Ex. No. 1	Inheritance		
Date of Exercise	20.07.2016	Date of Upload	19.08.2016

Aim

To develop **Library Management System** using C# for various distinct keeping in my mind the necessary constraints and concepts.

Description

When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the **base class**, and the new class is referred to as the **derived class**. The idea of inheritance implements the **IS-A relationship.** For example, mammal IS A animal, dog IS-A mammal hence dog IS-A animal as well, and so on.

There are two distinct types of inheritance :

Implementation inheritance which is a derived type adopts the base type ' s implementation of each function.
Interface inheritance which inherits only the signatures of the functions and does not inherit any implementations.

Base and Derived Classes

A class can be derived from more than one class or interface, which means that it can inherit data and functions from multiple base classes or interfaces.

The syntax used in C# for creating derived classes is as follows:

```
<acess-specifier> class <base_class>
{
  ...
}
class <derived_class> : <base_class>
{
  ...
}
```

Abstract Classes and Functions

- An abstract class cannot be instantiated
- Abstract function does not have an implementation
- Must be overridden in any non - abstract derived class
- An abstract function is automatically virtual
- If any class contains any abstract functions, that class is also abstract

```
abstract class Building
{
private bool damaged = false; // field
public abstract decimal CalculateHeatingCost(); // abstract method
}
```

Sealed Classes and Methods

- Sealed class, can ’ t be inherit
- Sealed method, can ’ t be override
-

```
sealed class FinalClass {
// etc }
class DerivedClass : FinalClass
// wrong. Will give compilation error
{ // etc }
class MyClass { public sealed virtual void FinalMethod() { // etc. } }
class DerivedClass : MyClass {
public override void FinalMethod()
// wrong. Will give compilation error { }
 }
```

C# does not support multiple inheritance. However, you can use interfaces to implement multiple inheritance. The following program demonstrates this:

```
public interface Compute : Addition
{
void Sub();
}
public interface Addition
{
void Add();
}
public interface Subtraction
{
void Sub();
}
class Computation : Addition, Subtraction
{
public void Add(){ }
public void Sub(){ }
}
```

Program

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;

namespace LibraryManagementSystem
{
    //books structure
    public struct Books
    {
        public string title;
        public string author;
        public string subject;
        public  int book_id ;
        public string status;
        public int book_count;
        public string book_status;
    }

    //Abstract class as methods have no implementation
    abstract class Security {
        public static string ReadPassword() {
            return "for masking input";
        }

        public static void CheckPwd(Books[] arr) { }

    }

    //Inheriting Security class functions with same signature
    class Program:Security
    {
        public static int student_book_count = 3;
        public static int faculty_book_count = 5;

         public static  List<String>  book_issue_list = new List<String>();

        public static Hashtable BookIssueHash = new Hashtable();

        public static int count = 1;

        static void Main(string[] args)
        {

            Books[] bookarr = new Books[10];
            String[] users = {"admin","student","faculty"};
            CheckPwd(bookarr);

        }
        //Method Hiding
        public new static string ReadPassword()
```

```
        {
            string password = "";
            ConsoleKeyInfo info = Console.ReadKey(true);
            while (info.Key != ConsoleKey.Enter)
            {
                if (info.Key != ConsoleKey.Backspace)
                {
                    Console.Write("*");
                    password += info.KeyChar;
                }
                else if (info.Key == ConsoleKey.Backspace)
                {
                    if (!string.IsNullOrEmpty(password))
                    {
                        // remove one character from the list of password characters
                        password = password.Substring(0, password.Length - 1);
                        // get the location of the cursor
                        int pos = Console.CursorLeft;
                        // move the cursor to the left by one character
                        Console.SetCursorPosition(pos - 1, Console.CursorTop);
                        // replace it with space
                        Console.Write(" ");
                        // move the cursor to the left by one character again
                        Console.SetCursorPosition(pos - 1, Console.CursorTop);
                    }
                }
                info = Console.ReadKey(true);
            }
            // add a new line because user pressed enter at the end of their password
            Console.WriteLine();
            return password;
        }
        //Method Hiding
        public new static void  CheckPwd(Books[] arr) {
            Console.WriteLine("**-------------------------Welcome to Library Management
System--------------------------**");
            Console.WriteLine("\n");
            Console.Write("Username:");
            String user = Console.ReadLine();
            user = user.ToLowerInvariant();
            Console.Write("Password:");
            var password = ReadPassword();
                if (user.Equals("admin") && password.Equals("admin"))
                {
                //for admins
                AdminClass.AdminFun(arr);

                }
                else if (user.Equals("faculty") && password.Equals("faculty"))
                {
                //for faculty
                FacultyClass.FacultyFun(arr);

                }
                else if (user.Equals("student") && password.Equals("student"))
                {
                //for student
```

```
                StudentClass.StudentFun(arr);
                }

                else
                {
                    Console.WriteLine("\aBad Attempt!!! :( ");
                    CheckPwd(arr);
                }

        }

    }

    class FacultyClass {

        public static void FacultyFun(Books[] arr)
        {
            StudentClass stuobj = new StudentClass();
            AdminClass adminobj = new AdminClass();
            FacultyClass facobj = new FacultyClass();
            Console.WriteLine();
            Console.WriteLine("1.Search for Books\n2.Reserve Book\n3.Borrow
Book\n4.Return Book\n5.Renew a Book\n6.View Book Issue
Details\n7.Logout\n_____________________________");
            int input = Convert.ToInt32(Console.ReadLine());
            switch (input)
            {
                case 1:
                    stuobj.SearchBooks(arr);
                    FacultyFun(arr);
                    break;

                case 2:
                    Console.WriteLine("1.Reserve via Search ");
                    Console.WriteLine("2.Reserve via Book_id
\n_____________________________");
                    int cho = Convert.ToInt32(Console.ReadLine());
                    if (cho == 1) { stuobj.SearchBooks(arr);facobj.ReserveBooks(arr);  }
                    else { facobj.ReserveBooks(arr); }
                    FacultyFun(arr);

                    break;
                case 3:
                    Console.WriteLine("1.Borrow via Search ");
                    Console.WriteLine("2.Borrow via Book_id \n_____________________________");
                    int ch = Convert.ToInt32(Console.ReadLine());
                    if (ch == 1) { stuobj.SearchBooks(arr); }
                    else { stuobj.BorrowBooks(arr, "faculty"); }
                    FacultyFun(arr);
                    break;

                case 4:
                    stuobj.ReturnBooks(arr,"faculty");
                    FacultyFun(arr);
                    break;

                case 6:
```

```
                    stuobj.BookIssueDeatils();
                    FacultyFun(arr);
                    break;

                case 5:
                    adminobj.ViewBooks(arr);
                    FacultyFun(arr);
                    break;
                case 7:
                    stuobj.LoginPage(arr);
                    break;

                default:
                    Console.WriteLine("_____________________Invalid Choice :(
_____________________");
                    FacultyFun(arr);
                    break;
            }
        }

        public void ReserveBooks(Books[] bookarr)
        {
            Console.WriteLine("_____________________Please Enter the book_id to be
Reserved ?_____________________");
            int id = Convert.ToInt32(Console.ReadLine());
            if (id != 0 && id > 0)
            {
                if ((bookarr[id].book_id).Equals(id))
                {

                    if (Program.student_book_count > 0 && Program.faculty_book_count > 0)
                    {
                        if ((bookarr[id].book_count) > 0)
                        {
                            String val = "reserved";
                            val = val.ToUpper();
                            bookarr[id].book_status = val;
                            Console.WriteLine("_____________________The book is
Reserved successfully :)_____________________");
                        }
                        else
                        { Console.WriteLine("_____________________Out of Stock :(
_____________________"); }
                    }
                    else { Console.WriteLine("_____________________Dear Staff, You
have currently issued  3 Books_____________________"); }
                }
                else { Console.WriteLine("_____________________Book_ID is not matching
:( _____________________"); ReserveBooks(bookarr); }
            }
            else
            {
                Console.WriteLine("_____________________Enter a Valid Book_Id
Please_____________________");
                ReserveBooks(bookarr);
            }
        }
```

```
    }

   interface CommonFunctions {
         void LoginPage(Books[] arr);
    }


    class AdminFunctions {
        public virtual void InsertBooks(Books[] bookarr) { }
        public virtual void ViewAccounts() { throw new NotImplementedException(); }
        public virtual void ViewBooks(Books[] arr) { }
    }

    class AdminClass:AdminFunctions,CommonFunctions {

        public static void AdminFun(Books[] arr)
        {
            AdminClass adminobj = new AdminClass();
            Console.WriteLine("1.Insert Books\n2.View Books\n3.Alter Books\n4.Alter
Accounts\n5.Logout\n_____________________");
            int input = Convert.ToInt32(Console.ReadLine());
            switch(input)
            {
                case 1:
                    adminobj.InsertBooks(arr);
                    AdminFun(arr);
                    break;

                case 2:
                    adminobj.ViewBooks(arr);
                    AdminFun(arr);
                    break;

                case 3:
                    break;

                case 4:
                    break;

                case 5:
                    adminobj.LoginPage(arr);
                    break;

                default:
                    Console.WriteLine("Invalid Choice :(");
                    AdminFun(arr);
                    break;
            }

        }
        //sealing the admin functions
        public override sealed  void InsertBooks(Books[] bookarr)
        {
            Console.WriteLine("How many you want to enter ?");
            int num_books = Convert.ToInt32(Console.ReadLine());
            int initial = Program.count;
            Console.WriteLine(initial);
```

```
            for (int i = initial; i <(num_books+initial); i++)
            {
                bookarr[i].book_id =(Program.count)++;
                Console.WriteLine("Enter the Title of the Book?");
                bookarr[i].title = (Console.ReadLine()).ToUpper();
                Console.WriteLine("Enter the Author of the Book?");
                bookarr[i].author = (Console.ReadLine()).ToUpper();
                Console.WriteLine("Enter the Subject of the Book?");
                bookarr[i].subject = (Console.ReadLine()).ToUpper();
                Console.WriteLine("Enter the Total Count of the Book?");
                bookarr[i].book_count = Convert.ToInt32(Console.ReadLine());
                bookarr[i].book_status = "unreserved";
            }
            Console.WriteLine("The Entered Books are :");
            ViewBooks(bookarr);
        }
        //Using the base function for throwing Exception
        public override void ViewAccounts()
        {
            base.ViewAccounts();
        }
        //sealing the admin functions
        public override sealed void ViewBooks(Books[] arr)
        {
            int length = arr.Length;

Console.WriteLine("Book_ID\t\tTITLE\t\tAUTHOR\t\tSUBJECT\t\tBOOK_COUNT\t\tStatus");
            for (int i = 0; i < length; i++)
            {
                if (arr[i].book_id != 0) {

Console.WriteLine("{0}\t\t{1}\t\t{2}\t\t{3}\t\t{4}\t\t{5}",arr[i].book_id, arr[i].title,
arr[i].author, arr[i].subject,arr[i].book_count,arr[i].book_status);
                }
            }

        }


        public void LoginPage(Books[] arr) {
            Console.WriteLine("Logging out of the System");
            Program.CheckPwd(arr);
        }
    }

    interface Bookfunctions {
        void SearchBooks(Books[] bookarr);
        void LocalSearch(Books[] bookarr, int choice, String title, String author, int
id);
        void BorrowBooks(Books[] bookarr, String username);
        void BookIssueDeatils();
        void ReturnBooks(Books[] bookarr, String username);
    }

    interface Displayfunction {
         void LocalView(Books[] arr, int[] matchval);
    }
```

```
    //Deriving Functions from Interface Bookfunctions
    //Deriving Functions from Interface Displayfunctions
    class StudentClass:CommonFunctions,Bookfunctions,Displayfunction {

        public static void StudentFun(Books[] arr)
        {
            StudentClass stuobj = new StudentClass();
            AdminClass adminobj = new AdminClass();
            Console.WriteLine();
            Console.WriteLine("\n1.Search for Books\n2.Borrow Book\n3.Return Book\n4.View
Book Issue Details\n5.View All Books\n6.Log Out\n_____________________________");
            int input = Convert.ToInt32(Console.ReadLine());
                switch (input)
                {
                    case 1:
                        stuobj.SearchBooks(arr);
                        StudentFun(arr);
                        break;

                    case 2:
                        Console.WriteLine("1.Borrow via Search ");
                        Console.WriteLine("2.Borrow via Book_id
\n_____________________________");
                        int ch = Convert.ToInt32(Console.ReadLine());
                    if (ch == 1){ stuobj.SearchBooks(arr); }
                    else { stuobj.BorrowBooks(arr,"student"); }
                    StudentFun(arr);
                        break;

                    case 3:
                        stuobj.ReturnBooks(arr,"student");
                        StudentFun(arr);
                        break;

                    case 4:
                    stuobj.BookIssueDeatils();
                        StudentFun(arr);
                        break;

                case 5:
                    adminobj.ViewBooks(arr);
                    StudentFun(arr);
                    break;
                case 6:
                    stuobj.LoginPage(arr);
                    break;

                    default:
                        Console.WriteLine("Invalid Choice :(");
                        StudentFun(arr);
                        break;
                }
        }

        public  void LoginPage(Books[] arr)
        {
```

```
            Console.WriteLine("Logging out of the System");
            Program.CheckPwd(arr);
        }

        public void SearchBooks(Books[] bookarr) {
            String title = "", author = "";
            Console.WriteLine("Please Input the Search type ");
            Console.WriteLine("1.Title\n2.Author\n3.Book_id\n4.Title & author");
            int cho = Convert.ToInt32(Console.ReadLine());
            switch (cho) {
                case 1:
                    Console.WriteLine("Enter the title of the Book ?");
                    title = Console.ReadLine();
                    title = title.ToUpper();
                    LocalSearch(bookarr, 1,title,author,0);
                    break;
                case 2:
                    Console.WriteLine("Enter the author of the Book ?");
                    author = Console.ReadLine();
                    author = author.ToUpper();
                    LocalSearch(bookarr, 2,title,author,0);
                    break;
                case 3:
                    Console.WriteLine("Enter the id of the Book ?");
                    int id = Convert.ToInt32(Console.ReadLine());
                    LocalSearch(bookarr, 3, title,author,id);
                    break;
                case 4:
                    Console.WriteLine("Enter the title of the Book ?");
                    title = Console.ReadLine();
                    title = title.ToUpper();
                    Console.WriteLine("Enter the author of the Book ?");
                    author = Console.ReadLine();
                    author = author.ToUpper();
                    LocalSearch(bookarr, 4, title, author,0);
                    break;
                default:
                    break;
            }

        }

        public void LocalSearch(Books[] bookarr,int choice,String title,String author,
int id) {
            int[] array =new int[bookarr.Length];
            int countval= 0;
            if (choice == 1)
            {
                for (int i = 1; i < bookarr.Length; i++)
                {
                    if ((bookarr[i].book_id) != 0) {
                        if ((bookarr[i].title).Equals(title))
                        { array[countval++] = bookarr[i].book_id; }
                    }
                }
                LocalView(bookarr, array);
            }
```

```
            else if (choice == 2)
            {
                for (int i = 1; i < bookarr.Length; i++)
                {
                    if ((bookarr[i].book_id) != 0)
                    {
                        if ((bookarr[i].author).Equals(author))
                        { array[countval++] = bookarr[i].book_id; }
                    }
                }
                LocalView(bookarr, array);
            }
            else if (choice == 3)
            {
                for (int i = 1; i < bookarr.Length; i++)
                {
                    if ((bookarr[i].book_id) != 0)
                    {
                        if ((bookarr[i].book_id).Equals(id))
                        { array[countval++] = bookarr[i].book_id; }
                    }
                }
                LocalView(bookarr, array);
            }
            else {
                for (int i = 1; i < bookarr.Length; i++)
                {
                    if ((bookarr[i].book_id) != 0)
                    {
                        if (((bookarr[i].title).Equals(title))||
((bookarr[i].author).Equals(author)))
                        { array[countval++] = bookarr[i].book_id; }
                    }
                }
                LocalView(bookarr, array);
            }
        }

        public void LocalView(Books[] arr,int[] matchval) {
            int length = arr.Length;
            if (matchval[0] == 0) { Console.WriteLine("\n_____Sorry :(
No records are found_____"); }
            else {
                Console.WriteLine("Book_ID\t\tTITLE\t\tAUTHOR\t\tSUBJECT");
                Console.WriteLine("_______\t\t_____\t\t______\t\t_______");
                foreach (int i in matchval)
                {
                    //Console.WriteLine("The val of i is " + i);
                    if (arr[i].book_id != 0)
                    {
                        Console.WriteLine();
                        Console.WriteLine("{0}\t\t{1}\t\t{2}\t\t{3}", arr[i].book_id,
arr[i].title, arr[i].author, arr[i].subject);
                    }

                }
            }
```

```
        }

        public void BorrowBooks(Books[] bookarr,String username) {
            Console.WriteLine("Please Enter the book_id to be borrowed ?");
            int id = Convert.ToInt32(Console.ReadLine());
            if (id != 0)
            {
                if ((bookarr[id].book_id).Equals(id))
                {
                    if (username.Equals("student"))
                    {
                        if (Program.student_book_count > 0)
                        {
                            if ((bookarr[id].book_count) > 0)
                            {
                                String val = "reserved";val = val.ToUpper();
                                if (bookarr[id].book_status.Equals(val)) {
Console.WriteLine("_____________________The book is Reserved by
faculty_____________________"); }
                                else {
                                    (bookarr[id].book_count)--;
(Program.student_book_count)--;
                                    String value = bookarr[id].book_id + "\t" + username
+ "\t" + DateTime.Now + "\t" + (DateTime.Now).AddDays(15);
                                    value = value.ToUpper();
                                    //Program.book_issue_list.Add(value);
                                    int IssueNo = new Random().Next(999, 99999);
                                    Program.BookIssueHash.Add(IssueNo, value);
                                    Console.WriteLine("\tDear" + username + ",your Issue
no is [{0}]", IssueNo);
                                    Console.WriteLine("\tBook Issued Successfully :) ");
                                }
                            }
                            else
                            { Console.WriteLine("_____________________Out of Stock :(
_____________________"); }
                        }
                        else { Console.WriteLine("Dear Student, You cannot issue more
than 3 Books"); }
                    }
                    else if (username.Equals("faculty"))
                    {
                        if (Program.faculty_book_count > 0)
                        {
                            if ((bookarr[id].book_count) > 0)
                            {
                                (bookarr[id].book_count)--; (Program.faculty_book_count)-
-;
                                String value = bookarr[id].book_id + "\t" + username +
"\t" + DateTime.Now + "\t" + (DateTime.Now).AddYears(1);
                                value = value.ToUpper();
                                Program.book_issue_list.Add(value);
                                int IssueNo = new Random().Next(999, 99999);
                                Program.BookIssueHash.Add(IssueNo, value);
                                Console.WriteLine("\tDear" + username + ",your Issue no
is [{0}]", IssueNo);
```

```
                                Console.WriteLine("_____________________Book Issued
Successfully :) _____________________");
                            }
                            else {Console.WriteLine("_____________________Out of Stock
:(_____________________");}
                        }
                        else { Console.WriteLine("_____________________Dear staff, You
cannot issue more than 5 Books_____________________"); }
                    }
                }
                else { Console.WriteLine("_____________________Book_ID is not matching
:(_____________________"); BorrowBooks(bookarr, username); }
            }
            else {
                Console.WriteLine("_____________________Enter a Valid Book_Id
Please_____________________");
                BorrowBooks(bookarr,username);
            }
        }

        public void BookIssueDeatils() {
            ICollection key = Program.BookIssueHash.Keys;
            if (key.Count != 0)
            {
                Console.WriteLine("\nBook_id\tHolder\tIssued On\t\tTo be Returned");
                Console.WriteLine("_______\t______\t_________\t\t_____________");
                // Get a collection of the keys.

                foreach (int k in key)
                {
                    Console.WriteLine(Program.BookIssueHash[k]);
                }
            }
            else {
                Console.WriteLine("\n\t---------------No Issue Records are found---------
------\t");
            }

        }

        public void ReturnBooks(Books[] bookarr,String username) {
            Console.WriteLine("Enter the Issue No to return your books ?");
            int Issno = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Your issue record is:
\n{0}",Program.BookIssueHash[Issno]);
            String text = Program.BookIssueHash[Issno].ToString();
            String[] sub = text.Split('\t');
            //Console.WriteLine("After Splitting");
            //foreach(String data in sub) { Console.WriteLine(data); }
            int id = Convert.ToInt32(sub[0]);
            Program.BookIssueHash.Remove(Issno);
            if (username.Equals("student")) { (Program.student_book_count)++; } else {
(Program.faculty_book_count)++; }
            (bookarr[id].book_count)++; bookarr[id].book_status="unreserved";
            Console.WriteLine("_____________________Book is returned Successfully
:)_____________________");
        }
```

```
    }

}
```

Output

```
**--------------------------Welcome to Library Management System--------------------------**


Username:admin
Password:*****
1.Insert Books
2.View Books
3.Alter Books
4.Alter Accounts
5.Logout
_____________________
1
How many you want to enter ?
2
1
Enter the Title of the Book?
html
Enter the Author of the Book?
thomas
Enter the Subject of the Book?
web
Enter the Total Count of the Book?
2
Enter the Title of the Book?
C Pr
Enter the Author of the Book?
yash
Enter the Subject of the Book?
Programng
Enter the Total Count of the Book?
5
The Entered Books are :
Book_ID         TITLE           AUTHOR          SUBJECT         BOOK_COUNT              Status
1               HTML            THOMAS          WEB             2               unreserved
2               C PR            YASH            PROGRAMNG               5               unreserved
1.Insert Books
```

```
1.Search for Books
2.Borrow Book
3.Return Book
4.View Book Issue Details
5.View All Books
6.Log Out
_____________________________
2
1.Borrow via Search
2.Borrow via Book_id
_____________________________
1
Please Input the Search type
1.Title
2.Author
3.Book_id
4.Title & author
1
Enter the title of the Book ?
C Pr
Book_ID          TITLE            AUTHOR           SUBJECT
_______          _____            ______           _______

2                C PR             YASH             PROGRAMNG


1.Search for Books
2.Borrow Book
3.Return Book
4.View Book Issue Details
5.View All Books
6.Log Out
_____________________________
2
1.Borrow via Search
2.Borrow via Book_id
_____________________________
2
Please Enter the book_id to be borrowed ?
2
        Dearstudent,your Issue no is [64167]
        Book Issued Successfully :)
```

```
1.Search for Books
2.Borrow Book
3.Return Book
4.View Book Issue Details
5.View All Books
6.Log Out
_____________________________
3
Enter the Issue No to return your books ?
64167
Your issue record is:
2       STUDENT 19-AUG-16 21:07:14      03-SEP-16 21:07:14
_____________________________Book is returned Successfully :)_______


1.Search for Books
2.Borrow Book
3.Return Book
4.View Book Issue Details
5.View All Books
6.Log Out
_____________________________
```

If tried to borrowed more than 3 times

```
Please Enter the book_id to be borrowed ?
1
Dear Student, You cannot issue more than 3 Books
```

To reserve the book by faculty

```
Username:faculty
Password:*******

1.Search for Books
2.Reserve Book
3.Borrow Book
4.Return Book
5.Renew a Book
6.View Book Issue Details
7.Logout
_____________________
2
1.Reserve via Search
2.Reserve via Book_id
_____________________
2
_____________________Please Enter the book_id to be Reserved ?_____________________
1
_____________________The book is Reserved successfully :)_____________________

1.Search for Books
2.Reserve Book
3.Borrow Book
4.Return Book
5.Renew a Book
6.View Book Issue Details
7.Logout
_____________________
7
**-------------------------Welcome to Library Management System-------------------------**


Username:admin
Password:*****
1.Insert Books
2.View Books
3.Alter Books
4.Alter Accounts
5.Logout
_____________________
2
Book_ID         TITLE           AUTHOR          SUBJECT         BOOK_COUNT              Status
1               HTML            THOMAS          WEB DEV         2               RESERVED
1.Insert Books
2.View Books
3.Alter Books
4.Alter Accounts
5.Logout
_____________________
```

Issue Details

```
_____________________________
2
Please Enter the book_id to be borrowed ?
1
        Dearstudent,your Issue no is [81823]
        Book Issued Successfully :)


1.Search for Books
2.Borrow Book
3.Return Book
4.View Book Issue Details
5.View All Books
6.Log Out
_____________________________
4

Book_id Holder  Issued On               To be Returned
_______ _______ _________               _________________
1       STUDENT 19-AUG-16 21:16:45      03-SEP-16 21:16:45
```

Result

The above programmed is compiled successfully and the screenshots are well described with successful outputs and constraints.

[Dr. S.P. Jeno Lovesum]