    cout<<"Sring:\t";
    s.show();
    cout<<"\n";
}
```

```
void String::operator>>(String s){
   cout<<"Reverse String:\t";
   for(int i=strlen(s.str);i>=0;i--){
      cout<<s.str[i];
   }
   cout<<"\n";
}

void String::operator~(){
   bool ans=true;
   int len = strlen(str)-1;
   for(int i=0;i<=len/2;i++){
      // cout<<str[i]<<" "<<str[len-i]<<endl;
      if(str[i]!=str[len-i]){
         ans=false;
         break;
      }
   }
   if(ans) cout<<"Yes It is palindrome\n";
   else cout<<"No it is not palindrome\n";
}

int String::operator/(String s)
{
  int flag=0,k,i,j,len=strlen(str),len1=strlen(s.str)-1;
  for(i=0;i<len;i++)
   {
```

```
        if(str[i]==s.str[0])
        {
         if(str[i+len1]==s.str[len1])
           {
            for(j=i,k=0;j<i+len1+1,k<len1;j++,k++)
              {
               if(str[j]==s.str[k])
                 flag=1;
               else
                {
                  flag=0;
                  break;
                }
              }
           }
        }
     }
    if(flag==0)
      return 0;  //not a substring

      return 1;  //it is a substring
}

int main(){
    while(true){
        int choice=8,br=0,c;
        String a,b;
```

```
        a.input();
        b.input();
        cout<<"Main Menu\n\t1. Equality\n\t2. String copy\n\t3. Concat\n\t4. Show\n\t5.
    Reverse\n\t6. Palindrome\n\t7. Sub String\n\t8. Exit\nPlease enter your choice\t";
        cin>>choice;
        switch (choice)
        {
        case 1:
           a==b;
           break;
        case 2:
           b=a;
           cout<<"The String is copied: ";
           b.show();
           cout<<endl;
           break;
        case 3:
           a+b;
           a.show();
           cout<<"\n";
           break;
        case 4:
           b<<a;
           break;
        case 5:
           b>>a;
           break;
```

```
        case 6:
           ~a;
           break;
        case 7:
           c=a/b;
           if(c){
              cout<<"YES\n";
           }else cout<<"NO\n";
           break;
        case 8:
           br=1;
           break;
        default:
           cout<<"Enter Valid option\n";
           break;
        }
        if(br) break;
    }
    return 0;
}
```

Output:

```
PS D:\sem-5\oop_Lab> cd "d:\sem-5\oop_Lab\" ; if ($?) { g++ 8.cpp -o 8 } ; if ($?) { .\8 }
Enter String: level
Enter String: q
Main Menu
        1. Equality
        2. String copy
        3. Concat
        4. Show
        5. Reverse
        6. Palindrome
        7. Sub String
        8. Exit
Please enter your choice        6
Yes It is palindrome
Enter String: tanmay
Enter String: a
Main Menu
        1. Equality
        2. String copy
        3. Concat
        4. Show
        5. Reverse
        6. Palindrome
        7. Sub String
        8. Exit
Please enter your choice        6
No it is not palindrome
```

P9: Write a program to implement a class Stack using Linked List.

Source code:

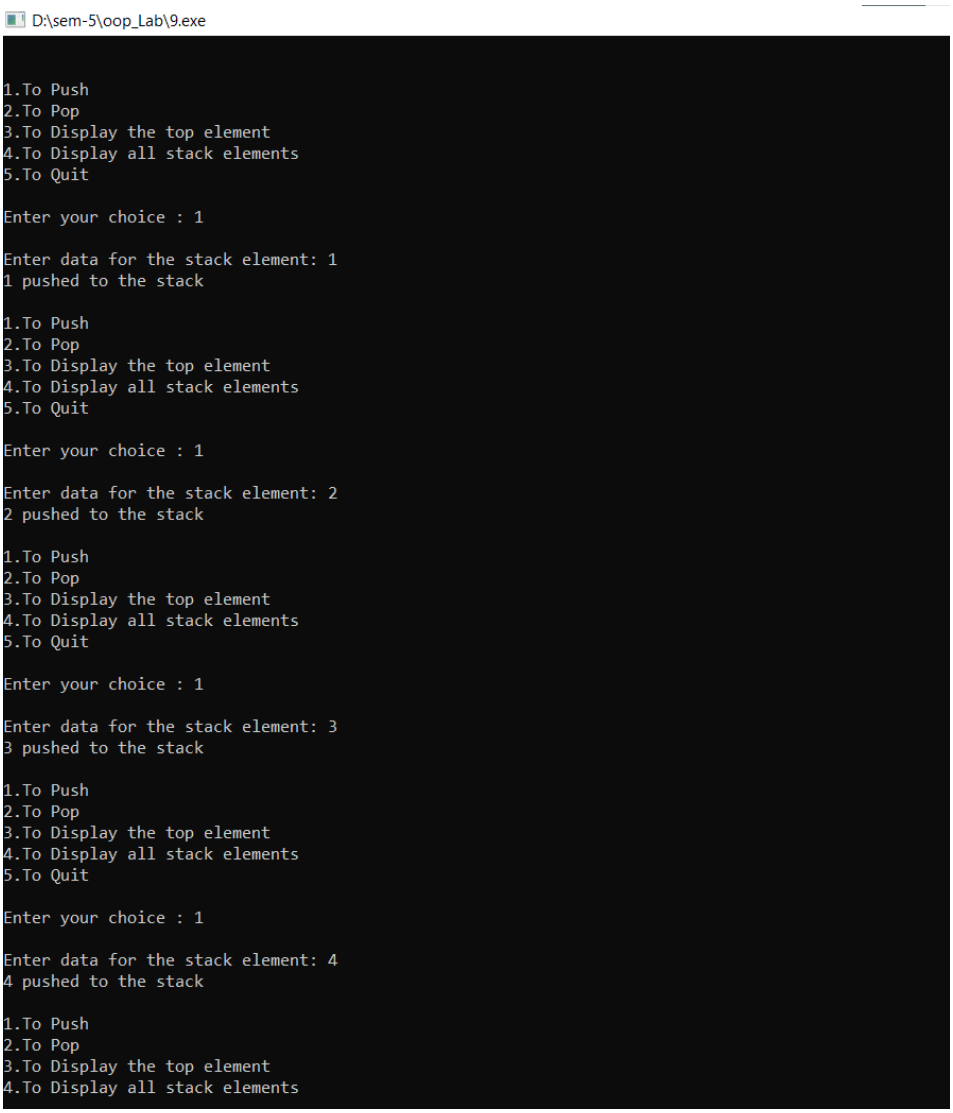
```
#include <iostream>
using namespace std;

class Stak_Node {
private:
        int data;
        Stak_Node* next;
public:
   Stak_Node* newNode(int);
   void push(Stak_Node**, int);
   int pop(Stak_Node**);
   int isempty(Stak_Node*);
   int front(Stak_Node*);
   void print(Stak_Node*);
};

Stak_Node* Stak_Node::newNode(int data){
        Stak_Node* stackNode = new Stak_Node();
        stackNode->data = data;
        stackNode->next = NULL;
        return stackNode;
}
```

```
void Stak_Node::push(Stak_Node** root, int data){
        Stak_Node* stackNode = newNode(data);
        stackNode->next = *root;
        *root = stackNode;
   cout<<data<<" pushed to the stack";
}

int Stak_Node::pop(Stak_Node** root){
        if (isempty(*root))
                return INT_MIN;

        Stak_Node* temp = *root;
        *root = (*root)->next;
        int popped = temp->data;
        return popped;
}

int Stak_Node::isempty(Stak_Node* root){
        return !root;
}

int Stak_Node::front(Stak_Node* root){
        if (isempty(root))
                return INT_MIN;
        return root->data;
}
```

```
void Stak_Node::print(Stak_Node* root){
   cout<<"Elements present in stack are (top to bottom) : ";
   while(!isempty(root)) {
      cout<<front(root)<<"\t";
                  pop(&root);
         }
}

int main(){
         Stak_Node* root = NULL;
   int choice,item;

   while(1){
      cout<<"\n\n1.To Push\n2.To Pop\n3.To Display the top element\n4.To Display all stack elements\n5.To Quit\n\nEnter your choice : ";
      cin>>choice;
      switch(choice){
       case 1 :
            cout<<"\nEnter data for the stack element: ";
            cin>>item;
            root->push(&root, item);
            break;
       case 2:
            cout<<root->pop(&root)<<" is popped from stack\n";
            break;
       case 3:
```

```
            cout<<"\n"<<root->front(root)<<" is on the top\n"<<endl;
            break;
        case 4:
            root->print(root);
            break;
        case 5:
            exit(1);
        default:
            cout<<"\nINCORRECT INPUT\n";
       }
    }

        return 0;
}
```

Output:


D:\sem-5\oop_Lab\9.exe
1.To Push
2.To Pop
3.To Display the top element
4.To Display all stack elements
5.To Quit
Enter your choice : 1
Enter data for the stack element: 1
1 pushed to the stack
1.To Push
2.To Pop
3.To Display the top element
4.To Display all stack elements
5.To Quit
Enter your choice : 1
Enter data for the stack element: 2
2 pushed to the stack
1.To Push
2.To Pop
3.To Display the top element
4.To Display all stack elements
5.To Quit
Enter your choice : 1
Enter data for the stack element: 3
3 pushed to the stack
1.To Push
2.To Pop
3.To Display the top element
4.To Display all stack elements
5.To Quit
Enter your choice : 1
Enter data for the stack element: 4
4 pushed to the stack
1.To Push
2.To Pop
3.To Display the top element
4.To Display all stack elements

```
D:\sem-5\oop_Lab\9.exe
5.To Quit

Enter your choice : 1

Enter data for the stack element: 4
4 pushed to the stack

1.To Push
2.To Pop
3.To Display the top element
4.To Display all stack elements
5.To Quit

Enter your choice : 3

4 is on the top



1.To Push
2.To Pop
3.To Display the top element
4.To Display all stack elements
5.To Quit

Enter your choice : 2
4 is popped from stack


1.To Push
2.To Pop
3.To Display the top element
4.To Display all stack elements
5.To Quit

Enter your choice : 4
Elements present in stack are (top to bottom) : 3       2       1

1.To Push
2.To Pop
3.To Display the top element
4.To Display all stack elements
5.To Quit

Enter your choice : 5

--------------------------------
Process exited after 36.85 seconds with return value 1
Press any key to continue . . .
```

P10: Implementation of Inheritance

Source Code

```
#include <iostream>
#include <string>

using namespace std;

class Employee{
private:
   string name;
   string id;
   int salary;

public:
   Employee() {}
   Employee(string n, string i, int s){
      id=i;
      name=n;
      salary = s;
   }
   ~Employee(){
   }
   void getEmployee(){
      cout<<"Employee id: "<<id<<"\nEmployee Name: "<<name<<"\nEmployee salary: "<<salary<<endl;
   }
};
```

```
class Scientist: protected Employee{
private:
   int num_publications;
   int num_awards;
   string *publications;
   string *awards;
public:
   Scientist() {}
   Scientist(int num_pub, string *pub, string *aw,int n_a, string name, string id, int
salary):Employee(name,id,salary){
      num_publications = num_pub;
      num_awards = n_a;
      publications = new string[num_publications];
      for(int i=0;i<num_publications;i++){
         publications[i] =pub[i];
      }
      awards = new string[n_a];
      for(int i=0;i<n_a;i++){
         awards[i] =aw[i];
      }
   }
   ~Scientist(){
      delete[] publications;
      delete[] awards;
   }
   void getScientist(){
```

```
        cout<<"\nScientist Details:\n";
        getEmployee();
        cout<<"\nTotal publications: "<<num_publications<<endl;
        cout<<"Publications:\n";
        for(int i=0;i<num_publications;i++){
           cout<<" "<<publications[i]<<endl;
        }
        cout<<"\nTotal Awards: "<<num_awards<<endl;
        cout<<"Awards:\n";
        for(int i=0;i<num_awards;i++){
           cout<<" "<<awards[i]<<endl;
        }cout<<"\n";
    }

};

class Manager : protected Employee{
private:
    string title;
    int yrs_of_exp;
    int teams;
public:
    Manager(){}
    Manager(string t, int yrs, int tm,string name, string id, int salary):Employee(name,id,salary){
        title=t;
        yrs_of_exp=yrs;
        teams = tm;
```

```
    }
    ~Manager(){
    }
    void getManager(){
        cout<<"\nManager Details: \n";
        getEmployee();
        cout<<"Title as Manager: "<<title<<endl;
        cout<<"Years of Experience: "<<yrs_of_exp<<endl;
        cout<<"Teams Managed: "<<teams<<endl;
    }
};

class Laborer : protected Employee{
private:
    int overtime;
    int wage_over;
    int leaves;
public:
    Laborer(){}
    Laborer(int o, int w_o, int l,string name, string id, int salary) :Employee(name,id,salary){
        overtime = o;
        wage_over = w_o;
        leaves = l;
    }
    ~Laborer(){
    }
    void getLaborer(){
```

```
        cout<<"\nLabourer Details: \n"<<endl;
        getEmployee();
        cout<<"Overtime: "<<overtime<<endl;
        cout<<"Wage in overtime: "<<wage_over<<endl;
        cout<<"Total leaves: "<<leaves<<endl;
    }
};

int main(){

    int sci_num=0,man_num=0, lab_num=0;
    Scientist *arr_s[100];
    Manager *arr_m[100];
    Laborer *arr_l[100];

    while(true){
        int choice=0,c, yrs, teams;
        int salary;
        int n_p,n_a,n_o, wage, leaves;
        string name, id,title;
        cout<<"\nEnter:\n 1 to add Scientist\n 2 to add Manager\n 3 to add Laborer\n 4 to show
all Scientists\n 5 to show all Managers\n 6 to show all Laborer\n any num to exit\n";
        cin>>choice;
        switch(choice){
            case 1:
            {
                sci_num++;
```

```
cout<<"Enter Employee ID, Name and Salary\n";
cin.clear();
cin.sync();
getline(cin, id);
getline(cin, name);
cin>>salary;
cout<<"Enter number of publications\t";
cin>>n_p;
string *a_p = new string[n_p];
cout<<"Enter publications name\n";
for(int i=0;i<n_p;i++){
   string s;
   cin.clear();
   cin.sync();
   getline(cin, s);
   a_p[i]=s;
}
cout<<"Enter number of awards\t";
cin>>n_a;
string *a_a = new string[n_a];
cout<<"Enter awards name\n";
for(int i=0;i<n_a;i++){
   string s;
   cin.clear();
   cin.sync();
   getline(cin, s);
   a_a[i] = s;
```

```
            }
            arr_s[sci_num-1] = new Scientist(n_p,a_p,a_a, n_a,name,id, salary);
            break;
        }
        case 4:
        {
            for(int i=0;i<sci_num;i++){
                (*arr_s[i]).getScientist();
            }
            break;
        }
        case 2:
        {
            man_num++;
            cout<<"Enter Employee ID, Name and Salary\n";
            cin.clear();
            cin.sync();
            getline(cin, id);
            getline(cin, name);
            cin>>salary;
            cout<<"Choose any position:\n 1. General Manager\n 2. Assistant Manager\n 3.
Production Manager\n";
            cin>>c;
            // cin.clear();
            // cin.sync();
            switch(c){
                case 1:
```

```
            title = "General Manager";
            break;
        case 2:
            title = "Assistant Manager";
            break;
        case 3:
            title="Production Manager";
            break;
        default:
            title="Unknown Manager";
            break;
    }
    cout<<"Enter number of teams managed\t";
    cin>>teams;
    cout<<"Enter numbers of years of experience\t";
    cin>>yrs;
    arr_m[man_num-1] = new Manager(title,yrs,teams, name,id,salary);
    break;
}
case 5:
{
    for(int i=0;i<man_num;i++){
        (*arr_m[i]).getManager();
    }
    break;
}
case 3:
```

```
        {
            lab_num++;
            cout<<"Enter Employee ID, Name and Salary\n";
            cin.clear();
            cin.sync();
            getline(cin, id);
            getline(cin, name);
            cin>>salary;
            cout<<"Enter overtime hours\t";
            cin>>n_o;
            cout<<"Enter wage in overtime\t";
            cin>>wage;
            cout<<"Enter number of leaves\t";
            cin>>leaves;
            arr_l[lab_num-1] = new Laborer(n_o,wage,leaves,name, id, salary);
            break;
        }
        case 6:
        {
            for(int i=0;i<lab_num;i++){
                (*arr_l[i]).getLaborer();
            }
            break;
        }
        default:
        {
            choice = 0;
```

```
                break;
            }
        }
        if(choice == 0) break;
    }

    return 0;
}
```

Output:

```
Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
 any num to exit
1
Enter Employee ID, Name and Salary
100
Mudit Malhotra
20000
Enter number of publications    1
Enter publications name
one
Enter number of awards  1
Enter awards name
alpha

Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
 any num to exit
1
Enter Employee ID, Name and Salary
101
Karan Kapoor
3000
Enter number of publications    0
Enter publications name
Enter number of awards  1
Enter awards name
theta

Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
 any num to exit
2
```

```
Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
 any num to exit
2
Enter Employee ID, Name and Salary
102
Tanmay Vig
1000000
Choose any position:
 1. General Manager
 2. Assistant Manager
 3. Production Manager
1
Enter number of teams managed   8
Enter numbers of years of experience    5

Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
 any num to exit
6

Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
 any num to exit
3
Enter Employee ID, Name and Salary
103
Paras
10000
Enter overtime hours    6
Enter wage in overtime  100
Enter number of leaves  4
```

D:\sem-5\oop_Lab\10.exe

```
10000
Enter overtime hours    6
Enter wage in overtime  100
Enter number of leaves  4

Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
 any num to exit
2
Enter Employee ID, Name and Salary
104
Manuj Monga
100000
Choose any position:
 1. General Manager
 2. Assistant Manager
 3. Production Manager
2
Enter number of teams managed   5
Enter numbers of years of experience    4

Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
 any num to exit
4

Scientist Details:
Employee id: 100
Employee Name: Mudit Malhotra
Employee salary: 20000

Total publications: 1
Publications:
 one

Total Awards: 1
Awards:
 alpha
```

D:\sem-5\oop_Lab\10.exe

```
Total Awards: 1
Awards:
 alpha


Scientist Details:
Employee id: 101
Employee Name: Karan Kapoor
Employee salary: 3000

Total publications: 0
Publications:

Total Awards: 1
Awards:
 theta


Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
 any num to exit
5

Manager Details:
Employee id: 102
Employee Name: Tanmay Vig
Employee salary: 1000000
Title as Manager: General Manager
Years of Experience: 5
Teams Managed: 8

Manager Details:
Employee id: 104
Employee Name: Manuj Monga
Employee salary: 100000
Title as Manager: Assistant Manager
Years of Experience: 4
Teams Managed: 5

Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
```

D:\sem-5\oop_Lab\10.exe

```
Teams Managed: 8

Manager Details:
Employee id: 104
Employee Name: Manuj Monga
Employee salary: 100000
Title as Manager: Assistant Manager
Years of Experience: 4
Teams Managed: 5

Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
 any num to exit
6

Labourer Details:

Employee id: 103
Employee Name: Paras
Employee salary: 10000
Overtime: 6
Wage in overtime: 100
Total leaves: 4

Enter:
 1 to add Scientist
 2 to add Manager
 3 to add Laborer
 4 to show all Scientists
 5 to show all Managers
 6 to show all Laborer
 any num to exit
0

--------------------------------
Process exited after 224.7 seconds with return value 0
Press any key to continue . . .
```

P11: A company pays its employees on a weekly basis. The employees are of four types: Salaried employees are paid a fixed weekly salary regardless of the number of hours worked, hourly employees are paid by the hour and receive overtime pay (i.e., 1.5 times their hourly salary rate) for all hours worked in excess of 40 hours, commission employees are paid a percentage of their sales and base-salaried commission employees receive a base salary plus a percentage of their sales. For the current pay period, the company has decided to reward salaried commission employees by adding 10% to their base salaries. Write a c++ program that performs payroll calculations of the company polymorphically.

Source Code:

```
#include<iostream>
#include<string>

using namespace std;

class Employee {
   protected:
      string name;
      float salary;
   public:
      Employee(){ }
```

```
        Employee(string n, float s){
           name=n;salary=s;
        }
        virtual void getDetail(){
           cout<<"\nEmployee Name: "<<name<<"\nSalary: "<<salary<<"\n\n";
        }
        virtual float getSalary(){
           return salary;
        }
};

class SalariedEmployee : public Employee {
   public:
      SalariedEmployee(string name, float salary): Employee(name, salary){ }
      void getDetail(){
         cout<<"\nEmployee Name: "<<name<<"\nType: "<<"Salaried\nSalary:
"<<salary<<"\n\n";
      }
};

class HourlyEmployee : public Employee
{
   protected:
      int hours;
   public:
      HourlyEmployee(string name, float salary, int hrs): Employee(name, salary){
         hours=hrs;
```

```
        }
        float getsalary(){
           if(hours<=40) return salary*hours;
           else return salary*40 +1.5*salary*(hours-40);
        }
        void getDetail(){
           cout<<"\nEmployee Name: "<<name<<"\nType: "<<"Hourly\nSalary:
"<<getsalary()<<"\n\n";
        }
};

class CommissionEmployee : public Employee {
   protected:
      float commission;
   public:
      CommissionEmployee(string name,float salary,float c):Employee(name, salary){
         commission = c;
      }
      virtual float getSalary(){
         return salary*commission/100;
      }
      virtual void getDetail(){
         cout<<"\nEmployee Name: "<<name<<"\nType: Commissioned\nSalary:
"<<getSalary()<<"\n\n";
      }

};
```

```
class BasePlusCommissionEmployee : public CommissionEmployee{
   protected:
      float base_amount;
      float increment;
   public:
      BasePlusCommissionEmployee(string name, float salary, float base, float c, float
incr):CommissionEmployee(name, salary,c){
         base_amount = base;
         increment = incr;
      }
      float getSalary(){
         return base_amount + base_amount*increment/100 + salary*commission/100;
      }
      void getDetail(){
         cout<<"\nEmployee Name: "<<name<<"\nType: Commissioned\nSalary: "<<
getSalary()<<"\n\n";
      }

};

int main(){
   Employee* arr[100];
   int count=0;
   while(true){
      int choice = 0;
      cout<<"Enter:\n 1. to create a Salaried Employee\n 2. to create a Hourly Employee\n 3. to
create a Comission Employee\n 4. to create a Base Plus Commission Employee\n 5. to get
payroll of all employees\n any other number to exit\n";
```

```
        cin>>choice;
        string name;
        float salary;
        switch(choice){
           case 1:{
              count++;
              cout<<"Enter name: ";
              cin.clear();
              cin.sync();
              getline(cin,name);
              cout<<"Enter salary: ";
              cin>>salary;
              arr[count-1]=new SalariedEmployee(name, salary);
              cout<<"Employee Details:\n";
              arr[count-1]->getDetail();
              break;
           }
           case 2: {
              count++;
              int hrs;
              cout<<"Enter name: ";
              cin.clear();
              cin.sync();
              getline(cin,name);
              cout<<"Enter Hourly salary: ";
              cin>>salary;
              cout<<"Enter Hours worked: ";
```

```
            cin>>hrs;
            arr[count-1]=new HourlyEmployee(name, salary,hrs);
            cout<<"Employee Details:\n";
            arr[count-1]->getDetail();
            break;
        }
        case 3: {
            count++;
            float com;
            cout<<"Enter name: ";
            cin.clear();
            cin.sync();
            getline(cin,name);
            cout<<"Enter sales made: ";
            cin>>salary;
            cout<<"Enter commission: ";
            cin>>com;
            arr[count-1]=new CommissionEmployee(name, salary,com);
            cout<<"Employee Details:\n";
            arr[count-1]->getDetail();
            break;
        }
        case 4: {
            count++;
            float com, base;
            cout<<"Ente name: ";
            cin.clear();
```

```
            cin.sync();
            getline(cin,name);
            cout<<"Enter sales made: ";
            cin>>salary;
            cout<<"Enter commission: ";
            cin>>com;
            cout<<"Enter base salary: ";
            cin>>base;
            arr[count-1]=new BasePlusCommissionEmployee(name, salary,base, com, 10.0);
            cout<<"Employee Details:\n";
            arr[count-1]->getDetail();
            break;
        }
        case 5:{
            float sum=0;
            for(int i=0;i<count;i++){
                arr[i]->getDetail();
                sum+=arr[i]->getSalary();
            }
            cout<<"Total spenditure: "<<sum<<endl;
            break;
        }
        default:{
            choice=0;
            break;
        }
    }
```

```
        if(choice==0) break;
    }


    return 0;
}
```

Output:

D:\sem-5\oop_Lab\11.exe

```
Enter:
 1. to create a Salaried Employee
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
1
Enter name: Tanmay Vig
Enter salary: 100000
Employee Details:

Employee Name: Tanmay Vig
Type: Salaried
Salary: 100000

Enter:
 1. to create a Salaried Employee
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
2
Enter name: Mudit Malhotra
Enter Hourly salary: 4000
Enter Hours worked: 44
Employee Details:

Employee Name: Mudit Malhotra
Type: Hourly
Salary: 184000

Enter:
 1. to create a Salaried Employee
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
3
Enter name: Mukul Monga
Enter sales made: 1000000
Enter commission: 20
Employee Details:

Employee Name: Mukul Monga
Type: Commissioned
Salary: 200000
```

D:\sem-5\oop_Lab\11.exe

```
Employee Name: Mukul Monga
Type: Commissioned
Salary: 200000

Enter:
 1. to create a Salaried Employee
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
4
Ente name: Manuj Monga
Enter sales made: 200000
Enter commission: 25
Enter base salary: 10000
Employee Details:

Employee Name: Manuj Monga
Type: Commissioned
Salary: 61000

Enter:
 1. to create a Salaried Employee
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
5

Employee Name: Tanmay Vig
Type: Salaried
Salary: 100000


Employee Name: Mudit Malhotra
Type: Hourly
Salary: 184000


Employee Name: Mukul Monga
Type: Commissioned
Salary: 200000


Employee Name: Manuj Monga
Type: Commissioned
```

D:\sem-5\oop_Lab\11.exe

```
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
5

Employee Name: Tanmay Vig
Type: Salaried
Salary: 100000


Employee Name: Mudit Malhotra
Type: Hourly
Salary: 184000


Employee Name: Mukul Monga
Type: Commissioned
Salary: 200000


Employee Name: Manuj Monga
Type: Commissioned
Salary: 61000

Total spenditure: 365000
Enter:
 1. to create a Salaried Employee
 2. to create a Hourly Employee
 3. to create a Comission Employee
 4. to create a Base Plus Commission Employee
 5. to get payroll of all employees
 any other number to exit
8

--------------------------------
Process exited after 99.35 seconds with return value 0
Press any key to continue . . .
```

P12: Templates-Write a program in C++ to implement a generic Stack.

Source code:

```
#include<iostream>

#include<string>

#define SIZE 5


using namespace std;


template <class T>

class Stack{

   private:

      int top, capacity;

      T* arr;

   public:

      Stack(int size=SIZE);

      void push(T k);

      T pop();

      T peek();

      int size();

      bool isEmpty();

      bool isFull();
```

```
        ~Stack(){
            delete[] arr;
        }
};
template <class T>
Stack<T>::Stack(int size){
   arr = new T[size];
   capacity = size;
   top = -1;
}

template <class T>
void Stack<T>::push(T k){
   if(isFull()){
      cout<<"Stack is Full!"<<endl;
   }else{
      cout<<"Pushing "<<k<<" to the stack\n";
      arr[++top]=k;
   }
}

template <class T>
T Stack<T>::pop(){
```

```
    T top_ele;
    if(isEmpty()){
        cout<<"Stack is Empty!"<<endl;
    }else{
        top_ele = arr[top--];
        cout<<"removing "<<top_ele<<" from Stack\n";
    }
    return top_ele;
}

template<class T>
T Stack<T>::peek(){
    return arr[top];
}

template<class T>
int Stack<T>::size(){
    return top+1;
}

template<class T>
bool Stack<T>::isEmpty(){
    return top==-1;
```

```
}


template<class T>
bool Stack<T>::isFull(){
   return top==capacity-1;
}


int main(){
   Stack<string> st;
   st.pop();
   st.push("Hey");
   st.push("two");
   cout<<st.pop()<<endl;
   cout<<"size of stack: "<<st.size()<<endl;
   st.push("three");
   st.push("four");
   st.push("five");
   cout<<"size of stack: "<<st.size()<<endl;
   st.push("six");
   if(st.isFull()){
      cout<<"Stack is full\n";
   }
   st.push("seven");
```

```
    st.pop();



    return 0;

}
```

Output:

```
Stack is Empty!
Pushing Hey to the stack
Pushing two to the stack
removing two from Stack
two
size of stack: 1
Pushing three to the stack
Pushing four to the stack
Pushing five to the stack
size of stack: 4
Pushing six to the stack
Stack is full
Stack is Full!
removing six from Stack

--------------------------------
Process exited after 0.05603 seconds with return value 0
Press any key to continue . . .
```

P13: Inheritance in Java: Implement a manager class that inherits from the Employee class in Java. Implement the relevant constructors also.

Source Code:

```
import java.util.*;


class Employee{
   int sNo;
   String name;
   long salary;

   public Employee(int sNo, String name, long salary) {
      this.sNo = sNo;
      this.name = name;
      this.salary = salary;
   }
}

class Manager extends Employee{
   String position;

   public Manager(int sNo, String name, long salary, String position){
```

```
        super(sNo,name,salary);

        this.position = position;

    }



    public void getData(){

        System.out.println("Emoplyee Details:\nSerial Number: "+ this.sNo+"\nName: "+
this.name+"\nPosition: "+ this.position + "\nSalary "+this.salary);

    }



}



public class inherit{

    public static void main(String[] args){



        Manager m = new Manager(100,"Tanmay Vig",1000000,"General Manager");

        m.getData();



    }

}
```

Output:

```
Emoplyee Details:
Serial Number: 100
Name: Tanmay Vig
Position: General Manager
Salary 1000000
PS D:\sem-5\oop_Lab>
```