Laboratory Manual

For

OBJECT ORIENTED PROGRAMMING

(IT-303)

B.Tech (IT)

SEM III

June 2010

Faculty of Technology
Dharmsinh Desai University
Nadiad.
www.ddu.ac.in

LIST OF EXPERIMENTS

A) Student Exam Result Program
B) Scan mixed character string with space and count capital letter, lower case letters and Digits.
C) i) Exchange two variable values by function.
   ii) Sum of Structure variable and returning structure variable.
D) i) Employees Total Salary calculation.
   ii) Room area calculation
E) Define class with constructer and destructor which will count no. of object created and destroyed.
F) i) Plus operator overloading.
   ii) Minus operator overloading
   iii) Study of friend function
G) Study of Inheritance
H) Implement string as new data type
I) Stack with Virtual function
J) i) Write a program that writes 10 records and display all the records.
   ii) Store 10 integer nos. in file, in character and binary form.

LABWORK BEYOND CURRICULA

1. Write a program that finds the index of the element of the array provided by user.
2. Implement Selection Sort algorithm.

Sample Experiment

1 AIM: Telephone Directory with Basic Inserting and Searching facility With Use of file.
2 TOOLS/APPARATUS: Turbo *C*
3 STANDARD PROCEDURES:

COMMON PROCEDURE:

Step 1: Create a folder in either E or F drive with your Id Number or Name Followed by RollNo.

Step 2: Start the TC (Turbo C) from Desktop Icon or Go To Directory D:/TC/BIN/ and run tc.exe . An Editor will be start.

Step 3: Click on File Menu --> New. New (.cpp) file will be created. Again Click on File -> Save an dialog box is going open write the path to your directory e.g. E:\ADD_20\FileName.CPP and Press OK. Now your CPP program is going to save at your directory.

Step 4: Go To Option->Directories Check That Include Directory is Set As D:\TC\Include and Library Directory is Set To D:\TC\LIB

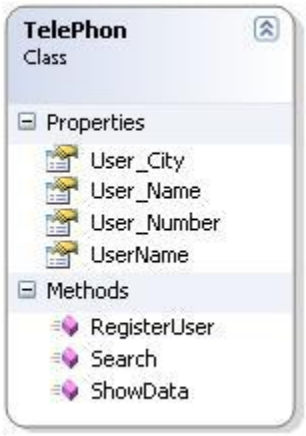
3.1 Analyzing the Problem:

- This Example is regarding to the Telephone directory with the Basic functionality Like Adding User and Searching the User by phone number or the UserName.
- From the noun Phrase analysis we come to know that we have to create Class Name "Telephone Directory" and Text file named "UserInfo" that stores the data base(User Info). Because we need to Map Real Object "Telephone Directory" to the Programming Language by creating the call called "Telephone_Dir" and its Methods.
- Class "Telephone Directory" has the data member like User Name, hone Number And City.
- We need to create the functionality like Register User and Search User by Name.

3.2 Designing the Solution:

- Here we need to create Telephone_Dir class having two char arry User_Name, User_city and long int User_Number to store user details.
- A file UserInfo.Txt to store all the details of register user.
- A class having three method RegisterUser for registering user to data base, Search for Searching user from database and ShowData to display the Searched User Details.

Class Diagram



3.3 Implementing the Solution

3.3.1 Writing Source Code

Teleph.Cpp

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
#include<string.h>
#include<iomanip.h>

class TelePhone_Dir
{
private:
 char User_Name[20];
char User_City[20];
char UserName[20];
long int User_Number;
public:

  void RegisterUser()
  {
    cout<<"   Enter The Details Of User " <<endl;
    cout<<"Enter the UserName     :   "<<endl;
    cin>>User_Name;
    cout<<"Enter the City Name    :  "<<endl;
    cin>>User_City;
    cout<<"Enter the Phone Number : "<<endl;
    cin>>User_Number;
  }
  void ShowData()
   {
```

```
    cout<<"All The Details Are "<<endl;
    cout<<setw(20)<<"User_Name"<<setw(20)<<"User_City"<<setw(20)<<"User_Number"<<
    endl;
    cout<<setw(20)<<User_Name<<setw(20)<<User_City<<setw(20)<<User_Number<<endl;
   }
 static void Search(char UserName[])
  {
    TelePhone_Dir T;
    int flag =0;
    fstream file;
    file.open("UserInfo.TEXT",ios::app|ios::out|ios::in);
    file.seekg(0);
    file.read((char*)&T,sizeof(T));
     while(!file.eof())
     {
            if(strcmp(T.User_Name,UserName)>-1)
              {
               T.ShowData();
                    break;
                  }
                  else
                  {
                    file.read((char*)&T,sizeof(T));
                  }
            }
       if(flag==0)
       {
        cout<<"User Is Not Register"<<endl;
        }
    }

};

 void main()
  {
  TelePhone_Dir t1;
  char choice_Reg;
  char UserName[20];
  fstream file;
  clrscr();
  file.open("UserInfo.TEXT",ios::app|ios::out|ios::in);
 do
 {
   clrscr();
   cout<<"   Enter The Details  "<<endl;
   cout<<"   1 .For UserRegistration  "<<endl;
```

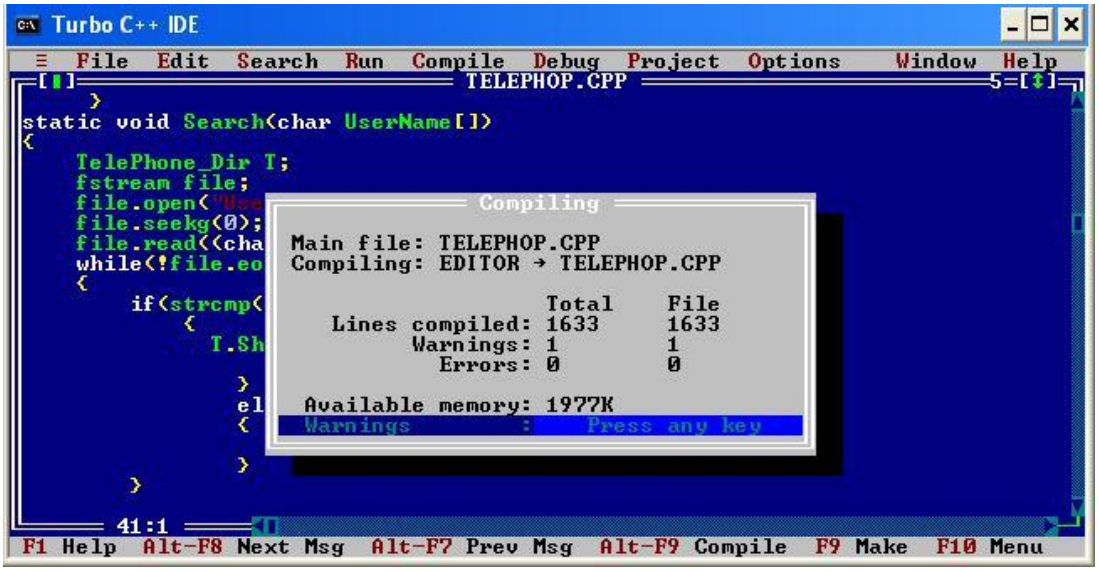
```
cout<<"   2 .For Search User By UserName "<<endl;
cout<<"   3 .For Exit from System  "<<endl;
cin>>choice;
        if(choice==1)
        {
         do
           {
                   clrscr();
                    t1.RegisterUser();
                    file.write((char*)&t1,sizeof(t1));
                    cout<<"\n Do you Want To Enter another User (y/n)? "<<endl;
                     cin>>choice_Reg;
           }while(choice_Reg=='y');
  }
  else if(choice==2)
        {
         do
           {
              clrscr();
              cout<<"Enter The UseName "<<endl;
              cin>>UserName;
              TelePhone_Dir::Search(UserName);
              cout<<"Do You want another (y/n)?"<<endl;
                   cin>>choice_Reg;
           }while(choice_Reg=='y');
        }
  else
        {
        break;
        }
        cout<<" Do You Want To Contine ....(y/n)?"<<endl;
        cin>>choice_Reg;
        if(choice_Reg == 'y')
           {
            clrscr();
            continue;
     }
    }while(choice<3);
  }
```

3.3.2 Compilation /Running and Debugging the Solution

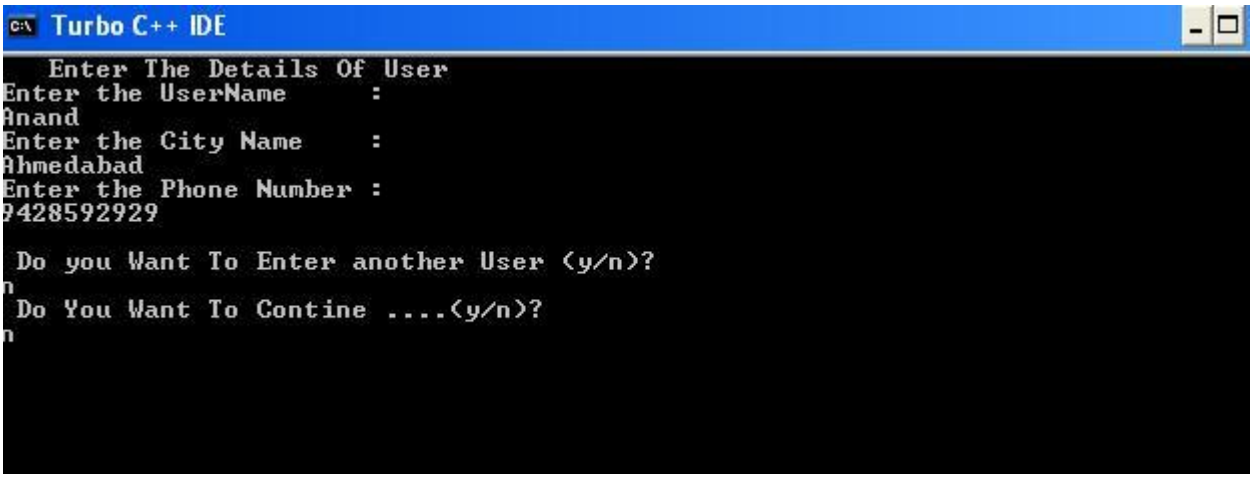
- Go to Compile Menu And Press the Compile For the Compilation of the code.
- If Successful Compilation is done then Run The Code Using ctrl + F9 key.

3.4 Testing the Solution

Screen shot for Compilation

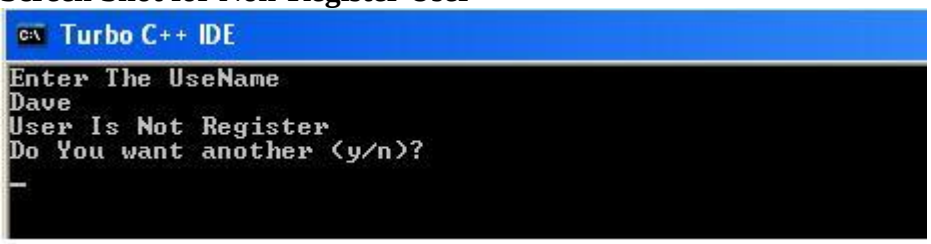


Screen shot for Registering a User



Screen shot for Searching User

Screen Shot for Non-Register User



4 Conclusions

- This Sample Experiment Demonstrates All Basic Steps required to create the Simple OOP Program.
- Starting from "noun phrase analysis", Designing the solution based on problem definition and "noun phrase analysis".
- Steps for compiling and running the code.
- This Experiment also demonstrates way how we can map real time objects to Programming Language and create an application that transforms Human efforts to the Automated System.

EXPERIMENT - 1

Aim: Student Exam Result Program (Using Structures)

Tools / Apparatus: Turbo C

Procedure:

- Define student structure with student name, city, branch, and 5 subject's marks
- In Main function accepts 4 student information with cin >> function.
- Display student name, city, branch, and percentage marks with heading.
- Use setw () function for formatted output and cout << function.
- We should include<iomanip.h> file to use setw() function.
- Definition of setw( ) is setw(integer number).
- A positive integer define right Justify the Text in defined number of Fields, while negative integer define left justify the Text in defines number of fields.
- USE: cout<<setw(5)<<"Name", this will right justify the Name in 5 fields.

EXPERIMENT - 2

Aim: Scan mixed character string with space and count capital letter, lower case letters and Digits. (Use of Loops and Decision statements)

Tools / Apparatus: Turbo C

Procedure:

- Define String of 50 char length.
- Accept only one by one char of string using do … while loop and getche () function.
- After Acceptance of each character, find type of it and increment corresponding counter.
- We should include<conio.h> file to use getche () function.
- Repeat this program with use of switch statement instead of if else structure.
- Use the ASCII value of the char to identify the char is in upper case , lower case, Numbers .

EXPERIMENT - 3

i) **Aim**: Exchange two variable values by function

Tools / Apparatus: Turbo C

Procedure:

- Exchange two variable value passing, pass by reference method.
- Create a separate function called Swap having null return type and accepting two integer as arguments.
- Create a separate function called Swap having null return type and accepting two integer reference as arguments. void swap (int&,int&).
- Exchange two variable values without use of the temporary variable.
- a = a+b; b=a-b; a=a-b;

ii) **Aim**: Sum of Structure variable and returning structure variable.

Tools / Apparatus: Turbo C

Procedure:

- Define Distance structure consists of feet as integer type and inch as float values.
- Define function with two distance object as argument , which returns sum of the both object.
- Formula:

```
Distance temp;
temp.feet = Obj1.feet + Obj2.feet;
temp.inch = Obj1.inch + Obj2.inch;
if(temp.inch >=12)
{
  temp.feet++;
  temp.inch-=12;}
```

EXPERIMENT - 4

i) **Aim**: Employees Total Salary calculation.

Tools / Apparatus: Turbo C

Procedure:

- Define a class Employee with data employee name, city, basic salary, dearness allowance (DA) and house rent (HRA).
- Define getdata (), calculate (), and display () functions.
- Calculate function should find the total salary and display function should display it.
- Total = basic+basic*da/100+basic*hra/100;

ii) **Aim**: Room area calculation with use class and structure

Tools / Apparatus: Turbo C

Procedure:

- Define structure distance with feet and inches.
- Define a class Room with length and width data of distance type.
- Class should contain getdata() function for getting the values from User, calculate() which will calculate the area of the room and Display it.
- Formula:

  area of room = length * width;
  length.feet+=length.inch/12;
  width.feet += width.inch/12;
  area of room = lengt.feet * width.feet;

EXPERIMENT - 5

Aim: Define class with constructer and destructor which display following output:-

Line 1: Object 1 is created.
Line 2: Display function call in main.
Line 3: Object 2 is created.
Line 4: Object 1 is destroyed.
Line 5: Object 2 is destroyed

Tools / Apparatus: Turbo C

Procedure:

- We have define 3 count variable 1st is line counter, 2nd is Object counter and 3rd is destroy object counter.
- All this count variables are static variables.
- Initialize then outside class datatype classname :: variable name = 0;
- Increment the object counter in constructor , increment destroy object counter in destructor , increment line counter in constructor and destructor.

EXPERIMENT - 6

i)Aim: Plus operator overloading.
Tools / Apparatus: Turbo C
Procedure:

- Define time class with function which overload + operator to add two time objects.
- Having data Member int hours,mins and seconds.
- Time time :: operator +(time 12)

ii) Aim: Minus operator overloading

Tools / Apparatus: Turbo C

Procedure:

- Define distance class with function which overload – operator to subtract one distance object from another.
- Having data member int feet and float inches. Subtraction should subtract feet from feet and inch from inch values.

iii) Aim: Study of friend function

Tools / Apparatus: Turbo C

Procedure:

- Overload + operator as a friend function for distance class so it allows d3 = d2 + 10.
- Distance class has the data member, int feet and float inch.
- Friend distance operator + (distance, distance) function prototype

EXPERIMENT - 7

i) Aim: Study of inheritance

Tools / Apparatus: Turbo C

Procedure:

- Define one student class contains university and degree data.
- Define one employee class contains employee name and salary data.
- Now derive manager class from above class which contains all 4 details. Inheritance type should be private.
- All the classes have getdata() and showdata() function to display the results.

EXPERIMENT - 8

Aim: Implement string as new data type

Tools / Apparatus: Turbo C

Procedure:

- Create string class with copy initialization and assignment functionality.
- Overload constructor for 3 types initialization null, single char and multiple char string.

EXPERIMENT - 9

Aim: Stack with Virtual function

Tools / Apparatus: Turbo C

Procedure:

- Implement stack with push() and pop() functions for inserting and retrieving the value from stack.
- Create a stack class have data member as integer array , integer variable to point current location.
- Implement enhanced stack with underflow and overflow condition checking push and pop function.

EXPERIMENT - 10

i) Aim: Write a program that writes 10 records and display all the records.

Tools / Apparatus: Turbo C

Procedure:

- Create class test with getdata and showdata functions.
- In main define array of 10 objects.
- Create or open file in append mode with open function using ios flags.
- Get each object data and put it in file by write function.

ii) Aim: Store 10 integer nos. in file, in character and binary form.
Tools / Apparatus: Turbo C
Procedure: Same as Above.

EXPERIMENT - 11

Aim: Write a program that finds the index of the element of the array provided by user(Binary Search).

Tools / Apparatus: Turbo C

Procedure:

- Create the array of integer elements.
- Check whether the elements are sorted or not, If not then sort the elements in ascending order.
- With the help of recursion find the index of given element.

EXPERIMENT -12

Aim: Implement Selection Sort Algorithm.
Tools / Apparatus: Turbo C
Procedure:

- Create the array of integer elements.
- Sort the array of the element with this algorithm.

```
for i = 1:n,
   k = i
   for j = i+1:n, if a[j] < a[k], k = j
   → invariant: a[k] smallest of a[i..n]
   swap a[i,k]
   → invariant: a[1..i] in final position
end
```

.

References

Reference Books:

1. Mastering C++ by K.R.venugopal & rajkumar

2. Let us C++ by Yashwant Kanitkar