

# Object Oriented Programming (LAB)(Code : 210247)

Semester III - Computer Engineering

(Savitribai Phule Pune University)

Strictly as per the New Credit System Syllabus (2019 Course)  
Savitribai Phule Pune University w.e.f. academic year 2020-2021

## Dr. Rachana Anil Satao

Ph.D. (Computer Engineering)  
Assistant Professor,  
Department of Computer Engineering,  
Smt. Kashibai Navale College of Engineering,  
Vadgaon Budruk, Pune-411041

## Araddhana Arvind Deshmukh

M.E. Computer Engineering (SPPU), Pune  
Ph.D. Research Scholar, Aarhus University, Denmark  
Asst. Professor, Department of Computer Engineering,  
Smt. Kashibai Navale College of Engineering,  
Vadgaon Budruk, Pune-411041

## Deepa Ashish Padalkar

M.E. Computer Engineering  
Assistant Professor,  
Department of Computer Engineering,  
Smt. Kashibai Navale College of Engineering,  
Vadgaon Budruk, Pune-411041

## Ansari M. A.

Ph.D. (Computer Engineering)\*  
Assistant Professor,  
Department of Computer Engineering,  
Smt. Kashibai Navale College of Engineering,  
Vadgaon Budruk, Pune-411041



## **Object Oriented Programming (LAB) (Code : 210247)**

**Dr. Rachana A. Satao, Araddhana A. Deshmukh, Deepa A. Padalkar, Ansari M. A.**

(Semester III – Computer Engineering, (Savitribai Phule Pune University))

Copyright © by Authors. All rights reserved. No part of this publication may be reproduced, copied, or stored in a retrieval system, distributed or transmitted in any form or by any means, including photocopy, recording, or other electronic or mechanical methods, without the prior written permission of the publisher.

This book is sold subject to the condition that it shall not, by the way of trade or otherwise, be lent, resold, hired out, or otherwise circulated without the publisher's prior written consent in any form of binding or cover other than which it is published and without a similar condition including this condition being imposed on the subsequent purchaser and without limiting the rights under copyright reserved above.

**First Printed in India :** January 2015 (Pune University)

**First Edition :** August 2020

This edition is for sale in India, Bangladesh, Bhutan, Maldives, Nepal, Pakistan, Sri Lanka and designated countries in South-East Asia. Sale and purchase of this book outside of these countries is unauthorized by the publisher.

**Printed at :** 37/2, Ashtavinayak Industrial Estate, Near Pari Company,  
Narhe, Pune, Maharashtra State, India.  
Pune – 411041

**ISBN :** 978-93-89889-53-6

**Published by**

**TechKnowledge Publications**

**Head Office :** B/5, First floor, Maniratna Complex, Taware Colony, Aranyeshwar Corner,

Pune - 411 009. Maharashtra State, India

Ph : 91-20-24221234, 91-20-24225678.

Email : [info@techknowledgebooks.com](mailto:info@techknowledgebooks.com),

Website : [www.techknowledgebooks.com](http://www.techknowledgebooks.com)

[210247] (FID : PO121) (Book Code : ONPO121A)

(Book Code : ONPO121A)

# Syllabus

## 210247 : Object Oriented Programming (LAB)

<b>Teaching Scheme</b> <b>Lectures : 04 Hours/Week</b>	<b>Credit Scheme</b> <b>02</b>	<b>Examination Scheme and Marks</b> <b>Term Work : 25 Marks</b> <b>Practical : 25 Marks</b>
---	-----------------------------------	---

**Companion Course :** 210243: Object Oriented Programming(OOP), 210244: Computer Graphics

### Course Objectives :

To understand basics of Computer Graphics, apply various methods and techniques for implementing line-circle drawing, projections, animation, shading, illumination and lighting using concepts of Object Oriented Programming.

**Course Outcomes :** On completion of the course, learner will be able to :

**CO1 :** Understand and apply the concepts like inheritance, polymorphism, exception handling and generic structures for implementing reusable programming codes.

**CO2 :** Analyze the concept of file and apply it while storing and retrieving the data from secondary storages.

**CO3 :** Analyze and apply computer graphics algorithms for line-circle drawing, scan conversion and filling with the help of object oriented programming concepts.

**CO4 :** Understand the concept of windowing and clipping and apply various algorithms to fill and clip polygons.

**CO5 :** Apply logic to implement, curves, fractals, animation and gaming programs.

### Guidelines for Instructor's Manual

The instructor's manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

### Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

### Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, punctuality.

### Guidelines for Practical Examination

Problem statements must be decided jointly by the internal examiner and external examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, transparent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of student's academics.

### Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. Use of open source software is encouraged. Based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.

Operating System recommended :- 64-bit Open source Linux or its derivative

Programming tools recommended: - Open Source C++ Programming tool like G++/GCC, OPENGL.

□□□

List of Assignments	
Group A	
1.	Implement a class Complex which represents the Complex Number data type. Implement the following 1. Constructor (including a default constructor which creates the complex number 0+0i). 2. Overload operator+ to add two complex numbers. 3. Overload operator* to multiply two complex numbers. 4. Overload operators << and >> to print and read Complex Numbers
2.	Develop a program in C++ to create a database of student's information system containing the following information: Name, Roll number, Class, Division, Date of Birth, Blood group, Contact address, Telephone number, Driving license no. and other. Construct the database with suitable member functions. Make use of constructor, default constructor, copy constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete as well as exception handling.
3.	Imagine a publishing company which does marketing for book and audio cassette versions. Create a class publication that stores the title (a string) and price (type float) of publications. From this class derive two classes: book which adds a page count (type int) and tape which adds a playing time in minutes (type float).  Write a program that instantiates the book and tape class, allows user to enter data and displays the data members. If an exception is caught, replace all the data member values with zero values.
Group B	
4.	Write a C++ program that creates an output file, writes information to it, closes the file, open it again as an input file and read the information from the file.
5.	Write a function template for selection sort that inputs, sorts and outputs an integer array and a float array.
Group C	
6.	Write C++ program using STL for sorting and searching user defined records such as personal records (Name, DOB, Telephone number etc) using vector container. <b>OR</b> Write C++ program using STL for sorting and searching user defined records such as Item records (Item code, name, cost, quantity etc) using vector container.
7.	Write a program in C++ to use map associative container. The keys will be the names of states and the values will be the populations of the states. When the program runs, the user is prompted to type the name of a state. The program then looks in the map, using the state name as an index and returns the population of the state.

□□□



**INDEX**

**Program L1 ..... L-1**

Implement a class Complex which represents the Complex Number data type. Implement the following 1. Constructor (including a default constructor which creates the complex number 0+0i). 2. Overload operator+ to add two complex numbers. 3. Overload operator\* to multiply two complex numbers. 4. Overload operators << and >> to print and read Complex Numbers.

**Program L2 ..... L-6**

Develop a program in C++ to create a database of student's information system containing the following information: Name, Roll number, Class, Division, Date of Birth, Blood group, Contact address, Telephone number, Driving license no. and other. Construct the database with suitable member functions. Make use of constructor, default constructor, copy constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete as well as exception handling.

**Program L3 ..... L-9**

Imagine a publishing company which does marketing for book and audio cassette versions. Create a class publication that stores the title (a string) and price (type float) of publications. From this class derive two classes: book which adds a page count (type int) and tape which adds a playing time in minutes (type float). Write a program that instantiates the book and tape class, allows user to enter data and displays the data members. If an exception is caught, replace all the data member values with zero values.

**Program L4 ..... L-12**

Write a C++ program that creates an output file, writes information to it, closes the file, open it again as an input file and read the information from the file.

**Program L5 ..... L-14**

Write a function template for selection sort that inputs, sorts and outputs an integer array and a float array.

**Program L6 ..... L-15**

Write C++ program using STL for sorting and searching user defined records such as personal records (Name, DOB, Telephone number etc) using vector container.

OR

Write C++ program using STL for sorting and searching user defined records such as Item records (Item code, name, cost, quantity etc) using vector container.

**Program L7 ..... L-21**

Write a program in C++ to use map associative container. The keys will be the names of states and the values will be the populations of the states. When the program runs, the user is prompted to type the name of a state. The program then looks in the map, using the state name as an index and returns the population of the state.



# Lab Manual

## Part I : Object Oriented Programming

### Group A

#### Program L1

Implement a class Complex which represents the Complex Number data type. Implement the following  
1. Constructor (including a default constructor which creates the complex number  $0+0i$ ). 2. Overload operator+ to add two complex numbers. 3. Overload operator\* to multiply two complex numbers. 4. Overload operators << and >> to print and read Complex Numbers.

#### Solution :

```
#include<iostream>
#include<stdlib.h>
using namespace std;
class complex_number
{
    float r;
    float i;
public:
    complex_number()
    {
        r=0.0;
        i=0.0;
    }
    complex_number(float temp1, float temp2)
    {
        r=temp1;
        i=temp2;
    }
    //display number
    friend void operator<<(ostream &a, complex_number &b)
    {
        a<<"\n"<<b.r<<" " + "<<b.i<<" i";
    }
    friend istream &operator>>( istream &input, complex_number &num )
    {
```



```
    input >> num.r >> num.i;
    return input;
}

complex_number operator+(complex_number);
complex_number operator*(complex_number);
};

complex_number complex_number::operator +(complex_number A)
{
    complex_number B;
    B.r=r+A.r;
    B.i=i+A.i;
    return B;
}

complex_number complex_number::operator *(complex_number A)
{
    complex_number B;
    B.r=r-A.r;
    B.i=i-A.i;
    B.r=((r)*(A.r))-((i)*(A.i));
    B.i=((r)*(A.i))+((A.r)*(i));
    return B;
}

int main()
{
    int ch;
    complex_number A;
    complex_number B;
    complex_number C;
    cout<<"\n Operations on complex numbers";
    while(1)
    {
        cout<<"\n Menu";
        cout<<"\n1. Create constructor";
        cout<<"\n2. Accept numbers";
        cout<<"\n3. Display numbers";
        cout<<"\n4. Add numbers";
        cout<<"\n5. Multiply numbers";
```



```
cout<<"\n6. Exit";
cout<<"\n Enter your choice: ";
cin>>ch;
switch(ch)
{
    case 1:
        cout<<"\n Complex number after initialization ";
        cout<<A;
        cout<<B;
        break;
    case 2:
        cout<<"\n Enter first complex number (First real and then imaginary part): ";
        cin>>A;
        cout<<"\n Enter second complex number (First real and then imaginary part): ";
        cin>>B;
        break;
    case 3:
        cout<<"\n First complex number";
        cout<<A;
        cout<<"\n Second complex number";
        cout<<B;
        break;
    case 4:
        C=A+B;
        cout<<"\n Addition of two complex numbers is";
        cout<<C;
        break;
    case 5:
        C=A*B;
        cout<<"\n Multiplication of two complex numbers is";
        cout<<C;
        break;
    case 6:
        exit(0);
}
}
return 0;
}
```



**Output :**

Operations on complex numbers

Menu

1. Create constructor
2. Accept numbers
3. Display numbers
4. Add numbers
5. Multiply numbers
6. Exit

Enter your choice: 1

Complex number after initialization

$0 + 0i$

$0 + 0i$

Menu

1. Create constructor
2. Accept numbers
3. Display numbers
4. Add numbers
5. Multiply numbers
6. Exit

Enter your choice: 2

Enter first complex number (First real and then imaginary part): 5 10.2

Enter second complex number (First real and then imaginary part): 15 20.4

Menu

1. Create constructor
2. Accept numbers
3. Display numbers
4. Add numbers
5. Multiply numbers
6. Exit

Enter your choice: 3

First complex number

$5 + 10.2i$

Second complex number



$15 + 20.4 i$

Menu

1. Create constructor
2. Accept numbers
3. Display numbers
4. Add numbers
5. Multiply numbers
6. Exit

Enter your choice: 4

Addition of two complex numbers is

$20 + 30.6 i$

Menu

1. Create constructor
2. Accept numbers
3. Display numbers
4. Add numbers
5. Multiply numbers
6. Exit

Enter your choice: 5

Multiplication of two complex numbers is

$-133.08 + 255 i$

Menu

1. Create constructor
2. Accept numbers
3. Display numbers
4. Add numbers
5. Multiply numbers
6. Exit



## Program L2

Develop a program in C++ to create a database of student's information system containing the following information: Name, Roll number, Class, Division, Date of Birth, Blood group, Contact address, Telephone number, Driving license no. and other. Construct the database with suitable member functions. Make use of constructor, default constructor, copy constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete as well as exception handling.

### Solution :

```
#include <iostream>
#include <string>
#include <cstring>
#include <stdlib.h>
using namespace std;
class student
{
    int roll;
    char name[30];
    float marks;
public:
    student()
    {
        roll=0;
        marks=0;
        strcpy(name, "");
    }
    student(int roll, char name[30], float marks)
    {
        this->roll=roll;
        strcpy(this->name, name);
        this->marks=marks;
    }
    void accept()
    {
        cout<<"\n Enter name: ";
        cin>>name;
        cout<<"\n Enter Roll Number: ";
        cin>>roll;
        cout<<"\n Enter marks: ";
        cin>>marks;
    }
}
```



```
void display()
{
    cout<<"\n\n Name:  "<<name;
    cout<<"\n Roll Number: "<<roll;
    cout<<"\n Marks:  "<<marks;
}

};

int main()
{
    int i,ch, n;
    student s[10] ;
    while(1)
    {
        cout<<"\n1.Create\n2.Display\n3. Exit";
        cout<<"\n Enter your choice: ";
        cin>>ch;
        switch(ch)
        {
            case 1:
                cout<<"\n Enter how many student information you want to store: ";
                cin>>n;
                for(i=0;i<n;i++)
                {
                    cout<<"\n Enter information of "<<i+1 <<"th student: ";
                    s[i].accept();
                }
                break;
            case 2:
                cout<<"\n Student database is as follows";
                for(i=0;i<n;i++)
                    s[i].display();
                break;
            case 3:
                exit(0);
        }
    }
    delete [] s;
    return 0;
}
```



**Output :**

```
1.Create
2.Display
3. Exit
Enter your choice: 1
Enter how many student information you want to store: 3
Enter information of 1th student:
Enter name: dmjadhav
Enter Roll Number:10
Enter marks:66.66
Enter information of 2th student:
Enter name: maansari
Enter Roll Number:11
Enter marks:66.66
Enter information of 3th student:
Enter name: xyz
Enter Roll Number:12
Enter marks: 66
1.Create
2.Display
3.Exit
Enter your choice:2
Student database is as follows
Name: dmjadhav
Roll Number:10
Marks:66.66
Name:maansari
Roll Number: 11
Marks: 66.66
Name: xyz
Roll Number: 12
Marks: 66
```

**Program L3**

Imagine a publishing company which does marketing for book and audio cassette versions. Create a class publication that stores the title (a string) and price (type float) of publications. From this class derive two classes: book which adds a page count (type int) and tape which adds a playing time in minutes (type float). Write a program that instantiates the book and tape class, allows user to enter data and displays the data members. If an exception is caught, replace all the data member values with zero values.

**Solution :**

```
#include <iostream>
#include <stdexcept>
#include <typeinfo>
#include <string>
#include <cstring>
#include <stdlib.h>
using namespace std;
class publish
{
    public:
    char title[20];
    float price;
    void getdata()
    {
        cout << "\n Enter title and price : ";
        cin >> title >> price;
    }
    void putdata()
    {
        cout << "\n Information of publication: ";
        cout << "\n Title: " << title << "\t Price:" << price;
    }
};
class book: public publish
{
    int page_count;
    public:
    void getdata()
    {
        cout << "\n Enter page count : ";
        cin >> page_count;
    }
}
```



```
void putdata(publish p)
{
    cout<<"\n Information of book: ";
    cout<<"\nTitle: "<<p.title<<"\t Price:"<<p.price<<"\t Page count"<<page_count;
}
};
class CD: public publish
{
    float play_time;
public:
    void getdata()
    {
        cout<<"\n Enter play time : ";
        cin>>play_time;
    }
    void putdata(publish p)
    {
        cout<<"\n Information of CD: ";
        cout<<"\n Title: "<<p.title<<"\t Price:"<<p.price<<"\t Page count"<<play_time;
    }
};
int main()
{
    int ch;
    publish p;
    book b;
    CD c;
    while(1)
    {
        cout<<"\n1. Enter publication information";
        cout<<"\n2. Enter book info";
        cout<<"\n3. Enter CD info";
        cout<<"\n4. Exit";
        cout<<"\n Enter your choice";
        cin>>ch;
        switch(ch)
        {
            case 1:
```



```
p.getdata();
p.putdata();
break;
case 2:
    b.getdata();
    b.putdata(p);
    break;
case 3:
    c.getdata();
    c.putdata(p);
    break;
case 4:
    exit(0);
}
}
return 0;
}
```

**Output :**

1. Enter publication information

2. Enter book info

3. Enter CD info

4. Exit

Enter your choice1

Enter title and price : techmax 34543

Information of publication:

Title: techmax Price:34543

1. Enter publication information

2. Enter book info

3. Enter CD info

4. Exit

Enter your choice2

Enter page count : 345

Information of book:

Title: techmax Price:34543 Page count345

1. Enter publication information

2. Enter book info





3. Enter CD info

4. Exit

Enter your choice3

Enter play time : 56.5

Information of CD:

Title: techmax Price:34543 Page count56.5

### Group B

#### Program L4

Write a C++ program that creates an output file, writes information to it, closes the file, open it again as an input file and read the information from the file.

#### Solution :

```
#include<iostream>
#include<fstream>
using namespace std;
class file
{
    char name[20];
    int emp_id;
    float salary;
public:
    void accept()
    {
        cin>>name;
        cin>>emp_id;
        cin>>salary;
    }
    void display()
    {
        cout<<"\n"<<name<<"\t"<<emp_id<<"\t"<<salary;
    }
};
int main()
{
```



```
file o[5];
fstream f;
int i,n;
f.open("input"); //create file
cout<<"\n How many employee information wanted to store:";
cin>>n;
cout<<"\n Enter information of 3 employees (Enter name, emp_id, salary)";
for(i=0;i<n;i++)
{
    cout<<"\n Enter information of "<<i<<" employee ";
    o[i].accept(); //accept input from user
    f.write((char* ) &o[i], sizeof(o[i])); //write object in file
}
f.close();
f.open("input",ios::in);
cout<<"\n Information of employee is as follows";
for(i=0;i<n;i++)
{
    f.read((char* ) &o[i], sizeof(o[i])); //read data from file
    o[i].display(); //display data
}
f.close();
return 0;
}
```

**Output :**

```
How many employee information wanted to store:3
Enter information of 3 employees (Enter name, emp_id, salary)
Enter information of 0 employee xyz      10    70000
Enter information of 1 employee zbc      11    75000
Enter information of 2 employee mnq 12    80000
Information of employee is as follows
xyz  10    70000
zbc  11    75000
mnq  12    80000
```

**Program L5**

Write a function template for selection sort that inputs, sorts and outputs an integer array and a float array.

**Solution :**

```
#include<iostream>
using namespace std;
template<typename T>
void sort(T a[], int n)
{
    int pos_min,temp, i;

    for (int i=0; i < n-1; i++)
    {
        pos_min = i;
        for (int j=i+1; j < n; j++)
        {
            if (a[j] < a[pos_min])
                pos_min=j;
        }
        if (pos_min != i)
        {
            temp = a[i];
            a[i] = a[pos_min];
            a[pos_min] = temp;
        }
    }

    //display sorted elements
    cout<<"\n Sorted elements are :\n";
    for(i=0;i<n; i++)
        cout<<"\t"<<a[i];
}

int main()
{
    int n,i;
    int a[10];
    float b[10];

    cout<<"\n Selection sort using function template";
    cout<<"\n Sorting int numbers";
    cout<<"\n Enter how many number wanted to sort";
```



```
    cin >> n;
    for(i=0; i<n; i++)
        cin >> a[i];
    sort<int>(a, n);
    cout << "\n Sorting floating point numbers";
    cout << "\n Enter how many number wanted to sort";
    cin >> n;
    for(i=0; i<n; i++)
        cin >> b[i];
    sort<float>(b, n);
    return 0;
}
```

**Output :**

```
Selection sort using function template
Sorting int numbers
Enter how many number wanted to sort5
5 4 3 2 1
Sorted elements are :
    1    2    3    4    5
Sorting floating point numbers
Enter how many number wanted to sort5
5.5 6.6 456.44 2.2 1.2
Sorted elements are :
1.2    2.2    5    6    456
```

**Group C****Program L6**

Write C++ program using STL for sorting and searching user defined records such as personal records (Name, DOB, Telephone number etc) using vector container.

**OR**

Write C++ program using STL for sorting and searching user defined records such as Item records (Item code, name, cost, quantity etc) using vector container.

**Solution :**

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



```
using namespace std;

// class student declaration
class student
{
public:
    int rollno;
    string name;
    char mob[20];
    bool operator==(const student &student1)
    {
        return(rollno==student1.rollno);
    }
    bool operator<(const student &student1)
    {
        return(rollno<student1.rollno);
    }

    friend ostream& operator << (ostream &out, const student &);
    friend istream& operator >> (istream &in, student &k);
}; // class student ends

// overloading operator <<
ostream & operator << (ostream &out, const student &k)
{
    out<<"\n\t\t"<<k.rollno<<"\t\t"<<k.name<<"\t\t"<<k.mob;
    return out;
}

// overloading operator >>
istream & operator >> (istream &in , student &k)
{
    cout<<"\nEnter Roll No : ";
    in>>k.rollno;
    cout<<"\nEnter Name : ";
    in>>k.name;
    cout<<"\nEnter mob : ";
    in>>k.mob;
```



```
    return in;
}

bool sort_func(const student &x, const student &y)
{
    return(x.rollno<y.rollno);
}

// function to accept record
vector<student> read()
{
    int n;
    student k;
    vector<student> j;
    cout<< "\nEnter total no. of students : ";
    cin>>n;
    for(int i=0;i<n;i++)
    {
        cin>>k;
        j.push_back(k);
    }
    return j;
}

void printfunction(const student &k)
{
    cout<<k;
}

// Function For Display Record
void print( const vector<student> &j)
{
    cout<< "\n\t\tROLL NO\t\tNAME\t\tMobile Number";
    for_each(j.begin(),j.end(),printfunction);

}

// function to search record
void search( vector<student>&j )
{
```



```
        student k;
        cout<<"\nEnter Student Roll No To Search : ";
        cin>>k.rollno;
        cout<<"\n\n\t\tROLL NO\t\tNAME\t\tDATE OF BIRTH";
        vector<student>::iterator p;
        p=find(j.begin(),j.end(),k);
        if(p!=j.end())
            cout<<*p;
        else
            cout<<"\nNot found ";
    }

// Function to Sort Record
void sort( vector<student> &j)
{
    sort(j.begin(),j.end(),sort_func);
}

// Main Function
int main()
{
    vector<student> s;
    int ch;
    do
    {
        cout<<"\n\n\t\t* * * * Personal Record Database * * * *";
        cout<<"\n\t\t1.Create Record";
        cout<<"\n\t\t2.Display Record ";
        cout<<"\n\t\t3.Search Record";
        cout<<"\n\t\t4.Sort Record";
        cout<<"\n\t\t5.Quit Menu";
        cout<<"\n\t\t-----";
        cout<<"\n\t\tEnter your choice : ";
        cin >> ch;
        switch(ch)
        {
            case 1:
                s=read();
```



```
        break;
    case 2:
        print(s);
        break;
    case 3:
        search(s);
        break;
    case 4:
        sort(s);
        print(s);
        break;
    }
}while(ch!=5);
} // end of the main function
```

**Output :**

```
***** Personal Record Database *****
```

- 1.Create Record
- 2.Display Record
- 3.Search Record
- 4.Sort Record
- 5.Quit Menu

```
-----
Enter your choice : 1
```

```
Enter total no. of students : 4
```

```
Enter Roll No : 103
```

```
Enter Name : rachana
```

```
Enter mob : 1223122211
```

```
Enter Roll No : 101
```

```
Enter Name : seeta
```

```
Enter mob : 5432123427
```





Enter Roll No : 100

Enter Name : geeta

Enter mob : 7678456432

Enter Roll No : 102

Enter Name : neeta

Enter mob : 2343123211

\*\*\*\*\* Personal Record Database \*\*\*\*\*

- 1.Create Record
- 2.Display Record
- 3.Search Record
- 4.Sort Record
- 5.Quit Menu

-----  
Enter your choice : 2

ROLL NO	NAME	Mobile Number
103	rachana	1223122211
101	seeta	5432123427
100	geeta	7678456432
102	neeta	2343123211

\*\*\*\*\* Personal Record Database \*\*\*\*\*

- 1.Create Record
- 2.Display Record
- 3.Search Record
- 4.Sort Record
- 5.Quit Menu

-----  
Enter your choice : 3



Enter Student Roll No To Search : 101

ROLL NO	NAME	DATE OF BIRTH
101	seeta	5432123427

\*\*\*\*\* Personal Record Database \*\*\*\*\*

- 1.Create Record
- 2.Display Record
- 3.Search Record
- 4.Sort Record
- 5.Quit Menu

-----  
Enter your choice : 4

ROLL NO	NAME	Mobile Number
100	geeta	7678456432
101	seeta	5432123427
102	neeta	2343123211
103	rachana	1223122211

\*\*\*\*\* Personal Record Database \*\*\*\*\*

- 1.Create Record
- 2.Display Record
- 3.Search Record
- 4.Sort Record
- 5.Quit Menu

-----  
Enter your choice : 5

\*/

### Program L7

Write a program in C++ to use map associative container. The keys will be the names of states and the values will be the populations of the states. When the program runs, the user is prompted to type the name of a state. The program then looks in the map, using the state name as an index and returns the population of the state.

#### Solution :

```
#include <iostream>
#include <map>
#include <string>
#include <utility>
```



```
using namespace std;

int main()
{
    // initialize container
    typedef map<string, int> mapType;
    mapType populationMap;

    // insert elements in map
    populationMap.insert(pair<string, int>("Maharashtra", 123144223));
    populationMap.insert(pair<string, int>("Rajasthan", 81032689));
    populationMap.insert(pair<string, int>("Gujarat", 63872399));
    populationMap.insert(pair<string, int>("Karnataka", 67562686));
    populationMap.insert(pair<string, int>("Tamilnadu", 77841267));

    mapType::iterator iter;

    // display the size of the map
    cout << " * * * * Population of states in India * * * * \n";
    cout << "\nSize of populationMap: " << populationMap.size() << '\n';

    // find function will return an iterator to the matching element if it is
    // found or to the end of the map if the key is not found

    string state_name;
    cout << "\nEnter name of the state : ";
    cin >> state_name;

    iter = populationMap.find(state_name);
    if( iter != populationMap.end() )
        cout << state_name << "s populations is " << iter->second ;
    else
        cout << "Key is not in populationMap" << '\n';

    // clear the entries in the map using clear
    populationMap.clear();
}
```

**Output :**

```
* * * * Population of states in India * * * *
```

```
Size of population Map: 5
```

```
Enter name of the state : Rajasthan
```

```
Rajasthan's populations is 81032689
```

```
* * * * Population of states in India * * * *
```

```
Size of populationMap: 5
```

```
Enter name of the state : Maharashtra
```

```
Maharashtra's populations is 123144223
```

□□□