

Department of Computer Science

Gujarat University



Certificate

Roll No: 30

Seat No: _____

This is to certify that Mr./Ms. Rathod Ajinkya student of MCA Semester – II has duly completed his/her term work for the semester ending in June 2020, in the subject of Object Oriented Concept of Programming towards partial fulfilment of his/her Degree of Masters in Computer Applications.

Date of Submission

Internal Faculty

Head of Department

**DEPARTMENT OF COMPUTER SCIENCE
ROLLWALACOMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. – II**

ROLL NO : 30

NAME : Ajinkya Rathod

SUBJECT : Object Oriented Concepts And Programming(OOCP)

NO.	TITLE	PAGE NO.	DATE	SIGN
1	***** Theory Assignment 2 ***** -> Constructor -> Destructor -> Operator Overloading -> Friend Function -> Templates	9	29th June, 2020	
2	***** Theory Assignment 3 ***** -> Inheritance -> Polymorphism -> Pointers -> Files I/O -> STL	33	29th June, 2020	
3	*****Practical Assignment 1***** 1) : Demonstrate the use of static variables in a class by using it to count the number of objects created in the program. 2) : Define a class to represent a bank account. Include the following members: DATA MEMBERS MEMBER FUNCTIONS Name of depositor (1) To assign initial values Account Number (2) To Deposit the amount Type of Account (3) To withdraw an amount after checking the Balance amount in account (4) To display name and balance Write C++ program to handle 10 customers. 3) : Create class STUDENT having rollno, name and age as data members, also take subject with three subjects and initialize their value with minimum passing marks. Using member function, modify marks of student with specific rollno which is given by user. 4) : Create a class Rectangle. The class has attributes length and width, each of which defaults to 1. It has member functions that calculate the perimeter and the area of the rectangle. It has set and get functions for both length and width. The set functions should verify that length and width are each floating-point numbers larger than 0.0 and less	48	29th June, 2020	

D E P A R T M E N T O F C O M P U T E R S C I E N C E
ROLLWALACOMPUTERCENTRE
G U J A R A T U N I V E R S I T Y
M.C.A. – II

R O L L N O : 30

N A M E : Ajinkya Rathod

S U B J E C T : Object Oriented Concepts And Programming(OOCP)

than 20.0.

5) : Define a supplier class. The class contains details about the suppliers. One of the details is the list of items supplied by the supplier. Create a class Item to store item details. The items supplied by any given supplier are different and varying. Use dynamic memory allocation in the constructor function to achieve the solution.

6) : Define a class Student. Add data members as Rollno, Name, Marks_obtained, Max_marks and Percentage. Write member functions for reading values, calculating percentage and printing values of student. Define one more class as MCA_II. MCA_II contains array of students. MCA_II class should contain member functions as Add, delete, modify and replace. MCA_II is to be defined as friend of Student class.

7) : Define a class Car. Add data members as Make, Color, Size, and Cost. Write member functions for reading values and printing values of car. Define one more class as CarCollection. CarCollection contains array of cars. CarCollection class should contain member functions as Add, delete, modify and replace. CarCollection is to be defined as friend of Car class. Write C++ programs to test your classes.

8) : Use Employee and EmpCollection classes. Employee class contains details about employee and EmpCollection contains collection of employees in form of an array. Provide GetSubordinates friend function which returns an object of EmpCollection class which contains details of subordinates of a manager. The employee object describing manager is to be passed as a parameter.

9) : 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 are available, the total cost of the requested copies is displayed; otherwise the message

D E P A R T M E N T O F C O M P U T E R S C I E N C E
ROLLWALACOMPUTERCENTRE
G U J A R A T U N I V E R S I T Y
M.C.A. – II

R O L L N O : 30

N A M E : Ajinkya Rathod

S U B J E C T : Object Oriented Concepts And Programming(OOCP)

	"Required copies not in stock" is display Design a system using a class called books with suitable member functions and constructors. Include the following features also : (a) The number of successful and unsuccessful transaction should be recorded for the purpose of the statistical analysis. Use static data members to keep count of transactions. (b) The price of the books should be updated as and when required. Use a private member function to implement this.		
10)	Write a program to create class ‘Search’ having data members (int a[], x) and define member functions as void input(), void output(), void search(int position), void add(int value) to display result (Use New and Delete).		
4	<p>*****Practical Assignment 2*****</p> <p>1) WAP to use binary operator + add two object of class Numbers having num1 and num2 as its data members and display result.</p> <p>2) WAP to overload operator * which multiply a number to each element of an array within a class arrayContainer and display the result.</p> <p>3) WAP to Overload the *, +,-, ==, != and = operators for the complex class.</p> <p>4) WAP to define an object m1 of matrix class, use m1<< Cout.</p> <p>6) WAP to define a class Date with properties int month; int day; int year; overload the following operators. 6.1 + operator [a+b] (a is of date type and b is an integer), use the assumption that all years all years have 360 days and months 30 days. 6.2 – operator [a-b(same as above)] 6.3 = operator 6.4 <,<=,>,>= 6.5 ++,--[post and pre both]</p> <p>7) WAP to define a class Time with properties int hour; int minute; int second; overload the following operators. 7.1) + operator [a+b] (a is of time type and b is an integer) 7.2) – operator [a-b(same as above)]</p>	90	

**DEPARTMENT OF COMPUTER SCIENCE
ROLLWALACOMPUTERCENTRE
GUJARATUNIVERSITY
M.C.A. – II**

R O L L N O : 30

N A M E : Ajinkya Rathod

S U B J E C T : Object Oriented Concepts And Programming(OOCP)

7.3) = operator

7.4) <,<=,>,>=

7.5) ++,--[post and pre both]

8) Write a menu driven program that can perform the following functions on strings. (Use overloaded operators where possible).(Do not use predefined string class)

- 1. Compare two strings for equality (== operator)**
- 2. Check whether first string is smaller than the second (<= operator)**
- 3. Copy the string to another**
- 4. Extract a character from the string (Overload [])**
- 5. Reverse the string**
- 6. Concatenate two strings (+ operator) Console I/O and Manipulators:**

9) Design a manipulator to provide the following output specifications for printing float values (i) 5 column width (ii) Right justified (iii) 2 digits precision (iv) Filling unused spaces with +

10) Define a class marksheet. The class should contain a function PrintMarkSheet such that it prints the marksheets of a given student with three subject names and five marks for each subject. Define manipulators for displaying headings and footnotes. The function should display marksheets with respective headings and class. The marks should be aligned under the headings (Use either ios functions or manipulators).

**DEPARTMENT OF COMPUTER SCIENCE
ROLLWAL COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. – II**

R O L L N O : 30

N A M E : Ajinkya Rathod

S U B J E C T : Object Oriented Concepts And Programming(OOCP)

5	<p>*****Practical Assignment 3*****</p> <p>1) Write a program to generate templates function for swapping values of variables and show its use with integer, float and character type of data as input.</p> <p>2) Write an object oriented program to implement a generic Stack. Incorporate all the possible operation on Stack in the program.</p> <p>3) Write a generic function that will sort a character string, integer and float value. Create a menu with appropriate options and accept the values from the user.</p> <p>4) Write a template function called find(). This function searches an array for an object. It returns either the index of the matching object (if one is found) or -1 if no match is found.</p> <p>5) WAP Implement template sort with a non type size. Inheritance</p> <p>6) WAP to create base class Book having int id and char name as data members and respective functionality, show following types of inheritance and display the details of each kind of books, also calculate the total no of each type of books in proper format. Simple inheritance with derived class Sales Hierarchical inheritance with derived classes academics and thrillers Show use of constructor and destructor in above examples of inheritance.</p> <p>7) WAP to create student having data members (rollno, name, stream) as base class. Derive class subject with marks of 5 subjects and apply respective functionality. Calculate final result and display details of each student from derived class. (multilevel inheritance)</p> <p>8) An educational institution wishes to maintain a database</p>	173 29th June, 2020		
----------	---	---	--	--

D E P A R T M E N T O F C O M P U T E R S C I E N C E
ROLLWALACOMPUTERCENTRE
G U J A R A T U N I V E R S I T Y
M.C.A. – II

R O L L N O : 30

N A M E : Ajinkya Rathod

S U B J E C T : Object Oriented Concepts And Programming(OOCP)

of its employees.

The database is divided into a number of classes whose hierarchical relationships are shown in fig-1. The figure also shows the minimum information required for each class. Specify all the classes and define function to create the database and retrieve individual information as and when required. Write parameterized constructor for each class in the hierarchy.

9) Consider a class network of fig 2. The class master derives information from both account and admin classes which in turn derived derive information from the class person. Define all the four classes and write a program to create, update and display the information contained in master objects.

10) Create a class student from which the classes test and sports are derived. The class student has the name and rollno of the student. The class test has the marks of the internal test and the sports class has the marks of the sports test. The class student contains a virtual function display() which are implemented in the classes test and sports. Write a program which will take relative information and display it using pointer of the base class.

11) Write a Program to perform following operation on text file :

11.1) write content in a text file

11.2) read content from file

11.3) count no of word and no of lines in a file

11.4) copy contents of one file to another file

12) Write a program to create a file student to store name and marks of 5 students and then display them.

13) Define a class Result which contains the result of an MCA II written test. It should take list from a file and display on the screen such that at a time only ten candidates information is printed on the screen.

**DEPARTMENT OF COMPUTER SCIENCE
ROLLWALACOMPUTERCENTRE
GUJARATUNIVERSITY
M.C.A. – II**

R O L L N O : 30

N A M E : Ajinkya Rathod

S U B J E C T : Object Oriented Concepts And Programming(OOCP)

14) Use an Employee Class to write records of employee to a file. Include a menu that will allow the user to select any of the following features

- a. Add a new record.**
- b. Modify an existing record.**
- c. Retrieve and display an entire record for a given name.**
- d. Generate a complete list of all names, addresses and telephone numbers.**
- e. End of the computation.**

15) Write a program that stores and displays the records of the customer from a file the following information for account of the customer is to be stored.

Account no, account type, name, old balance, new balance, last payment, date of last payment. Also display the current account status by comparing

current payment and previous balance. Also calculate the current balance by subtracting the current payment from the previous balance.

16) WAP to create namespace having function for total_marks. Show its use in class ‘marks’ of students, display total marks of subjects using namespace.

17) WAP to define a vector and use it for student class to store and display information about student (Use STL).

Name: Aishwarya
Rathod

Subject: OOP

Assignment: 2

Assignment - 2

1. What is constructor?

Ans

Constructor is a method in a class which is invoked automatically when we declare a object.

Explicit Constructor

When a constructor is not stated, compiler automatically calls constructor.

But sometimes we need to give initial values we need explicit constructor for that.

Parameterized Constructor

The constructor having parameters is called Parameterized constructor.

or Explain copy constructor?

Ans Assignment operator works on primitive datatypes. what if we want it in user-defined datatype i.e. class.

So, we used copy constructor.

#include <iostream>

using namespace std;

class point

int x, y;

public:

point (int a, int b) :
x = a,

x = b;

}

void display () {

cout << a << b;

}

```
int main() {  
    point p1(1, 1);  
    p1.display();  
    cout << p1(100);  
    cout << A;
```

Q3 Explain M16

1. Member wise initialisation
of a class using constructor
function.
Ans. If starts with ; od has all
3 fildes separated by comma
2. The initialization has a normal
syntax of initialization
; a (-value) value .

e.g. `#include < iostream>`
`using namespace std;`

thus Time t

parts :-

int hours;
int minutes;
int seconds;

void showTime() {

cout << hours <<
min <<
secs;

Tra (int hours, int minutes,) {
int seconds;

~~Time~~

hours (+ hours),

minutes (+ minutes)

seconds (-1 - seconds)

2 4

3;

void main()

{

Time Time1(12,15,15);

Time2 ShowTime();

}

Q4 What is destructor? Why we need it?

A5 A destructor is more suggests which are if removed or constructors within delete.

Matrix:: ~Matrix()

2

for (int i = 0; i < n; i++) {

 delete p3;

}

class Test &

public:

test() {

 count++;

}

~test() {

 count--;

}

Q5)

Explain unary & binary
operator.

A5)

Unary

Let us consider the
unary minus operator $-$
operator when used as a
unary, takes just one operand

→ We know that this operator changes the sign of operand when applied to both data item.

→ The answer will be when applied to an object should change the sign of each of its data

Q) #include <iostream>
using namespace std;

Class Space 2

```
int x;  
int y;  
int z;
```

public:

```
void addData()  
{  
    x + y  
    z  
}
```

void openc();

y;

int main() {

space =

3, opent a C(10, ^ 10);

~~3, 10~~

y

Q6

Explain need of friend as operator function in overriding an operator.

Ans As returning value, we can use friend as operator朋友

→ At first, it is a handle.

There are two cases where it is really important to use friend function.

→ Instead if we want to provide we can,

→ If we use friend, both cases can be understood

~~int ext_s()~~

Q8

What is template function?
Explain non generic parameter
in template function.

Ans

Function templates are
generic functions which work
for any data type that is
passed to them

- The data type is not specified while writing the function
- while using

Non-numerical parameters.

1. we can also pass non-numeric arguments to a function.
2. However pass, we can see at all to print statements shown on the following.

template < class T >
 |
 | < typename T >

NEW INDIA
PAGE NO.: DATE

Old compilers don't support this

Template function in C++.

- > A template is simple, yet powerful tool.
- > The idea is to pass data type as parameter; so we don't need to write same code for different data types.
- > For instance, a company need to sort() different data types.

Rather than writing and maintaining multiple codes, if we can write sort() once and pass data type as parameter.

Syntax

template < typename * T >

keyword "class" can also be used

These are expanded at compile time. They are like macros.

Difference is compiler does type checking before template expansion.

Source code contains only function/class, but compiled code may contain multiple copies of functions/classes.

template <type>max T>

T mymax(T x, T y)

{
return (x > y) ? x : y;
}

int main()

{
cout << "Enter x";
cin >> x;

cout << "Enter y";
cin >> y;
}

a = myMax < int >(3, 7);

b = myMax < char >('q', 'e');
cout << a << b;
return 0;

}

Compiler internally generates and adds below code.

```
int myMax(int x, int y)
{
    return (x>y) ? x : y;
}
```

```
char myMax(char x, char y)
{
    return (x>y) ? x : y;
}
```

Generic programming enables one programmer to write general algos. which will work with all data types. It eliminates need to create different algos if data type is int, float or a character.

Function Templates



We write generic function that can be used for various data types.

Example:- sort(), max(), min(), printarray().



Class Templates

They are useful when a class defines something that is independent of data type.

Can be useful for classes like.

- linked list
- Binary Tree
- Stack
- Queue
- Array, etc.

B

COD

Explain static data member in class template and explain use of export keyword.

Ans

Export:

The export keyword is useful when follows the second model i.e. when we define a template function at single place and its declaration at other place.

This is useful in instant we can find them useful for other application.

2. static data members in
class template.

Ans

The template class can also
have static data members.

→ The way a static member is
defined is → analog to way
member functions are defined
outside the template class

Q1 Explain ~~advantages~~ of C++
over C.

- A) In C++, object-oriented.
Objects represent classes in C++.
cout is an object of the
output streams class, while also
in an object of the input stream class,
- 2) In C++, it is possible to
have the program accept inputs
known as well-defined manipulators
in C++.

Q12 Explain iOS number function
for floating.

a. width()

Ans It specifies the minimum field width for display.

The width() results itself after the first @

cont of "Rallivit";

b. precision()

It specifies precision, that is number of digits to be displayed after the decimal point.

(3) fill C)

The function fill the subsequent empty portions of field by the character

cont. fill ('+'):

cont. with (0) ;

cont C " Roll over";

(b) setf()

This separates the first few are as before right justified scientific notation.

(c) unsef()

This removes the undo operation in sy().

Q13

what is manipulators.

Ao

manipulators are special function
for formatting.

They can do all the formating
that is done by the i/o
write function.

setw()

Ans

It is set the width of field
cout << setw(5) << dd;

2)

setprecision()

Used to set the
floating point precision after
decimal points.

cout << setprecision(3) << cell;

(3)

setfill()

Used to fill empty obtain
after using any the
cout put.

cout << setfill('x') << str(a);

(4) endl :

Used to line in next line

cout << "Hello" << endl <<
"World".

Output:

Hello
world.

Name: Ajinkya
Rathod

Subject: OOPC

Assignment 3

Q1 what is inheritance? Explain multiple inheritance and how to solve that?

Aⁿ The mechanism of deriving a new class from an old class is called inheritance.

A derived class with only one base class is called single inheritance and with several base classes is called multiple inheritance.

Multiple Inheritance

A class can inherit the attribute of two or more base classes shown in below fig.

→ Multiple inheritance allows us to combine the defining parts of different classes.

NEW INDIA
PAGE NO.: DATE

IB-1

IB-2

IB-3

ID

Q

Explain derivation using
different access modifier?

A:

+ Public derivations:

First be:

The first case where a class
class a derived via a public
access specific base like basic
class. The private member of
base class is copied by derived
class is not visible.

Base

Derived

private:

prt a;

private:

prt a;

public:

int ac;

public:

Void B();

Protected Privacy:

In case of protection the protected member of base class become protected member at derived class.

Object Polymorphism

This type of poly is achieved by function overriding.

Function Overriding:

On other hand consider a sub class has definition of one of sub class. Then this class is said to be overridden.

Q5

Describe use of pointers.

Ans Pointer is address that refers to another data variable by specifying memory address rather than data.

→ Declaring & initializing pointer

datatype * pointer-variable;

Ex) ~~#include <iostream.h>~~
~~#include <conio.h>~~

```
int main() {  
    int a=10;  
    int * p;
```

$p \rightarrow a$;

cout << value is $\rightarrow p$;

$* p = * p + 10;$

y

Q6

Explain

(i) `fseek()`

It is used to move file
pointer.

→ `SEEK - END` → End
`SEEK - SET` → Start
`SEEK - CUR` → Current

(ii) `f.eof()`.

Syntax: int `f.eof`.

The `f.eof` function takes
a file object as argument

and return an integer value
which specifies it end of
file.

It is defined in `iostream`

(ii) `fread()`:

The `fread()` function is used to read a specific no. of characters from given input stream.

syntax: `size_t fread (void *buff,`
`size of size, file fptr)`

#include <fstream>

#include <conio.h>

using namespace std;

int main()

{

FILE *fp;

char buff[100];

fp = fopen("data.txt", "r");

while(!feof(fp)) 2

read(buffer, sizeof(buffer)), fp;

cout << buffer;

)

fclose(fp);

3

(ii) fopen()

The fopen() function opens
a file indicated by

file as return a stream
associated with that file.

If r, wa, rt, we, -> Text

rb, wb, ab, wbt, I

Binary

(V)

close

fout close (file + fptr)

The `fclose()` function - takes a `sfptr` argument, a file stream which is to be closed.
All data that are buffered but not written yet flushed to the o.s. and all unflushed buffered data are discarded.

Q2 Explain Namespace in Java

→ Namespaces can be used
used in various ways.

It's possible to have many
class,

→ Have same declaration about
same access specifier like
public, private, etc.

using namespace etc;

Q9 Advantages of saving a data in binary form:

Ans In binary files, numbers are stored in binary form,

It is stored in ms. D. Though characters are still stored in ASCII equivalents.

→ Binary file, a record is usually of the same file.

It also makes possible to provide file using a random access.

Q10 Explain BTL in detail?

→ STL stands for standard Template library is generic software components and algorithms by objects called containers.

→ STL components which are new part of the standard C++ library are defined in using namespace std.

* Component of STL

→ The STL contains several components but its core are three key components. They are

- Containers
- Algorithm
- Iterator

* Container

A container is an object that actually stores data.

It may store data in organized manner. The STL containers are implemented by template classes and therefore customized to type of data.

* Algorithm

An algo is procedure that is used to process the data contained in the container. The STL includes many different kind of algs.

Iteration

An iterator is an object that points to an element in a container. We can use iterator to move through contents of containers.

We can implement or declare them. Iterators connect algorithms with containers and play a key role in manipulations of data stored in containers.

ASSIGNMENT: 1

ASSIGNMENT: 1

ASSIGNMENT: 1

```
//  
=====  
=====  
// Q1: Static Starts  
  
#include<iostream>  
#include<iomanip>  
  
using namespace std;  
  
class Static_count  
{  
    private :  
        static int count;  
  
    public :  
        Static_count()  
        {  
            count++;  
        }  
  
        void countObj();  
  
};  
  
void Static_count :: countObj()  
{
```

```
    cout << "Static int count: " << count << endl;
}

int Static_count :: count;

// Driver Program
int main()
{
    Static_count sc1;
    sc1.countObj();

    Static_count sc2;
    sc2.countObj();

    Static_count sc3;
    sc3.countObj();

    return 0;
}

// Q1: Static Ends
// =====
```

```
//  
=====  
=====  
// Q2: Bank Account Starts  
  
#include<iostream>  
  
using namespace std;  
  
class BankAccount  
{  
    private :  
        static int count;  
        int id;  
        int balance = 0;  
  
    public :  
        BankAccount()  
        {  
            id = count++;  
        }  
  
        int askChoice();  
        void display();  
        void deposit();
```

```
void withdraw();
void task(int);
};

int BankAccount :: askChoice()
{
    int choice = 0;

    while (!choice) {
        cout << endl << " =====Enter
Choice===== \n\n";
        cout << "\n\nEnter your choice" << endl << " 1.
Check balance" << endl << " 2. Deposit" << endl
        << " 3. Withdraw" << endl;

        cin >> choice;

        if(choice < 1 || choice > 3) {

            cout << endl << " *** Invalid Choice *** ";
            cout << endl << " * T R Y _ A G A I N * ";
            choice = 0;
        }
    }

    return choice;
}
```

```
int askAccountId()
{
    int accountId;
    cout << endl << "Enter your accountId: ";
    cin >> accountId;
    if(accountId < 1 || accountId > 10) {
        cout << endl << " *** Invalid Id *** ";
        cout << endl << " * T R Y _ A G A I N * ";
        accountId = askAccountId();
    }
    return accountId;
}
```

```
void BankAccount :: display()
{
    cout << endl;
    cout << endl << " =====Bank
balance===== \n\n";
    cout << " Id: " << id << endl;
    cout << " Bal. " << balance << endl;
}
```

```
void BankAccount :: task(int choice)
{
    if ( choice == 1 ) { display(); }
    else if ( choice == 2 ) { deposit(); }
```

```
        else if ( choice == 3 ) { withdraw(); }

    else    { cout << "Invalid Choice"; }

}

void BankAccount :: deposit()
{
    int deposit_amt;
    cout << endl << "
=====Deposit===== \n\n";
    cout << "Enter amount to deposit: ";
    cin >> deposit_amt;

    this->balance += deposit_amt;
    this->display();
}

void BankAccount :: withdraw()
{
    int withdraw_amt = 0;
    cout << endl << "
=====Withdraw===== \n\n";

    while(!withdraw_amt) {

        cout << "Enter amount to withdraw: ";
```

```
    cin >> withdraw_amt;

    if(withdraw_amt <= this->balance) {
        this->balance -= withdraw_amt;
    }
    else {
        withdraw_amt = 0;
        cout << endl << " *** Amount you entered is
more than current balance*** ";
    }
    this->display();
}

}
```

```
int BankAccount :: count = 1;

int main()
{
    BankAccount ba[10];

    for (int i = 0; i < 10; ++i) {
        ba[i].display();
    }

    int bank_task = 0;
```

```
while(!bank_task) {  
  
    int accountId = askAccountId();  
  
    int choice    = ba[accountId - 1].askChoice();  
  
    ba[accountId - 1].task(choice);  
    cout << endl << "Enter 0 to continue: ";  
    cin >> bank_task;  
  
}  
  
return 0;  
}
```

// Q2: Bank Account Ends

//

=====

=====

//

=====

=====

// Q3: Student-Class Starts

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

using namespace std;
```

```
class Student
{
    private :
        static int count;
        int roll;
        char name[50];
        int age;
        int marks[3];
```

```
    public :
        Student();
        void getData();
        void display();
        void editDetails();
```

```
};
```

```
Student :: Student()
{
    int subjects = 3;
```

```
roll = ++count;
for (int i = 0; i < subjects; ++i) {
    marks[i] = 35;
}
}

void Student :: getData()
{
    cout << "Enter name of Roll No. " << this->roll << ": ";
    cin >> this->name;

    cout << "Enter age of Roll No. " << this->roll << ": ";
    cin >> this->age;

    cout << endl;
}

void Student :: display()
{
    int subjects = 3;

    cout << endl << " Roll no: " << roll;
    cout << endl << " Name: " << name;
    cout << endl << " Age: " << age;

    for (int i = 0; i < subjects; ++i) {
```

```
        cout << endl << " Subject" << i << ":" <<
marks[i];
}
```

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

```
void Student :: editDetails()
{
```

```
    int marks;
    for (int i = 0; i < 3; ++i) {
        cout << "Enter marks of Subject " << i << endl;
        cin >> marks;
        this->marks[i] = marks;
    }
}
```

```
int askRoll()
{
```

```
    int roll;
    cout << endl << "Enter roll no whose details you want to
edit " << endl;
    cin >> roll;
    return roll;
}
```

```
int Student :: count;
```

```
int main()
{
    int stud_count = 3;
    Student stud[3];
    for (int i = 0; i < stud_count; ++i) {
        stud[i].getData();
        stud[i].display();
    }

    int editNo = 1;

    while (editNo) {
        editNo = askRoll() - 1;
        stud[editNo].editDetails();
        cout << endl << "Press any key to continue or 0 to
exit : ";
        cin >> editNo;
    }

    for (int i = 0; i < stud_count; ++i) {
        stud[i].display();
    }

    return 0;
}
```

```
// Q3: Student Class Ends
```

```
//
```

```
=====
```

```
=====
```

```
//
```

```
=====
```

```
=====
```

```
// Q4: Rectangle Starts
```

```
#include<iostream>
```

```
using namespace std;
```

```
class Rectangle
```

```
{
```

```
    private :
```

```
        float width;
```

```
        float height;
```

```
public :  
    Rectangle()  
    {  
        width = 1.0;  
        height = 1.0;  
  
        this->width = width;  
        this->height = height;  
    }  
  
    void getData();  
    float area();  
    float perimeter();  
};  
  
void Rectangle :: getData()  
{  
    cout << endl;  
    do {  
        cout << "Enter Height(0.0 - 20.00) ";  
        cin >> this->height;  
  
        cout << "Enter Width(0.0 - 20.00) ";  
        cin >> this->width;  
  
    } while(height < 0.0 || height > 20.00 || width < 0.0 ||  
          width > 20.00);
```

```
}
```

```
float Rectangle :: area()
{
    return this->width * this->height;
}
```

```
float Rectangle :: perimeter()
{
    return 2 * (this->width + this->height);
}
```

```
int main()
{
    Rectangle r;
    r.getData();

    cout << endl << "Perimeter: " << r.perimeter() <<
endl;
    cout << endl << "Area: " << r.area() << endl;

    return 0;
}
```

```
// Q4: Rectangle Ends
```

```
//  
=====  
=====  
  
//  
=====  
=====  
  
// Q5: Supplier Starts  
  
#include<iostream>  
#include<string>  
#include <new> // noexcept  
  
using namespace std;  
  
//class Suppliers  
class Suppliers  
{  
    private :  
        char supplier_name[50];  
        int no_of_items;  
  
    public:  
        int getData(void);  
        int putData(void);
```

```
};
```

```
// class items
class Items
{
    private :
        char items[5][50];
    public :
        void getData(int);
        void putData(int);
};
```

```
// Functions of Suppliers Class
```

```
int Suppliers :: getData(void)
{
    this -> no_of_items = 0;

    cout << endl << "Enter Name of Supplier: ";
    cin >> this -> supplier_name;

    while (this -> no_of_items < 1) {
        cout << endl << "Enter no of items supplied by " << this
-> supplier_name << ": ";
        cin >> this -> no_of_items;
```

```
    if (this -> no_of_items < 1 || this -> no_of_items > 5)
        cout << endl << "Enter number of items between 1-
5";
    }

    return this -> no_of_items;
}
```

```
int Suppliers :: putData(void)
{
    cout << endl << "======" << endl;
    cout << this -> supplier_name;

    return this -> no_of_items;
}
```

//Functions of item class

```
void Items :: getData(int no_of_items)
{
    for (int i = 0; i < no_of_items; i++) {
        cout << "Enter Item name: " << i + 1 << ": ";
        cin >> this -> items[i];
    }
}
```

```
void Items :: putData(int no_of_items)
{
    for (int i = 0; i < no_of_items; i++) {
        cout << endl << "\t" << this -> items[i];
    }
}

// Driver Code
int main ()
{
    int number_of_suppliers;
    int no_of_items;
    cout << "How many suppliers are there ";
    cin >> number_of_suppliers;

    Suppliers *suppliers = new(nothrow)
Suppliers[number_of_suppliers];

    Items *items = new(nothrow) Items[number_of_suppliers];

    if(!suppliers || !items) {
        perror ("Error");
    }

    for (int i = 0; i < number_of_suppliers; i++) {
        no_of_items = suppliers[i].getData();
        items[i].getData(no_of_items);
    }
}
```

```
}
```

```
for (int i = 0; i < number_of_suppliers; i++) {  
    no_of_items = suppliers[i].putData();  
    items[i].putData(no_of_items);  
}  
  
cout << endl;  
return 0;
```

```
}
```

```
// Q5: Supplier Ends
```

```
//
```

```
=====
```

```
//
```

```
=====
```

```
// Q6: Students, Mca2 Starts
```

```
#include <iostream>
```

```
#include <new> //nothrow
```

```
#include <string>

using namespace std;

class Students
{
public:
    int roll;
    char name[20];
    int marks[3];
    int max_marks;
    float per;

public :
    int getRoll();
    void getData(void);
    void putData(void);

Students ()
{
    this -> max_marks = 0;
}

};

class Mca2 : public Students // public derivation
{
```

```
public :  
    void add(int n);  
    void remove(int);  
    void modify(int);  
    void replace(int, int);  
};  
  
void Students :: getData (void)  
{  
    int total = 0;  
    cout << endl << "Enter roll no: ";  
    cin >> this -> roll;  
  
    cout << endl << "Enter name: ";  
    cin >> this -> name;  
  
    for (int i = 0; i < 3 ; i++) {  
        cout << endl << "Enter marks of Sub " << i + 1 << ":";  
        cin >> this -> marks[i];  
  
        total += this -> marks[i];  
  
        if (this -> marks[i] > this -> max_marks) {  
            this -> max_marks = this -> marks[i];  
        }  
    }  
}
```

```
this -> per = total / 3;

}

void Students :: putData (void)
{
    if (!this -> roll) {
        return;
    }

    cout << endl << "===== " << endl;

    cout << "Roll: " << this -> roll << endl;
    cout << "Name: " << this -> name << endl;

    cout << "Marks : ";
    for (int i = 0; i < 3 ; i++) {
        cout << this -> marks[i] << " ";
    }

    cout << endl;

    cout << "Max Marks: " << max_marks << endl;
    cout << "Per: " << per << endl;
```

```
}
```

```
void Mca2 :: remove(int roll)
{
    if (this -> roll == roll) {
        this -> roll = 0;
    }
}
```

```
void Mca2 :: modify(int roll)
{
```

```
    if (this -> roll != roll) {
        return;
    }
```

```
    int total = 0;
```

```
    cout << endl << "Modifying record for roll:" << this -> roll;
```

```
    cout << endl << "Enter roll no: ";
    cin >> this -> roll;
```

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

```
    cin >> this -> name;

    for (int i = 0; i < 3 ; i++) {
        cout << endl << "Enter marks of Sub " << i + 1 << ": "
    ;
        cin >> this -> marks[i];

        total += this -> marks[i];

        if (this -> marks[i] > this -> max_marks) {
            this -> max_marks = this -> marks[i];
        }
    }

    this -> per = total / 3;

}

void Mca2 :: replace(int roll1, int roll2)
{
    if (this -> roll == roll1) {
        this -> roll = roll2;
    }
    else if (this -> roll == roll2) {
        this -> roll = roll1;
    }
}
```

```
}
```

```
//Driver Code
```

```
int main ()
```

```
{
```

```
    int n = 0, choice = 1;
```

```
    cout << "Enter number of students in Mca2: ";
```

```
    cin >> n;
```

```
    Mca2 *mca2 = new(nothrow) Mca2[50];
```

```
    if (!mca2) {
```

```
        perror ("Error: ");
```

```
}
```

```
    for (int i = 0; i < n; i++) {
```

```
        mca2[i].getData();
```

```
}
```

```
    for (int i = 0; i < n; i++) {
```

```
        mca2[i].putData();
```

```
}
```

```
    while (choice) {
```

```
        cout << endl << "Enter your choice: " << endl;
```

```
    cout << "1. Add" << endl << "2. Remove" << endl <<
"3. Modify" << endl << "4. Replace" << endl << "0. Exit" <<
endl;

    cin >> choice;

    if (choice == 1) {
        int add_n = 0;

        while (add_n < 1) {
            cout << endl << "How many students you want to
add: ";
            cin >> add_n;

            if (add_n < 1) {
                cout << endl << "Enter a positive number : ";
            }
        }

        n += add_n;

        for (int i = n - add_n; i < n; i++) {
            mca2[i].getData();
        }

    }

    else if(choice == 2) {
```

```
int roll;

cout << "Enter roll no you want to remove: ";
cin >> roll;

for (int i = 0; i < n; i++) {
    mca2[i].remove(roll);
}

else if(choice == 3) {
    int roll;

    cout << "Enter roll no you want to modify: ";
    cin >> roll;

    for (int i = 0; i < n; i++) {
        mca2[i].modify(roll);
    }

else if(choice == 4) {
    int roll1, roll2;
```

```
cout << "Enter roll no you want to modify: ";
cin >> roll1 >> roll2;

for (int i = 0; i < n; i++) {
    mca2[i].replace(roll1, roll2);
}

else if(choice == 5) {
    for (int i = 0; i < n; i++) {
        mca2[i].putData();
    }
}

for (int i = 0; i < n; i++) {
    mca2[i].putData();
}

return 0;
}
```

```
// Q6: Student, mca2 Ends
//=====
=====

//=====
=====

// Q7: Cars Starts

#include <iostream>
#include <new> //nothrow
#include <string>

using namespace std;

class Car
{
public:
    int make;
    char color[20];
    int size;
    float cost;

public :
```

```
    int getRoll();
    void getData(void);
    void putData(void);

    friend class CarCollection;
};
```

```
class CarCollection
{
public :
    void add(int n);
    void remove(Car &, int);
    void modify(Car &, int);
    void replace(Car &,int, int);
};
```

```
void Car :: getData (void)
{
    cout << endl << "=====";
    cout << endl << "Enter make no: ";
    cin >> this -> make;

    cout << endl << "Enter color: ";
    cin >> this -> color;
```

```
    cout << endl << "Enter size: ";
    cin >> this -> size;

    cout << endl << "Enter cost: ";
    cin >> this -> cost;

}

void Car :: putData (void)
{
    if (!this -> make) {
        return;
    }

    cout << endl << "===== " << endl;

    cout << "Make: " << this -> make << endl;
    cout << "Color: " << this -> color << endl;
    cout << "Size: " << this -> size << endl;
    cout << "Cost: " << this -> cost << endl;
}

void CarCollection :: remove(Car& s, int make)
{
    if (s.make == make) {
```

```
s.make = 0;  
}  
}  
  
void CarCollection :: modify(Car& s,int make)  
{  
  
    if (s.make != make) {  
        return;  
    }  
  
    cout << endl <<"Modifying record for make:" << s.make;  
  
    cout << endl << "Enter make no: ";  
    cin >> s.make;  
  
    cout << endl << "Enter color: ";  
    cin >> s.color;  
  
    cout << endl << "Enter size: ";  
    cin >> s.size;  
  
    cout << endl << "Enter cost: ";  
    cin >> s.cost;
```

```
}
```

```
void CarCollection :: replace(Car& s,int make1, int make2)
{
    if (s.make == make1) {
        s.make = make2;
    }
    else if (s.make == make2) {
        s.make = make1;
    }
}
```

```
//Driver Code
```

```
int main ()
{
    int n = 0, choice = 1;
```

```
cout << "Enter number of car: ";
cin >> n;
```

```
CarCollection *mca2 = new(nothrow) CarCollection[50];
Car *car = new(nothrow) Car[50];
```

```
if (!mca2) {
```

```
    perror ("Error: ");
}

for (int i = 0; i < n; i++) {
    car[i].getData();
}

for (int i = 0; i < n; i++) {
    car[i].putData();
}

while (choice) {
    cout << endl << "Enter your choice: " << endl;
    cout << "1. Add" << endl << "2. Remove" << endl <<
"3. Modify" << endl << "4. Replace" << endl << "0. Exit" <<
endl;
    cin >> choice;

    if (choice == 1) {
        int add_n = 0;

        while (add_n < 1) {
            cout << endl << "How many car you want to add:
";
            cin >> add_n;

            if (add_n < 1) {
```

```
        cout << endl << "Enter a positive number : ";

    }

}

n += add_n;

for (int i = n - add_n; i < n; i++) {
    car[i].getData();
}

else if(choice == 2) {
    int make;

    cout << "Enter make no you want to remove: ";
    cin >> make;

    for (int i = 0; i < n; i++) {
        mca2[i].remove(car[i], make);
    }
}

else if(choice == 3) {
    int make;
```

```
cout << "Enter make no you want to modify: ";
cin >> make;

for (int i = 0; i < n; i++) {
    mca2[i].modify(car[i],make);
}

else if(choice == 4) {
    int make1, make2;

    cout << "Enter make no you want to modify: ";
    cin >> make1 >> make2;

    for (int i = 0; i < n; i++) {
        mca2[i].replace(car[i], make1, make2);
    }

    else if(choice == 5) {
        for (int i = 0; i < n; i++) {
            car[i].putData();
        }
    }
}
```

```
 }

}

for (int i = 0; i < n; i++) {
    car[i].putData();
}

return 0;
```

// Q7: Cars Ends

//

=====

//

=====

// Q8: Employee, Employee Collection Starts

```
#include <iostream>
```

```
#include <new> //nothrow
#include <string>

using namespace std;

class Employee;

class EmpCollection
{
public :
    void getData(Employee &);
    void putData(Employee &);

};

class Employee
{
public:
    int id;
    char emp_name[20];

public :
    int getRoll();
    friend void EmpCollection :: getData(Employee &);
    friend void EmpCollection :: putData(Employee &);
```

```
};
```

```
void EmpCollection :: getData (Employee &c)
```

```
{
```

```
    cout << endl << "======" ;
```

```
    cout << endl << "Enter id no: " ;
```

```
    cin >> c.id ;
```

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

```
    cin >> c.emp_name ;
```

```
}
```

```
void EmpCollection :: putData (Employee &c)
```

```
{
```

```
    cout << endl << "======" << endl ;
```

```
    cout << "Make: " << c.id << endl ;
```

```
    cout << "Color: " << c.emp_name << endl ;
```

```
}
```

```
//Driver Code
```

```
int main ()
```

```
{
```

```
int n = 0;

cout << "Enter number of employee: ";
cin >> n;

Employee *employee = new(nothrow) Employee[50];
EmpCollection *empCollection = new(nothrow)
EmpCollection[50];

for (int i = 0; i < n; i++) {
    empCollection[i].getData(employee[i]);
}

for (int i = 0; i < n; i++) {
    empCollection[i].putData(employee[i]);
}

return 0;

}

// Q8: Employee, Employee Collection Ends
```

//

=====

=====*****

ASSIGNMENT: 2

ASSIGNMENT: 2

ASSIGNMENT: 2

```
//  
=====
```

// Q10: Manipulators and good look Starts

```
#include<stdio.h>
#include<conio.h>
#include<iostream>
#include<iomanip>
#include<string>
using namespace std;
```

namespace A

{

ostream & PrintLine(ostream & pout)

{

```
pout<<"\n-----"  
-----<<endl;
```

```
return pout;
```

}

}

class Marksheet

```
{  
  
    string sub[3],mark_type[3],course_name;  
    int marks[3][3];  
  
public:  
    Marksheet()  
    {  
        mark_type[0] = "Term Work";  
        mark_type[1] = "Sessional-1";  
        mark_type[2] = "Sessional-2";  
        sub[0]="Advanced Programming";  
        sub[1]="Object Oriented Programming";  
        sub[2]="Database Management System";  
    }  
    void Input()  
    {  
        int j=0,i=0;  
  
        cout<<"\n Enter Course Name:";  
        getline(cin,course_name);  
  
        for(i=0;i<3;i++)  
        {  
            cout<<"\n" <<sub[i];  
            cout<<"\n-----";  
        }  
    }  
};
```

```
    for(j=0;j<3;j++)
    {
        cout<<"\n Enter " <<mark_type[j]<<"
Marks:";

        cin>>marks[i][j];
    }

}

void Display()
{
    int j=0,i=0;

    cout<<"\n Course Name :"<<course_name;

    for(i=0;i<3;i++)
    {
        cout<<"\n" <<sub[i];
        cout<<"\n-----";
        for(j=0;j<3;j++)
        {
            cout<<"\n Enter " <<mark_type[j]<<"
Marks :"<<marks[i][j];
        }
    }
}
```

```
void PrintMarksheet()
{
    int i,j,sum[3],tot=0;

    cout.width(60);
    cout.fill(' ');
    cout<<"GUJRAT UNIVERSITY";

    cout<<A::PrintLine;

//cout<<setw(20);
    cout.width(70);
    cout.fill(' ');
    cout.setf(ios::internal);

    cout<<course_name;
    cout.width(30);
    cout.fill(' ');
    cout<<"|";
    cout<<A::PrintLine;

    cout<<setw(30)<<"Subject
Name"<<setw(20)<<setiosflags(ios::right)<<setfill('
')<<mark_type[0]<<setw(15)<<setfill('
')<<mark_type[1]<<setw(20)<<setfill('
')<<mark_type[2]<<setw(10)<<setfill('
')<<"Total"<<setw(5)<<setfill(' ')<<"|";
    cout<<A::PrintLine;
```

```

    for(i=0;i<3;i++)
    {
        for(j=0;j<1;j++)
        {

            sum[i]=marks[i][j]+marks[i][j+1]+marks[i][j+2];

            cout<<"\n"<<setw(30)<<sub[i]<<setw(20)<<marks[i][j]
            ]<<setw(15)<<marks[i][j+1]<<setw(20)<<marks[i][j+2]<<s
            etw(10)<<sum[i]<<setw(5)<<setfill(' ')<<"|";
        }
    }

    for(i=0;i<3;i++)
    {
        tot=tot+sum[i];
    }

    cout<<A::PrintLine;

    cout<<setw(40)<<setfill(' ')<<"Percentage
is:"<<((tot*100)/900)<<setw(55)<<setfill('
')<<tot<<setw(2)<<setfill(' ')<<"|";;

    cout<<A::PrintLine;

}

};


```

```
void main()
{
    Marksheet m1;
    m1.Input();
    //m1.Display();
    m1.PrintMarksheet();
    getch();
}
```

=====

OUTPUT

=====

Enter Course Name:MASTER OF COMPUTER APPLICATIONS

Advanced Programming

Enter Term Work Marks:45

Enter Sessional-1 Marks:78

Enter Sessional-2 Marks:56

Object Oriented Programming

Enter Term Work Marks:66

Enter Sessional-1 Marks:67

Enter Sessional-2 Marks:89

Database Management System

Enter Term Work Marks:57

Enter Sessional-1 Marks:54

Enter Sessional-2 Marks:56

GUJRAT UNIVERSITY

MASTER OF COMPUTER
APPLICATIONS |

Sessional-2	Subject Name	Term Work	Sessional-1
	Total		

56	Advanced Programming	45	78
	179		

Object Oriented Programming	66	67
89 222		
Database Management System	57	54
56 167		

Percentage is:63

568 |

// Q10: Manipulators and good look Ends

//

//

```
// Q1: Addition Operator Overload Starts
```

```
#include<iostream>
#include<conio.h>
using namespace std;

class Addition
{
    int num;

public:
    void getdata()
    {
        cout<<"\n Enter value:";
        cin>>num;
    }

    void display()
    {
        cout<<"\n Value is:"<<num;
    }

    Addition operator +(Addition tmp)
    {
        tmp.num = tmp.num+ num;
    }
}
```

```
    return tmp;
}

};

void main()
{
    Addition no1,no2,no3;
    no1.getdata();
    no1.display();
    no2.getdata();
    no2.display();

    no3=no1+no2;
    no3.display();
    getch();

}

// Q1: Addition Operator Overload Ends
// =====
```

```
//=====
=====

// Q2: Multiply Area with * Operator Starts

#include<iostream>
#include<conio.h>
using namespace std;

class ArrayContainer
{
    int array[30],n;
public:
    void getdata()
    {
        int i=0;
```

```
cout<<"\n Enter n:";  
cin>>n;  
for(i=0;i<n;i++)  
{  
    cout<<"\n Enter number:";  
    cin>>array[i];  
}  
  
}  
void display()  
{-  
{  
    int i=0;  
    for(i=0;i<n;i++)  
    {  
        cout<<"\n Enter number is:"<<array[i];  
    }  
}  
  
void operator *(int val)  
{  
    int res,i=0;  
    for(i=0;i<n;i++)  
    {  
        array[i] = array[i] * val;  
    }  
}
```

```
    for(i=0;i<n;i++)
    {
        cout<<"\n value is:"<<array[i];
    }
}

void main()
{
    ArrayContainer array1;

    array1.getdata();
    array1.display();

    array1*(2);
    array1.display();

    getch();
}

=====
```

Output

```
=====
```

Enter n:2

Enter number:10

Enter number:20

Enter number is:10

Enter number is:20

value is:20

value is:40

Enter number is:20

Enter number is:40

// Q2: Multiply Area with * Operator Ends

//

=====

=====

```
//  
=====
```

// Q3: Complex Class Starts

```
#include<iostream>
#include<conio.h>
using namespace std
```

```
class Complex  
{
```

```
int real;
```

float img;

public:

Complex()

{

real=0;

img=0.0;

}

void getdata()

{

```
cout<<"\n Enter real no:";
```

```
    cin>>real;
    cout<<"\n Enter img no:";
    cin>>img;
}

void display()
{
    cout<<"\n Real Number is:"<<real;
    cout<<"\n Img Number is:"<<img;
}
```

Complex operator +(Complex tmp)

```
{

    Complex c1 ;
    c1.real=real+tmp.real;
    c1.img=img+tmp.img;
```

```
return (c1);
```

```
}
```

Complex operator -(Complex tmp)

```
{

    Complex c1 ;
    c1.real=real-tmp.real;
    c1.img=img-tmp.img;
```

```
return (c1);
```

```
}
```

```
Complex operator *(Complex tmp)
{
    Complex c1;
    c1.real=real*tmp.real;
    c1.img=img*tmp.img;

    return (c1);
}

Complex operator ==(Complex tmp)
{
    Complex c1;

    if(real==tmp.real)
    {
        c1.real=1;
    }
    if(img==tmp.img)
    {
        c1.img=1.0;
    }

    return (c1);
}

Complex operator !=(Complex tmp)
{
    Complex c1;
```

```
    if(real!=tmp.real)
    {
        c1.real=1;
    }
    if(img!=tmp.img)
    {
        c1.img=1.0;
    }

    return (c1);
}

void operator =(Complex tmp)
{
    real=tmp.real;
    img=tmp.img;
}

};

void main()
{
    Complex c1,c2,c3;

    c1.getdata();
    cout<<"\n -----";
}
```

```
cout<<"\n object 1:";  
cout<<"\n -----";  
c1.display();
```

```
c2.getdata();  
cout<<"\n -----";  
cout<<"\n object 2:";  
cout<<"\n -----";  
c2.display();
```

```
cout<<"\n -----";  
cout<<"\n Addition:";  
cout<<"\n -----";  
c3=c1+c2;  
c3.display();
```

```
cout<<"\n -----";  
cout<<"\n Subtraction:";  
cout<<"\n -----";  
c3=c1-c2;  
c3.display();
```

```
cout<<"\n -----";  
cout<<"\n Multiplication:";  
cout<<"\n -----";
```

```
c3=c1*c2;  
c3.display();  
  
cout<<"\n -----";  
cout<<"\n Equal:";  
cout<<"\n -----";  
c3=c1==c2;  
c3.display();  
  
cout<<"\n -----";  
cout<<"\n Not Equal:";  
cout<<"\n -----";  
  
c3=c1!=c2;  
c3.display();  
  
cout<<"\n -----";  
cout<<"\n Assignment:";  
cout<<"\n -----";  
  
c1=c2;  
cout<<"\n -----";  
cout<<"\n object 1:";  
cout<<"\n -----";  
c1.display();
```

```
cout<<"\n -----";
cout<<"\n object 2:";
cout<<"\n -----";

c2.display();
getch();

}
```

```
/*
=====
Output
=====
```

Enter real no:10

Enter img no:20.10

object 1:

Real Number is:10

Img Number is:20.1

Enter real no:40

Enter img no:20.10

object 2:

Real Number is:40

Img Number is:20.1

Addition:

Real Number is:50

Img Number is:40.2

Subtraction:

Real Number is:-30

Img Number is:0

Multiplication:

Real Number is:400

Img Number is:404.01

Equal:

Real Number is:0

Img Number is:1

Not Equal:

Real Number is:1

Img Number is:0

Assignment:

object 1:

Real Number is:40

Img Number is:20.1

object 2:

Real Number is:40

Img Number is:20.1

*/

```
// Q3: Complex Class Ends  
//  
=====
```

```
//  
=====
```

```
// Q4: Overload <<, >> Operator Starts
```

```
#include<iostream>  
#include<string>  
#include<conio.h>  
using namespace std;
```

```
class student  
{  
    int rollno;  
    string name;  
public:
```

```
    friend ostream & operator <<(ostream & tempout,
student & s)
{
    tempout << "\n Roll number is: " << s.rollno
<< "\n";
    tempout << "Name is: " << s.name << "\n";

    return tempout;
}

    friend istream & operator >>(istream &
tempin,student & s)
{
    cout<<"Enter Roll no:";
    tempin>>s.rollno;
    cout<<"Enter Name:";
    tempin>>s.name;

    return tempin;
}

};
```

```
void main()
{
    student s1;

    cin>>s1;
```

```
cout<<s1;  
getch();  
}
```

```
=====
```

Output

```
=====
```

Enter Roll no:1

Enter Name:shivangi

Roll number is: 1

Name is: shivangi

// Q4: Overload <<, >> Operator Ends

//

```
=====
```

```
=====
```

```
//=====
=====

// Q5: Matrix Starts

#include<iostream>
#include<conio.h>
using namespace std;

class Matrix
{
    int matrix[10][10];

public:
    Matrix(){}

    void input()
    {
        for(int i=0;i<2;i++)
        {
```

```
    for(int j=0;j<2;j++)
    {
        cout<<"\n Enter Value:";
        cin>>matrix[i][j];
    }

}

void display()
{
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<2;j++)
        {
            cout<<"\n Value is:"<<matrix[i][j];
        }
    }
}
```

Matrix friend operator *(Matrix m4,int mul)

```
{

    for(int i=0;i<2;i++)
    {
```

```
        for(int j=0;j<2;j++)
        {
            m4.matrix[i][j]=mul * m4.matrix[i][j];
        }

    }

return m4;

}

Matrix friend operator *(int mul,Matrix m4)
{
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<2;j++)
        {
            m4.matrix[i][j]=mul * m4.matrix[i][j];
        }

    }

    return m4;
}

};
```

```
void main()
{
    Matrix m1,m2,m3;
    cout<<"\n Enter Value for 2*2 matrix";
    m1.input();
    m1.display();
    cout<<"\n -----";
    cout<<"\n Matrix*5";
    cout<<"\n -----";
    m2=m1*5;
    m2.display();

    cout<<"\n -----";
    cout<<"\n 5*Matrix";
    cout<<"\n -----";
    m3=5*m1;
    m3.display();
    getch();
}
```

=====

output

=====

Enter Value for 2*2 matrix

Enter Value:1

Enter Value:2

Enter Value:3

Enter Value:4

Value is:1

Value is:2

Value is:3

Value is:4

Matrix*5

Value is:5

Value is:10

Value is:15

Value is:20

5*Matrix

Value is:5

Value is:10

Value is:15

Value is:20

// Q5: Matrix Ends

//

=====

=====

//

=====

=====

// Q6: Date with class Starts

```
#include<iostream>
#include<string>
#include<conio.h>
using namespace std;

class Date
{
    int month,day,year;
public:
    void getdate()
    {
        cout<<"\n Enter Date:";
        cin>>day>>month>>year;
    }
    void display()
    {
        cout<<"\n Date
is:"<<day<<'/'<<month<<'/'<<year;
    }

    void operator +(int tday)
    {
        day=day + tday;

        while(day>30)
        {
            day=day-30;
        }
    }
}
```

```
month++;
if(month>12)
{
    month=1;
    year++;
}
}
```

```
void operator -(int tday)
{
    day=day - tday;

    while(day<0)
    {
        month--;
        if(month==0)
        {
            month=12;
            year--;
        }
        day=30+day;
    }
}
```

```
int operator >(Date d2)
```

```
{  
    int flag=0;  
  
    if(year>d2.year)  
    {  
        flag=1;  
  
    }  
    else if(year==d2.year)  
    {  
        if(month>d2.month)  
        {  
            flag=1;  
        }  
        else if(month==d2.month)  
        {  
            if(day>d2.day)  
            {  
                flag=1;  
            }  
            else  
            {  
                flag=0;  
            }  
        }  
    }  
}
```

```
    return flag;  
}  
  
int operator <(Date d2)  
{  
    int flag=0;  
  
    if(year<d2.year)  
    {  
        flag=1;  
  
    }  
    else if(year==d2.year)  
    {  
        if(month<d2.month)  
        {  
            flag=1;  
        }  
        else if(month==d2.month)  
        {  
            if(day<d2.day)  
            {  
                flag=1;  
            }  
            else  
            {  
            }  
        }  
    }  
}
```

```
    flag=0;  
}  
}  
}  
return flag;  
}  
  
int operator >=(Date d2)  
{  
    int flag=0;  
  
    if(year>d2.year)  
    {  
        flag=1;  
  
    }  
    else if(year==d2.year)  
    {  
        if(month>d2.month)  
        {  
            flag=1;  
        }  
        else if(month==d2.month)  
        {  
            if(day>=d2.day)  
            {  
                flag=1;  
            }  
        }  
    }  
}
```

```
        }
    else
    {
        flag=0;
    }
}

return flag;
}
```

```
int operator <=(Date d2)
{
    int flag=0;
```

```
    if(year<d2.year)
    {
        flag=1;

    }
    else if(year==d2.year)
    {
        if(month<d2.month)
        {
            flag=1;

        }
        else if(month==d2.month)
```

```
{  
    if(day<=d2.day)  
    {  
        flag=1;  
    }  
    else  
    {  
        flag=0;  
    }  
}  
return flag;  
}  
void operator ++()  
{  
    Date d1;  
    if(day==30)  
    {  
        day=1;  
        if(month==12)  
        {  
            month=1;  
            year++;  
        }  
        else  
            month++;  
    }  
}
```

```
    }
else
{
    day++;
}

}

void operator --()
{

    if(day==1)
    {
        day=30;
        if(month==1)
        {
            month=12;
            year--;
        }
        else
            month--;
    }
    else
    {
        day--;
    }
}
```

```
    }

};

void main()
{
    int day,res,flag=0,ch;
    char ch2='n';

    Date d1,d2;

    cout<<"\n-----";
    cout<<"\n 1.Addition(+) of Days";
    cout<<"\n 2.Subtraction(-) of Days";
    cout<<"\n 3.chack two date (>)"; 
    cout<<"\n 4.chack two date (>=)";
    cout<<"\n 5.chack two date (<)"; 
    cout<<"\n 6.chack two date (<=)";
    cout<<"\n 7.Increment date (++)";
    cout<<"\n 8.Decrement date (--)";
    cout<<"\n-----";

    do
    {

        cout<<"\n\n Enter Your choice:";
```

```
cin>>ch;

switch(ch)
{
    case 1:   d1.getdate();
                cout<<"\n Add Number of days:";
                cin>>day;

                d1+(day);
                cout<<"\n\n Date After Adding
"<<day<<"into date ";
                cout<<"\n-----
---";
                d1.display();
                break;

    case 2: d1.getdate();
                cout<<"\n Subtract Number of days:";
                cin>>day;

                d1-(day);
                cout<<"\n\n Date After Subtracting
"<<day<<"into date ";
                cout<<"\n-----
---";
                d1.display();
                break;
}
```

```
case 3: d1.getdate();
          d2.getdate();

          res=d1>d2;
          if(res==1)
          {
              cout<<"Date1 is greter.";
          }
          else
          {
              cout<<"Date1 is not greter.";
          }
          break;

case 4: d1.getdate();
          d2.getdate();

          res=d1>=d2;
          if(res==1)
          {
              cout<<"Date1 is greter.";
          }
          else
          {
              cout<<"Date1 is not greter.";
          }
```

```
break;

case 5: d1.getdate();
          d2.getdate();

          res=d1<d2;
          if(res==1)
          {
              cout<<"Date2 is greter.";
          }
          else
          {
              cout<<"Date2 is not greter.";
          }
          break;

case 6: d1.getdate();
          d2.getdate();

          res=d1<=d2;
          if(res==1)
          {
              cout<<"Date is greter.";
          }
          else
          {
```

```
        cout<<"Date is not greater.";  
    }  
    break;  
  
case 7: d1.getdate();  
    d1++;  
    d1.display();  
    break;  
  
case 8: d1.getdate();  
    d1--;  
    d1.display();  
    break;  
  
}  
  
cout<<"\n Do You Want to continue:";  
cin>>ch2;  
  
}while(ch2=='y' || ch2=='Y');  
  
getch();  
}
```

=====

Output

-
-
- 1.Addition(+) of Days
 - 2.Subtraction(-) of Days
 - 3.chack two date (>)
 - 4.chack two date (>=)
 - 5.chack two date (<)
 - 6.chack two date (<=)
 - 7.Increment date (++)
 - 8.Decrement date (--)
-

Enter Your choice:1

Enter Date:5 2 2019

Add Number of days:375

Date After Adding 375into date

Date is:20/2/2020

Do You Want to continue:y

Enter Your choice:2

Enter Date:5 1 2019

Subtract Number of days:25

Date After Subtracting 25into date

Date is:10/12/2018

Do You Want to continue:y

Enter Your choice:3

Enter Date:5 2 2018

Enter Date:1 2 2018

Date1 is greater.

Do You Want to continue:y

Enter Your choice:4

Enter Date: 4 5 2019

Enter Date:4 5 2019

Date1 is greater.

Do You Want to continue:y

Enter Your choice:5

Enter Date:5 4 2019

Enter Date:1 2 2018

Date2 is not greater.

Do You Want to continue:y

Enter Your choice:7

Enter Date:30 12 2019

Date is:1/1/2020

Do You Want to continue:y

Enter Your choice:8

Enter Date:1 1 2019

Date is:30/12/2018

Do You Want to continue:n

// Q6: Date with class Ends

//

=====

//

=====

// Q7: Time with class Starts

```
#include<iostream>
#include<string>
#include<conio.h>
using namespace std;

class Time
{
    int hh,mm,ss;
public:
    void getTime()
    {
        cout<<"Enter Hours:";
        cin>>hh;
        cout<<"Enter Minute:";
        cin>>mm;
        cout<<"Enter Secound:";
        cin>>ss;
    }

    void display()
    {
        cout<<"\n";
        cout<<"\n"<<hh<<":"<<mm<<":"<<ss;
```

}

void operator +(int tmin)

{

mm+=tmin;

while(mm>=60)

{

mm=mm-60;

hh++;

if(hh>24)

{

hh=1;

}

}

if(hh>24)

{

hh=1;

}

}

void operator -(int tmin)

{

mm=mm-tmin;

while(mm<0)

{

hh--;

if(hh==0)

{

hh=24;

}

mm=mm+60;

}

}

int operator >(Time t2)

{

int flag=0;

if(hh>t2.hh)

{

flag=1;

```
    }
else if(hh==t2.hh)
{
    if(mm>t2.mm)
    {
        flag=1;
    }
else if(mm==t2.mm)
{
    if(ss>t2.ss)
    {
        flag=1;
    }
else
{
    flag=0;
}
}
return flag;
}

int operator <(Time t2)
{
    int flag=0;
```

```
if(hh<t2.hh)
{
    flag=1;

}

else if(hh==t2.hh)
{
    if(mm<t2.mm)
    {
        flag=1;
    }

    else if(mm==t2.mm)
    {
        if(ss<t2.ss)
        {
            flag=1;
        }

        else
        {
            flag=0;
        }
    }
}

return flag;
}
```

```
int operator >=(Time t2)
{
    int flag=0;

    if(hh>t2.hh)
    {
        flag=1;
    }

    else if(hh==t2.hh)
    {
        if(mm>t2.mm)
        {
            flag=1;
        }

        else if(mm==t2.mm)
        {
            if(ss>=t2.ss)
            {
                flag=1;
            }

            else
            {
                flag=0;
            }
        }
    }
}
```

```
    }

    return flag;
}

int operator <=(Time t2)
{
    int flag=0;

    if(hh<t2.hh)
    {
        flag=1;

    }

    else if(hh==t2.hh)
    {
        if(mm<t2.mm)
        {
            flag=1;

        }

        else if(mm==t2.mm)
        {
            if(ss<=t2.ss)
            {
                flag=1;

            }

        }

    }

}
```

```
    else
    {
        flag=0;
    }
}

return flag;
}
```

```
void operator ++()
{
    if(mm==60)
    {
        mm=1;
        hh++;      }
    else
    {
        mm++;
    }
}
```

```
void operator --()
```

```
{  
  
    if(mm==0)  
    {  
        mm=60;  
        hh--;  
    }  
  
    else  
    {  
        mm--;  
    }  
  
}  
  
};  
  
void main()  
{  
    int min,res,flag=0,ch;  
    char ch2='n';  
  
    Time t1,t2;  
  
    cout<<"\n-----";  
    cout<<"\n 1.Addition(+) of Days";
```

```
cout<<"\n 2.Subtraction(-) of Days";
cout<<"\n 3.chack two date (>)";
cout<<"\n 4.chack two date (>=)";
cout<<"\n 5.chack two date (<)";
cout<<"\n 6.chack two date (<=)";
cout<<"\n 7.Increment date (++)";
cout<<"\n 8.Decrement date (--)";
cout<<"\n-----";
```

```
do
```

```
{
```

```
cout<<"\n\n Enter Your choice:";
cin>>ch;
```

```
switch(ch)
```

```
{
```

```
    case 1: t1.getTime();
```

```
        cout<<"\n Add Number of min:";
        cin>>min;
```

```
        t1+(min);
```

```
        cout<<"\n\n Time After Adding
"<<min<<"into time ";
```

```
        cout<<"\n-----
---";
```

```
        t1.display();
```

```
        break;

case 2: t1.getTime();
          cout<<"\n Subtract Number of min:";
          cin>>min;

          t1-(min);
          cout<<"\n\n Time After Adding
"<<min<<"into time ";
          cout<<"\n-----
---";
          t1.display();
          break;

case 3: t1.getTime();
          t2.getTime();

          res=t1>t2;
          if(res==1)
          {
              cout<<"Time1 is greter.";
          }
          else
          {
              cout<<"Time1 is not greter.";
          }
          break;
```

```
case 4: t1.getTime();
          t2.getTime();

          res=t1>=t2;

          if(res==1)
          {
              cout<<"Time1 is greter.";
          }
          else
          {
              cout<<"Time1 is not greter.";
          }
          break;

case 5: t1.getTime();
          t2.getTime();

          res=t1<t2;
          if(res==1)
          {
              cout<<"Time2 is greter.";
          }
          else
          {
```

```
        cout<<"Time2 is not greter.";  
    }  
    break;  
  
case 6: t1.getTime();  
        t2.getTime();  
  
        res=t1<=t2;  
        if(res==1)  
        {  
            cout<<"Time2 is greter.";  
        }  
        else  
        {  
            cout<<"Time2 is not greter.";  
        }  
    break;  
  
case 7: t1.getTime();  
        t1++;  
        t1.display();  
    break;  
  
case 8: t1.getTime();  
        t1--;  
        t1.display();
```

```
        break;  
  
    }  
  
    cout<<"\n Do You Want to continue:";  
    cin>>ch2;  
  
}while(ch2=='y' || ch2=='Y');
```

```
getch();  
}  
=====
```

OUTPUT

```
=====
```

- ```

1.Addition(+) of Days
2.Subtraction(-) of Days
3.chack two date (>)
4.chack two date (>=)
5.chack two date (<)
6.chack two date (<=)
7.Increment date (++)
8.Decrement date (--)
```

---

Enter Your choice:1

Enter Hours:12

Enter Minute:45

Enter Secound:10

Add Number of min:110

Time After Adding 110into time

---

14:35:10

Do You Want to continue:y

Enter Your choice:2

Enter Hours:12

Enter Minute:00

Enter Secound:10

Subtract Number of min:60

Time After Adding 60into time

---

11:0:10

Do You Want to continue:y

Enter Your choice:3

Enter Hours:10

Enter Minute:45

Enter Secound:07

Enter Hours:10

Enter Minute:45

Enter Secound:07

Time1 is not greter.

Do You Want to continue:y

Enter Your choice:4

Enter Hours:23

Enter Minute:21

Enter Secound:12

Enter Hours:23

Enter Minute:21

Enter Secound:12

Time1 is greter.

Do You Want to continue:y

Enter Your choice:5

Enter Hours:10

Enter Minute:10

Enter Secound:10

Enter Hours:10

Enter Minute:9

Enter Secound:45

Time2 is not greter.

Do You Want to continue:6

Enter Your choice:6

Enter Hours:14

Enter Minute:24

Enter Secound:60

Enter Hours:15

Enter Minute:10

Enter Secound:14

Time2 is greter.

Do You Want to continue:y

Enter Your choice:7

Enter Hours:25

Enter Minute:56

Enter Secound:50

25:57:50

Do You Want to continue:y

Enter Your choice:8

Enter Hours:60

Enter Minute:0

Enter Secound:45

59:60:45

Do You Want to continue:n

```
// Q7: Time Ends
//
=====
```

```
//
=====
```

// Q8 Starts

```
/*
```

Q8) Write a menu driven program that can perform the following functions on strings. (Use overloaded operators where possible).(Do not use predefined string class)

1. Compare two strings for equality (== operator)
2. Check whether first string is smaller than the second (<= operator)
3. Copy the string to another
4. Extract a character from the string (Overload [])

5. Reverse the string

6. Concatenate two strings (+ operator)

\*/

```
#include<iostream>
#include<conio.h>
using namespace std;

class String
{
 // string str;
 char str[20];
public:
 void getdata()
 {
 cout<<"\n Enter String:";
 cin.get(str,20);
 }

 void display()
 {
 int i=0;
 cout<<"\n";
 while(str[i]!='\0')
 {
```

```
 cout<<str[i];
 i++;
}

}

bool operator==(String s2)
{
 int i=0,j=0,flag=1;

 while(str[i]!='\0' && s2.str[j]!='\0' &&
str[i]==s2.str[j])
 {
 i++;
 j++;
 }

 if(str[i]=='\0' && s2.str[j]=='\0')
 {
 return true;
 }
 else
 {
 return false;
 }
}
```

```
}
```

```
bool operator <=(String s2)
{
 int i=0,j=0,flag=1,res;
 cout<<str;
 while(str[i]!='\0' && s2.str[j]!='\0' &&
str[i]==s2.str[j])
 {
 i++;
 j++;
 }
 if(str[i]=='\0' && s2.str[j]=='\0')
 {
 return false;
 }
 else
 {
 res=str[i]-s2.str[j];
 if(res<0)
 return true;
 else
 return false;
 }
}
```

```
void operator =(String s2)
{
 int i=0;

 while(s2.str[i]!='\0')
 {
 str[i]=s2.str[i];
 i++;
 }
 str[i]='\0';

}

char operator [](int index)
{
 return str[index];
}

String operator +(String s2)
{
 String str3;
 int i=0,j=0;

 while(str[i]!='\0')
 {
 str3.str[i]=str[i];
 }
}
```

```
i++;
}
while(s2.str[j]!='\0')
{
 str3.str[i]=s2.str[j];
 i++;
 j++;
}
str3.str[i]='\0';
return str3;
}
```

```
int length(char str[20])
```

```
{
 int len=0,i=0;
 while(str[i]!='\0')
 {
 i++;
 len++;
 }
}
```

```
return len;
```

```
}
```

```
void reverse()
```

```
{
```

```
int i=0,j=0,len,temp;
len=length(str);

i=0;
j=len-1;
while(i!=(len/2))
{
 temp=str[i];
 str[i]=str[j];
 str[j]=temp;
 i++;
 j--;
}

};

void main()
{
 char ch,cont='n';
 String s1,s2,s3;
 bool res;
 int index,choice;
 s1.getdata();
 flushall();
```

```
s2.getdata();
cout<<"\n-----";
cout<<"\n 1. Compare two strings for equality";
cout<<"\n 2. Check whether first string is smaller than
the second";
cout<<"\n 3. Copy the string to another";
cout<<"\n 4. Extract a character from the string";
cout<<"\n 5. Reverse the string";
cout<<"\n 6. Concatenate two strings";
cout<<"\n-----";
do
{
 cout<<"\n Enter Your Choice:";
 cin>>choice;

switch(choice)
{
 case 1:
 if(s1==s2)
 cout<<"\n Both Strings Are Equal.";
 else
 cout<<"\n Both Strings Are Not
Equal.";
 break;
 case 2:
 if(s1<=s2)
```

```
 cout<<"\n First String is Smaller than
Secound.";
```

```
else
```

```
 cout<<"\n First String is Not Smaller
than Secound.";
```

```
break;
```

```
case 3:
```

```
s2=s1;
```

```
cout<<"\n-----";
```

```
cout<<"\n Frist String";
```

```
cout<<"\n-----\n";
```

```
s1.display();
```

```
cout<<"\n-----";
```

```
cout<<"\n Secound String";
```

```
cout<<"\n-----\n";
```

```
s2.display();
```

```
break;
```

```
case 4:
```

```
s1.reverse();
```

```
cout<<"\n-----";
```

```
cout<<"\n Reverse String";
```

```
cout<<"\n-----\n";
```

```
s1.display();
```

```
break;
```

```
case 5:
```

```
 cout<<"\n Enter index:";
 cin>>index;
 ch=s1[index];
 cout<<"character is:"<<ch;
 break;

 case 6:
 s3=s1+s2;
 cout<<"\n -----";
 cout<<"\n String:";
 cout<<"\n -----"\n";
 s3.display();
 break;
 }
 cout<<"\n\n Do You Want to Continue(y/n)";
 cin>>cont;
}while(cont=='Y' || cont=='y');

getch();
}

/*
```

Enter String:Rollwala

Enter String:DCSGU

- 
1. Compare two strings for equality
  2. Check whether first string is smaller than the second
  3. Copy the string to another
  4. Extract a character from the string
  5. Reverse the string
  6. Concatenate two strings
- 

Enter Your Choice:1

Both Strings Are Not Equal.

Do You Want to Continue(y/n)y

Enter Your Choice:2

Rollwala

First String is Not Smaller than Secound.

Do You Want to Continue(y/n)y

Enter Your Choice:3

---

Frist String

---

Rollwala

-----

Secound String

-----

Rollwala

Do You Want to Continue(y/n)y

Enter Your Choice:4

-----

Reverse String

-----

alawlloR

Do You Want to Continue(y/n)y

Enter Your Choice:5

Enter index:2

character is:a

Do You Want to Continue(y/n)y

Enter Your Choice:6

-----

String:

-----

alawlloRRollwala

Do You Want to Continue(y/n)

\*/

// Q8: Ends

//

=====

=====

```
//=====
=====

// Q9: Use Manipulators Starts

#include<stdio.h>
#include<conio.h>
#include<iostream>
#include<iomanip>
#include<string>
using namespace std;

ostream & convert(ostream & pout)
{
 pout<<setw(5)<<setprecision(5)<<setfill('+')<<setiosflags(ios::right);
 return pout;
}

void main()
{
 float amt;
 cout<<"\n Enter Amount:";
```

```
 cin>>amt;
 cout<<convert;
 cout<<amt;

 getch();
}
```

```
/*
```

Enter Amount:1.2

++1.2

```
*/
```

```
// Q9: Use manipulators Ends
```

```
//
```

```
=====
```

```
=====
```

\*\*\*\*\*

\*\*\*

\*\*\*\*\*

\*\*\*

\*\*\*\*\*

\*\*\*

\*\*\*\*\*

\*\*\*

\*\*\*\*\*

\*\*\*

**ASSIGNMENT: 3**

**ASSIGNMENT: 3**

**ASSIGNMENT: 3**

\*\*\*\*\*

\*\*\*

\*\*\*\*\*

\*\*\*

\*\*\*\*\*

\*\*\*

\*\*\*\*\*

\*\*\*

\*\*\*\*\*

\*\*\*

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :1) Write a program to generate templates function  
for swapping values of

variables and show its use with integer, float and  
character type of data

as input.

=====

=====

=====

```
#include<iostream>
```

```
#include<conio.h>
```

```
using namespace std;
```

```
template <typename T>
```

```
void Swap(T *a,T *b)
```

```
{
```

```
 T temp;
```

```
 temp=*a;
```

```
 *a=*b;
```

```
 *b=temp;
```

```
}
```

```
void main()
{
 int a,b;
 float c,d;
 char ch1,ch2;

 cout<<"\n Enter A:";
 cin>>a;
 cout<<"\n Enter B:";
 cin>>b;
 cout<<"\n a is:"<<a<<"\n b is:"<<b;
 cout<<"\n-----";
 cout<<"\n After Swap";
 cout<<"\n-----";
 Swap<int>(&a,&b);
 cout<<"\n a is:"<<a<<"\n b is:"<<b;

 cout<<"\n\n Enter C:";
 cin>>c;
 cout<<"\n Enter D:";
 cin>>d;
 cout<<"\n C is:"<<c<<"\n D is:"<<d;
 cout<<"\n-----";
```

```
cout<<"\n After Swap";
cout<<"\n-----";
Swap<float>(&c,&d);
cout<<"\n C is:<<c<<"\n D is:"<<d;

cout<<"\n\n Enter character1:";
cin>>ch1;
cout<<"\n Enter character2:";
cin>>ch2;
cout<<"\n ch1 is:"<<ch1<<"\n ch2 is:"<<ch2;
cout<<"\n-----";
cout<<"\n After Swap";
cout<<"\n-----";
Swap<char>(&ch1,&ch2);
cout<<"\n ch1 is:"<<ch1<<"\n ch2 is:"<<ch2;

getch();
}

=====
```

OUTPUT  
=====

Enter A:12

Enter B:50

a is:12

b is:50

-----

After Swap

-----

a is:50

b is:12

Enter C:45.50

Enter D:32.5

C is:45.5

D is:32.5

-----

After Swap

-----

C is:32.5

D is:45.5

Enter character1:A

Enter character2:B

ch1 is:A

ch2 is:B

---

After Swap

---

ch1 is:B

ch2 is:A

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

---

---

---

Definition :10) Create a class student from which the classes test and sports are derived.

The class student has the name and rollno of the student. The class test has

the marks of the internal test and the sports class has the marks of the

sports test. The class student contains a virtual function display() which

are implemented in the classes test and sports. Write a program which will

take relative information and display it using pointer of the base class.

```
=====
=====
=====
```

```
#include<iostream>
#include<conio.h>
#include<string>
using namespace std;
```

```
class Student
```

```
{
```

```
protected:
```

```
 int rno;
```

```
 string name;
```

```
public :
```

```
 void add()
```

```
{
```

```
 cout<<"\n Enter rollno:";
```

```
 cin>>rno;
 cout<<"\n Enter name:";
 cin>>name;
}
virtual void display()
{
 cout<<"\n rollno is:"<<rno;
 cout<<"\n name is:"<<name;
}
};
```

```
class Test : public Student
{
protected:
 int internal_marks[5];
public:
 void add_marks()
 {
 for(int i=0;i<3;i++)
 {
 cout<<"\n Enter Internal marks:";
 cin>>internal_marks[i];
 }
 }
 void display()
{
```

```
for(int i=0;i<3;i++)
{
 cout<<"\n Internal marks
is:"<<internal_marks[i];
}

};

class Sports : public Student
{
protected:
 int sports_mark;
public:
 void add_sports_marks()
 {
 cout<<"\n Enter sports marks:";
 cin>>sports_mark;

 }
 void display()
 {
 cout<<"\n sports marks is:"<<sports_mark;

 }
};
```

```
void main()
{
 Student *s,s1;
 Test t1;
 Sports sp1;

 s=&s1;
 s->add();
 t1.add_marks();
 sp1.add_sports_marks();
 s->display();

 s=&t1;
 s->display();

 s=&sp1;
 s->display();

 getch();
}
```

=====

OUTPUT

=====

Enter rollno:1

Enter name:priya

Enter Internal marks:45

Enter Internal marks:89

Enter Internal marks:78

Enter sports marks:5

rollno is:1

name is:priya

Internal marks is:45

Internal marks is:89

Internal marks is:78

sports marks is:56

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :11) Write a Program to perform following operation  
on text file :

11.1) write content in a text file 11.2) read content  
from file

11.3) count no of word and no of lines in a file 11.4)  
copy contents of one

file to another file

```
=====
=====
=====

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

using namespace std;

class File
{
 char inputline[80],outputline[80];
public:
 void write_file(string file)
 {
 ofstream Entryfile(file);

 while(true)
 {
 cin.getline(inputline,80);
 if(!strcmp(inputline, "End"))
 break;
 Entryfile<<"\n"<<inputline;
 }
 }
}
```

```
 Entryfile.close();
}

void read_file(string file)
{

 ifstream Displayfile(file);
 while(!Displayfile.eof())
 {
 Displayfile.getline(outputline,80);
 cout<<"\n"<<outputline;
 }

 Displayfile.close();
}

void copy_file(string destination,string source)
{
 ofstream Entryfile(destination);
 ifstream Displayfile(source);
 while(!Displayfile.eof())
 {
 Displayfile.getline(outputline,80);
 Entryfile<<outputline<<endl;
 }

 Displayfile.close();
}
```

```
Entryfile.close();
cout<<"\n file Copied Successfully./";

}

void count_word_lines(string file)
{
 char ch;
 int lines=0,word=0;

 ifstream Displayfile(file);
 while(!Displayfile.eof())
 {
 Displayfile.unsetf(ios::skipws);
 Displayfile>>ch;

 if(ch ==' ')
 {
 word++;
 }
 if(ch=='\n')
 {
 lines++;
 word++;
 }
 }
}
```

```
 cout<<"\n Lines is:"<<lines;
 cout<<"\n Word is:"<<word;
}

};

void main()
{
 File fp;
 int ch;
 char choice='n';
 string file,source,destination;

 cout<<"\n -----";
 cout<<"\n 1.write content in a text file ";
 cout<<"\n 2.read content from file ";
 cout<<"\n 3.count no of word and no of lines in a file";
 cout<<"\n 4.copy contents of one file to another file";
 cout<<"\n-----";
 do
 {
 cout<<"\n Enter Your Choice:";
 cin>>ch;
```

```
switch(ch)
{
 case 1:
 cout<<"\n Enter Filename:";
 cin>>file;
 fp.write_file(file);
 break;

 case 2:
 cout<<"\n Enter Filename:";
 cin>>file;
 fp.read_file("data.txt");
 break;

 case 3:
 cout<<"\n Enter Filename:";
 cin>>file;
 fp.count_word_lines("data.txt");
 break;

 case 4:
 cout<<"\n Enter destination Filename:";
 cin>>destination;
 cout<<"\n Enter source Filename:";
 cin>>source;
 fp.copy_file("details.txt","data.txt");
 break;
}
```

```
cout<<"\n Do You Want to continue:";
cin>>choice;
}while(choice=='y'|| choice=='Y');

getch();
}
```

=====

## OUTPUT

=====

- 
- 1.write content in a text file
  - 2.read content from file
  - 3.count no of word and no of lines in a file
  - 4.copy contents of one file to another file
- 

Enter Your Choice:1

Enter Filename:data.txt

user is dump

hello world

End

Do You Want to continue:y

Enter Your Choice:2

Enter Filename:data.txt

user is dump

hello world

Do You Want to continue:y

Enter Your Choice:3

Enter Filename:data.txt

Lines is:2

Word is:5

Do You Want to continue:y

Enter Your Choice:4

Enter destination Filename:details.txt

Enter source Filename:data.txt

file Copied Successfully.

Do You Want to continue:y

Enter Your Choice:2

Enter Filename:details.txt

user is dump

hello world

Do You Want to continue:n

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

```
=====
=====
=====
```

Definition :12) Write a program to create a file student to store name and marks of 5 students and then display them.

```
=====
=====
=====
```

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

using namespace std;

class Student
{
 int rno,marks[3];
 string name;

public:
 void getdata()
 {
 cout<<"\n Enter Roll no:";
 cin>>rno;
 cout<<"\n Enter name:";
 cin>>name;
```

```
for(int i=0;i<3;i++)
{
 cout<<"\n Enter marks:";
 cin>>marks[i];
}

void display()
{
 cout<<"\n Roll no is:"<<rno;
 cout<<"\n Name is:"<<name;

 for(int i=0;i<3;i++)
 {
 cout<<"\n marks of sub "<<i+1<<
is:"<<marks[i];
 }
}

void main()
{
 Student obj;
 char Continue = 'n';

 //write data into file
 ofstream enterfile;
```

```
enterfile.open("student.txt",ios::out || ios::binary ||
ios::trunc);

if(!enterfile.is_open()){
 cout<<"Unable to open a file.";
}

else
{

 cout<<"\n Input Data:";
 do
 {

 obj.getdata();
 enterfile.write((char *)&obj,sizeof(obj));
 if(enterfile.fail())
 {
 cout<<"\n File write failed."
 }

 cout<<"\n Do you want to continue(y/n):";
 cin>>Continue;
 }while(Continue=='y');
}

enterfile.close();

//display data from file

ifstream display("student.txt",ios::in || ios::binary);
```

```
cout<<"\n Output:\n";
while(!display.eof())
{
 display.read((char *)&obj,sizeof(obj));
 if(display.fail())
 break;
 obj.display();
}

display.close();
getch();
}
```

=====

OUTPUT

=====

Input Data:

Enter Roll no:1

Enter name:sivangi

Enter marks:89

Enter marks:78

Enter marks:56

Do you want to continue:(y/n)y

Enter Roll no:2

Enter name:jinal

Enter marks:45

Enter marks:56

Enter marks:78

Do you want to continue(y/n):n

Output:

Roll no is:1

Name is:sivangi

marks of sub 1 is:89

marks of sub 2 is:78

marks of sub 3 is:56

Roll no is:2

Name is:jinal  
marks of sub 1 is:45  
marks of sub 2 is:56  
marks of sub 3 is:78

Roll No :30  
Name :Ajinkya Rathod  
Class :MCA-2  
Subject :Object Oriented Concepts & Programming

=====

=====

=====

=====

=====

Definition :13) Define a class Result which contains the result of an MCA II written

test. It should take list from a file and display on the screen such that

at a time only ten candidates information is printed on the screen.

```
=====
=====
=====
```

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

using namespace std;

class Result
{
 float per;
 int rno,marks[3],sum;
 string name;

public:
 void getdata()
 {
 sum=0;
 cout<<"\n Enter Roll no:";
 cin>>rno;
 cout<<"\n Enter name:";
 cin>>name;

 for(int i=0;i<3;i++)
 {
```

```
 cout<<"\n Enter marks:";
 cin>>marks[i];
 sum=sum+marks[i];
 }
 per=sum/3;
}

void result()
{
 cout<<"\n"<<rno<<"\t"<<name<<"\t"<<per;
}
};

void main()
{
 int cnt=0;
 Result obj;
 char Continue = 'n';

 //write data into file
 ofstream enterfile;
 enterfile.open("student.txt",ios::out || ios::binary ||
 ios::trunc);
 if(!enterfile.is_open())
 {
 cout<<"Unable to open a file.";
 }
 else
```

```
{
 cout<<"\n Input Data:";
 do
 {

 obj.getdata();
 enterfile.write((char *)&obj,sizeof(obj));
 if(enterfile.fail())
 {
 cout<<"\n File write failed.";
 }

 cout<<"\n Do you want to continue:(y/n)";
 cin>>Continue;
 }while(Continue=='y');
}
enterfile.close();

//display data from file

ifstream display("student.txt",ios::in || ios::binary);
cout<<"\n Output:\n";
cout<<"\n rno \t name \t per";
while(!display.eof() && cnt<10)
{
 cnt++;
```

```
 display.read((char *)&obj,sizeof(obj));
 if(display.fail())
 break;
 obj.result();
}

=====
OUTPUT
```

=====

Input Data:

Enter Roll no:1

Enter name:flora

Enter marks:45

Enter marks:89

Enter marks:78

Do you want to continue:(y/n)y

Enter Roll no:2

Enter name:heena

Enter marks:45

Enter marks:56

Enter marks:2

Do you want to continue:(y/n)

y

Enter Roll no:3

Enter name:mira

Enter marks:47

Enter marks:89

Enter marks:66

Do you want to continue:(y/n)y

Enter Roll no:4

Enter name:rajvi

Enter marks:45

Enter marks:78

Enter marks:89

Do you want to continue:(y/n)y

Enter Roll no:5

Enter name:meet

Enter marks:45

Enter marks:12

Enter marks:23

Do you want to continue:(y/n)y

Enter Roll no:6

Enter name:jinali

Enter marks:45

Enter marks:55

Enter marks:55

Do you want to continue:(y/n)y

Enter Roll no:7

Enter name:aditi

Enter marks:80

Enter marks:89

Enter marks:87

Do you want to continue:(y/n)y

Enter Roll no:8

Enter name:preet

Enter marks:56

Enter marks:65

Enter marks:54

Do you want to continue:(y/n)y

Enter Roll no:9

Enter name:uttam

Enter marks:78

Enter marks:89

Enter marks:66

Do you want to continue:(y/n)y

Enter Roll no:10

Enter name:janvi

Enter marks:78

Enter marks:88

Enter marks:87

Do you want to continue:(y/n)y

Enter Roll no:11

Enter name:kiraa

Enter marks:45

Enter marks:56

Enter marks:78

Do you want to continue:(y/n)n

Output:

| rno | name  | per |
|-----|-------|-----|
| 1   | flora | 70  |
| 2   | heena | 34  |
| 3   | mira  | 67  |
| 4   | rajvi | 70  |
| 5   | meet  | 26  |

6 jinali 51  
7 aditi 85  
8 preet 58  
9 uttam 77  
10 janvi 84

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :14) Use an Employee Class to write records of employee to a file. Include

a menu that will allow the user to select any of the following features

- a. Add a new record.
  - b. Modify an existing record.
  - c. Retrieve and display an entire record for a given name.
  - d. Generate a complete list of all names, addresses and telephone numbers.
  - e. End of the computation.
- =====
- =====
- =====

```
#include<iostream>
#include<string>
#include<fstream>
using namespace std;
class emp{
 int id;
 string name,address;
public:
 void getdata(){
 cout<<"Enter the emp id: ";
 cin>>id;
 cout<<"Enter the name: ";
 cin.ignore();
 getline(cin,name);
 cout<<"Enter the Address:";
```

```
 cin.ignore();
 getline(cin,address);
 }

void show(){
 cout<<"id: "<<id<<endl;
 cout<<"Name: "<<name<<endl;
 cout<<"Address: "<<address<<endl;
}

int search(int x){
 if(x==id)
 return 1;
 else
 return 0;
}

void update(string var_change,int choice){
 if(choice==1)
 name=var_change;
 else if(choice==2)
 address=var_change;
}

};

void main(){
 int choice,flag=0,temp_id,position,endposition,n,choice2;
 char var_continue='y';
 string var_change;
```

```
emp obj;
fstream file;
do{
 cout<<"1. Add\n2. Display\n3. Update\n4. Exit\n";
 cin>>choice;
 switch(choice){
 case 1:
 file.open("emp.txt",ios::in | ios::out |
ios::binary | ios::trunc);
 do{
 obj.getdata();
 file.write((char*) &obj,sizeof(obj));
 if(file.fail())
 cout<<"File write failed";
 cout<<"Do you want continue?
(y/n): ";
 cin>>var_continue;
 }while(var_continue!='n');
 file.close();
 break;
 case 2:
 file.open("emp.txt",ios::in | ios::out |
ios::binary);
 cout<<"1. Display Full List\n2. Display
Particular Detail\n";
 cin>>choice2;
 switch(choice2){
 case 1:
```

```
 file.seekg(ios::beg);
 while(file.read((char*)
&obj,sizeof(obj))){
 obj.show();
 }
 break;
 case 2:
 file.seekg(0,ios::end);
 endposition=file.tellg();
 n=endposition/sizeof(emp);
 cout<<"Enter emp id: ";
 cin>>n;
 position=(n-1)*sizeof(emp);
 file.seekg(position);
 file.read((char*)
&obj,sizeof(obj));
 obj.show();
 break;
 default:
 cout<<"Enter correct choice\n";
 }
 file.close();
 break;
case 3:
 file.open("emp.txt",ios::in | ios::out |
ios::binary);
 file.seekg(0,ios::end);
```

```
endposition=file.tellg();
n=endposition/sizeof(emp);
cout<<"Enter emp id: ";
cin>>n;
position=(n-1)*sizeof(emp);
file.seekg(position);
file.seekp(0);
file.seekp(position);
cout<<"Which Field you want to
change\n1. Name\n2. Address\n";
cin>>choice2;
switch(choice2){
 case 1:
 cout<<"Enter name: ";
 cin>>var_change;
 obj.update(var_change,choice2);
 file.write((char*)
&obj,sizeof(obj));
 break;
 case 2:
 cout<<"Enter Address: ";
 cin>>var_change;
 obj.update(var_change,choice2);
 file.write((char*)
&obj,sizeof(obj));
 break;
 default:
```

```
 cout<<"Enter correct choice\n";
 }

 file.seekg(0);
 file.close();
 break;

case 4:
 exit(0);
default:
 cout<<"Enter correct choice\n";
}

}while(1);

}
```

=====

OUTPUT

=====

1. Add
2. Display
3. Update
4. Exit

1

Enter the emp id: 101

Enter the name: shivangi

Enter the Address:surat

Do you want continue? (y/n): y

Enter the emp id: 102

Enter the name: heena

Enter the Address:rajkot

Do you want continue? (y/n): n

1. Add

2. Display

3. Update

4. Exit

2

1. Display Full List

2. Display Particular Detail

1

id: 101

Name: shivangi

Address:surat

id: 102

Name: heena

Address:rajkot

1. Add

2. Display

3. Update

4. Exit

2

1. Display Full List

2. Display Particular Detail

2

Enter emp id: 102

id: 102

Name: heena

Address:rajkot

1. Add
2. Display
3. Update
4. Exit

3

Enter emp id: 101

Which Field you want to change

1. Name
2. Address

1

Enter name: janvi

1. Add
2. Display
3. Update
4. Exit

2

1. Display Full List

2. Display Particular Detail

1

id: 101

Name: shivangi

Address:surat

id: 102

Name: heena

Address:rajkot

1. Add
  2. Display
  3. Update
  4. Exit
- 4

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :15) Write a program that stores and displays the records of the customer

from a file the following information for account of the customer is to be

stored. Account no, account type, name, old balance, new balance,

last payment, date of last payment. Also display the current account

status by comparing current payment and previous balance. Also calculate

the current balance by subtracting the current payment from the previous

balance.

=====

=====

=====

```
#include<iostream>
#include<ctime>
#include<fstream>
#include<string>
#include<sstream>
using namespace std;
```

```
class account{
```

```
 int acc_num,new_balance,last_payment,num;
```

```
string acc_type,name,transaction_date,transaction_type;;
public:
account(){
 acc_num=0;
 new_balance=0;
 last_payment=0;
}
void getdata(){
 cout<<"Enter account number: ";
 cin>>acc_num;
 cout<<"Enter name of customer: ";
 cin>>name;
 cout<<"Enter account type: ";
 cin>>acc_type;
 cout<<"Enter the opening balance: ";
 cin>>new_balance;
}
void show(){
 cout<<"Account Number: "<<acc_num<<endl;
 cout<<"Name: "<<name<<endl;
 cout<<"Balance:
"<<new_balance<<endl<<endl;
}
void show_transaction(int x){
 if(x==num){
 cout<<"Account Numebr: "<<num<<endl;
```

```
 cout<<"Transaction Date:
"<<transaction_date<<endl;
 cout<<"Transaction type:
"<<transaction_type<<endl;
 cout<<"Amount: "<<last_payment<<endl;
 cout<<"Balance: "<<new_balance<<endl;
 }
}

int transaction(int x,string v_date){
 int temp,choice;
 cout<<"1. Deposite\n2. Withdraw\n";
 cin>>choice;
 switch(choice){
 case 1:
 cout<<"Enter the amount you want to
deposite: ";
 num=x;
 transaction_date=v_date;
 cin>>temp;
 new_balance=new_balance+temp;
 last_payment=temp;
 transaction_type="Credit";
 cout<<"Transaction completed
successfully\n";
 break;
 case 2:
 cout<<"Enter the amount you want to
withdraw: ";
```

```
 cin>>temp;
 if(temp<=new_balance){
 num=x;
 transaction_date=v_date;
 new_balance=new_balance-temp;
 last_payment=temp;
 transaction_type="Debit";
 cout<<"Transaction completed
successfully\n";
 }
 else
 cout<<"Transaction failed due to
Insufficient balance in account\n";
 break;
 default:
 cout<<"Please enter correct choice\n";
 }
 return new_balance;
}
void upd_balance(int x){

 new_balance=x;
}
};

void main(){
```

```
//date code;
string v_date,v_month,v_year;
stringstream sd,sm,sy;
int a;
time_t t=time(NULL);
tm* timeptr=localtime(&t);
a=(timeptr->tm_mday);
sd<<a;
sd>>v_date;
a=((timeptr->tm_mon)+1);
sm<<a;
sm>>v_month;
a=((timeptr->tm_year)+1900);
sy<<a;
sy>>v_year;
v_date=v_date+"/"+v_month+"/"+v_year;
//date code complete.
```

```
account obj;
fstream file, enterfile;
int
count=0,choice,flag=0,balance=0,position,endposition,n;
char var_continue='y';
do{
 cout<<"\n1. Add Customer\n2. Customer Details\n3.
Perform Transaction\n4. Transaction History\n5. Exit\n";
 cin>>choice;
```

```
switch(choice){
 case 1:
 file.open("Account.txt",ios::out |
ios::binary | ios::app);
 if(!file)
 cout<<"File does not exists\n";
 else{
 cout<<"Input data:\n";
 do{
 obj.getdata();
 file.write((char*)
&obj,sizeof(obj));
 if(file.fail())
 cout<<"File write failed";
 cout<<"Do you want continue?
(y/n): ";
 cin>>var_continue;
 }while(var_continue!='n');
 }
 file.close();
 break;
 case 2:
 file.open("Account.txt",ios::out | ios::in |
ios::binary);
 file.seekg(0,ios::end);
 endposition=file.tellg();
 n=endposition/sizeof(account);
 cout<<"Enter account number: ";
```

```
 cin>>n;
 position=(n-1)*sizeof(account);
 file.seekg(position);
 file.read((char*) &obj,sizeof(obj));
 if(file.fail())
 cout<<"Account number does not
exists\n";
 else
 obj.show();
 file.seekg(0);
 file.close();
 break;
case 3:
 file.open("Account.txt",ios::out | ios::in |
ios::binary);
 file.seekg(0,ios::end);
 endposition=file.tellg();
 n=endposition/sizeof(account);
 cout<<"Enter account number: ";
 cin>>n;
 position=(n-1)*sizeof(account);
 file.seekg(position);
 file.read((char*) &obj,sizeof(obj));
 if(file.fail())
 cout<<"Account number does not
exists\n";
 else{
```

```
 enterfile.open("Transaction.txt",
ios::out | ios::binary | ios::app);
 balance=obj.transaction(n,v_date);
 enterfile.write((char*)
&obj,sizeof(obj));
 enterfile.close();
 }
 file.seekp(0);
 file.seekp(position);
 obj.upd_balance(balance);
 file.write((char*) &obj,sizeof(obj));
 file.seekg(0);
 file.close();
 break;
case 4:
 cout<<"Enter account number: ";
 cin>>n;
 file.open("Transaction.txt",ios::in | ios::out
| ios::binary);
 file.seekg(ios::beg);
 while(file.read((char*) &obj,sizeof(obj))){
 obj.show_transaction(n);
 }
 file.close();
 break;
case 5:
 exit(0);
```

```
 default:
 cout<<"Enter correct choice\n";
 }
}while(1);
}
```

=====

## OUTPUT

=====

1. Add Customer
2. Customer Details
3. Perform Transaction
4. Transaction History
5. Exit

1

Input data:

Enter account number: 1

Enter name of customer: shivangi

Enter account type: current

Enter the opening balance: 450

Do you want continue? (y/n): y

Enter account number: 2

Enter name of customer: preet

Enter account type: saving

Enter the opening balance: 780

Do you want continue? (y/n): n

1. Add Customer
2. Customer Details
3. Perform Transaction
4. Transaction History
5. Exit

2

Enter account number: 1

Account Number: 101

Name: shivangi

Balance: 5000

1. Add Customer
2. Customer Details
3. Perform Transaction
4. Transaction History
5. Exit

2

Enter account number: 2

Account Number: 102

Name: preet

Balance: 8000

1. Add Customer
2. Customer Details
3. Perform Transaction
4. Transaction History
5. Exit

3

Enter account number: 1

1. Deposite
2. Withdraw

1

Enter the amount you want to deposite: 500

Transaction completed successfully

1. Add Customer
2. Customer Details
3. Perform Transaction
4. Transaction History
5. Exit

2

Enter account number: 1

Account Number: 101

Name: shivangi

Balance: 5500

1. Add Customer
2. Customer Details
3. Perform Transaction
4. Transaction History
5. Exit

4

Enter account number: 1

Account Numebr: 1

Transaction Date: 5/6/2020

Transaction type: Credit

Amount: 500

Balance: 5500

1. Add Customer
2. Customer Details
3. Perform Transaction
4. Transaction History
5. Exit

5

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :16) WAP to create namespace having function for  
total\_marks. Show its use

in class 'marks' of students, display total marks of  
subjects using namespace.

=====

=====

=====

```
#include<iostream>
```

```
#include<conio.h>
```

```
#include<string>
```

```
using namespace std;
```

```
namespace
```

```
{
```

```
 int total(int marks[],int n)
```

```
{
```

```
int sum=0;

for(int i=0;i<n;i++)
{
 sum=sum+marks[i];
}

return sum;
}
```

```
class Marks
{
 int rno,marks[5],sum;
public:
 void addmarks()
 {
 cout<<"\n Enter Roll no:";
 cin>>rno;
 for(int i=0;i<5;i++)
 {
 cout<<"\n Enter Marks:";
 cin>>marks[i];
 }
 }

 void display()
```

```
{
 cout<<"\n Roll no is:"<<rno;
 for(int i=0;i<5;i++)
 {
 cout<<"\n Marks is:"<<marks[i];
 }
 sum=total(marks,5);
 cout<<"\n sum is:"<<sum;
}

};

void main()
{
 Marks m1;
 m1.addmarks();
 m1.display();
 getch();
}
```

=====

OUTPUT

=====

Enter Roll no:1

Enter Marks:45

Enter Marks:56

Enter Marks:44

Enter Marks:52

Enter Marks:32

Roll no is:1

Marks is:45

Marks is:56

Marks is:44

Marks is:52

Marks is:32

sum is:229

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :17) WAP to define a vector and use it for student class to store and display

information about student (Use STL).

=====

=====

=====

```
#include <iostream>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
#include<conio.h>
```

```
using namespace std;
```

```
class Student
```

```
{
```

```
private:
```

```
 int RollNumber;
```

```
 float TotalMarks;
```

```
public:
 Student(){
 }
 Student(int TempRollNumber, float TempTotalMarks) {
 RollNumber = TempRollNumber;
 TotalMarks = TempTotalMarks;
 }
 void operator = (Student TempStud){
 RollNumber = TempStud.RollNumber;
 TotalMarks = TempStud.TotalMarks;
 }
 bool operator < (Student TempStud) {
 return(TotalMarks < TempStud.TotalMarks);
 }
 friend ostream & operator <<(ostream & TempOut,
 Student & TempStud);
};

ostream & operator <<(ostream & TempOut, Student &
 TempStud) {
 TempOut << "The mark of roll number " <<
 TempStud.RollNumber << " is " << TempStud.TotalMarks;
 return TempOut;
}

void main() {
 vector <Student> StudMarks;
 float TempMarks;
```

```
int i = 0;

for(;;)
{
 cout << "Enter the mark for roll number " << i + 1
 << " Enter -1 to stop: ";
 cin >> TempMarks;

 if(TempMarks == -1) break;
 StudMarks.push_back(Student(i + 1, TempMarks));
 ++i;
}

cout << "The size of StudMarks is " <<
StudMarks.size() << endl;

vector <Student>::iterator index;
sort(StudMarks.begin(), StudMarks.end());

for(index = StudMarks.begin(); index < StudMarks.end();
++index)
 cout << *index << endl;
//return 0;
getch();
}
```

=====

## OUTPUT

=====

Enter the mark for roll number 1 Enter -1 to stop: 45

Enter the mark for roll number 2 Enter -1 to stop: 55

Enter the mark for roll number 3 Enter -1 to stop: 89

Enter the mark for roll number 4 Enter -1 to stop: -1

The size of StudMarks is 3

The mark of roll number 1 is 45

The mark of roll number 2 is 55

The mark of roll number 3 is 89

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

```
=====
=====
=====
```

Definition :2) Write an object oriented program to implement a generic Stack. Incorporate

all the possible operation on Stack in the program.

```
=====
=====
=====
```

```
#include<iostream>
```

```
#include<conio.h>
```

```
using namespace std;
```

```
template <typename T>
```

```
class Stack
```

```
{
```

```
 T stack[5];
```

```
 int top,i,res;
```

```
public:
```

```
 Stack()
```

```
{
```

```
 top=-1;
```

```
}
```

```
 void Push(T a)
```

```
{
```

```
if(top>5)
{
 cout<<"\n Stack is Overflow.";
}
else
{
 top++;
 stack[top]=a;
}

void Pop()
{
 if(top== -1)
 {
 cout<<"\n Stack is Underflow or Empty.";
 }
 else
 {
 stack[top]=NULL;
 top--;
 cout<<"\n Deleted Successfully.";
 }
}

void Display()
{
```

```
if(top== -1)
{
 cout<<"\n Stack is empty.";
}
else
{
 for(i=0;i<=top;i++)
 {
 cout<<"\n val is:"<<stack[i];
 }
}
};

void main()
{
 Stack <int>s1;

 int ch,val,res;
 char choice='n';

 cout<<"\n-----";
 cout<<"\n 1.Push";
 cout<<"\n 2.Pop";
 cout<<"\n 3.Display";
 cout<<"\n-----";
```

```
do
{
 cout<<"\n Enter Your Choice:";
 cin>>ch;

 switch(ch)
 {
 case 1:
 cout<<"\n Enter Value:";
 cin>>val;
 s1.Push(val);
 break;
 case 2:
 s1.Pop();
 break;
 case 3:
 s1.Display();
 break;
 }
 cout<<"\n Do You Want to continue:";
 cin>>choice;

}while(choice=='Y' || choice=='y');
getch();
}
```

=====

## OUTPUT

=====

-----  
1.Push

2.Pop

3.Display

-----

Enter Your Choice:1

Enter Value:10

Do You Want to continue:y

Enter Your Choice:1

Enter Value:20

Do You Want to continue:y

Enter Your Choice:1

Enter Value:30

Do You Want to continue:y

Enter Your Choice:3

val is:10

val is:20

val is:30

Do You Want to continue:y

Enter Your Choice:2

Deleted Successfully.

Do You Want to continue:y

Enter Your Choice:3

val is:10

val is:20

Do You Want to continue:y

Enter Your Choice:2

Deleted Successfully.

Do You Want to continue:y

Enter Your Choice:2

Deleted Successfully.

Do You Want to continue:y

Enter Your Choice:3

Stack is empty.

Do You Want to continue:y

Enter Your Choice:2

Stack is Underflow or Empty.

Do You Want to continue:n

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :3) Write a generic function that will sort a character string, integer

and float value. Create a menu with appropriate options and accept the

values from the user

=====

=====

=====

```
#include<iostream>
```

```
#include<conio.h>
```

```
#include<string>
```

```
using namespace std;
```

```
template <typename T>
```

```
void Bubble_sort(T arr[])
```

```
{
```

```
 int i,j,n=5;
```

```
 T temp;
```

```
 for(i=0;i<n;i++)
```

```
{
```

```
 for(j=0;j<n-i-1;j++)
```

```
{
 if(arr[j]>arr[j+1])
 {
 temp=arr[j];
 arr[j]=arr[j+1];
 arr[j+1]=temp;
 }
}
}
cout<<"\n Sorted Completed.";
}
```

template <typename T>

```
void Display(T arr[])
{
 int i;
 cout<<"\n-----";
 for(i=0;i<5;i++)
 {
 cout<<"\n Value is:"<<arr[i];
 }
}
```

void main()

```
{
int arr[5],i;
float farr[5];
char carr[10];

int ch,val,res;
char choice='n';

cout<<"\n-----";
cout<<"\n 1.Int";
cout<<"\n 2.Float";
cout<<"\n 3.char";
cout<<"\n-----";

do
{
 cout<<"\n Enter Your Choice:";
 cin>>ch;

 switch(ch)
 {
 case 1:
 for(i=0;i<5;i++)
 {
 cout<<"\n Enter Value:";
```

```
 cin>>arr[i];
}

Bubble_sort<int>(arr);
Display<int>(arr);
break;

case 2:
 for(i=0;i<5;i++)
 {
 cout<<"\n Enter Value:";
 cin>>farr[i];
 }

Bubble_sort<float>(farr);
Display<float>(farr);
break;

case 3:
 cout<<"Enter String:";
 cin>>carr;

Bubble_sort<char>(carr);
Display<char>(carr);
break;

case 4:
 break;
```

```
 }
 cout<<"\n Do You Want to continue:";
 cin>>choice;

}while(choice=='Y' || choice=='y');
getch();
}
```

=====

## OUTPUT

=====

-----  
1.Int

2.Float

3.char

-----

Enter Your Choice:1

Enter Value:4

Enter Value:7

Enter Value:8

Enter Value:9

Enter Value:1

Sorted Completed.

---

Value is:1

Value is:4

Value is:7

Value is:8

Value is:9

Do You Want to continue:y

Enter Your Choice:2

Enter Value:4.5

Enter Value:6.32

Enter Value:0.12

Enter Value:2.12

Enter Value:4.5

Sorted Completed.

---

Value is:0.12

Value is:2.12

Value is:4.5

Value is:4.5

Value is:6.32

Do You Want to continue:y

Enter Your Choice:3

Enter String:hello

Sorted Completed.

---

Value is:e

Value is:h

Value is:l

Value is:l

Value is:o

Do You Want to continue:n

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :4) Write a template function called find(). This function searches an array

for an object. It returns either the index of the matching object (if one is

found) or -1 if no match is found.

=====

=====

=====

```
#include<iostream>
```

```
#include<string>
```

```
#include<conio.h>
```

```
using namespace std;
```

```
class Student
```

```
{
```

```
 public:
```

```
 int rno;
```

```
 string name;
public:
 void addstudent()
 {
 cout<<"\n Enter Roll no:";
 cin>>rno;
 cout<<"\n Enter Name:";
 cin>>name;
 }
 void display()
 {
 cout<<"\n roll no is:"<<rno;
 cout<<"\n Name is:"<<name;
 }
};
```

```
template <typename T>

int find(T arrayobj[], int rollno,int n)
{
 int flag=0,index=0;
 for(int i=0;i<n;i++)
 {
 if(arrayobj[i].rno==rollno)
 { flag=1;
```

```
 index=i;
 break;
 }
}

if(flag==0)
 return -1;
else
 return index;
}

void main()
{
 Student s1[20];
 int rollno,n,index;

 cout<<"\n Enter total value of student data:";
 cin>>n;
 for(int i=0;i<n;i++)
 {
 s1[i].addstudent();
 }

 cout<<"\n Total Student is:";

 cout<<"\n-----";
 for(int i=0;i<n;i++)
```

```
{
 s1[i].display();
}
cout<<"\n-----";

cout<<"\n Enter roll no which you want to find:";
cin>>rollno;

index=find<Student>(s1,rollno,n);
if(index== -1)
 cout<<"\n rollno is not found";
else
 cout<<"\n rollno is found";
getch();
}
```

=====

OUTPUT

=====

Enter total value of student data:2

Enter Roll no:1

Enter Name:sa

Enter Roll no:2

Enter Name:mina

Total Student is:

-----  
roll no is:1

Name is:sa

roll no is:2

Name is:mina

-----  
Enter roll no which you want to find:4

not found

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :5) WAP Implement template sort with a non type size.

=====

=====

=====

```
#include<iostream>
#include<conio.h>
#include<string>
using namespace std;
```

template <typename T>

void Bubble\_sort(T arr[],int n) //Non-Type Argument

{

    int i,j;

    T temp;

    for(i=0;i<n;i++)

    {

        for(j=0;j<n-i-1;j++)

        {

            if(arr[j]>arr[j+1])

```
 {
 temp=arr[j];
 arr[j]=arr[j+1];
 arr[j+1]=temp;
 }
 }
cout<<"\n Sorted Completed.";
}
```

```
template <typename T>
```

```
void Display(T arr[])
{
 int i;
 cout<<"\n-----";
 for(i=0;i<5;i++)
 {
 cout<<"\n Value is:"<<arr[i];
 }
}
```

```
void main()
{
 int arr[5],i;
```

```
float farr[5];
char carr[5];

int ch,val,res;
char choice='n';

cout<<"\n-----";
cout<<"\n 1.Int";
cout<<"\n 2.Float";
cout<<"\n 3.char";
cout<<"\n-----";

do
{
 cout<<"\n Enter Your Choice:";
 cin>>ch;

 switch(ch)
 {
 case 1:
 for(i=0;i<5;i++)
 {
 cout<<"\n Enter Value:";
 cin>>arr[i];
 }
 }
}
```

```
Bubble_sort<int>(arr,5);
Display<int>(arr);
break;

case 2:
for(i=0;i<5;i++)
{
 cout<<"\n Enter Value:";
 cin>>farr[i];
}

Bubble_sort<float>(farr,5);
Display<float>(farr);
break;

case 3:
cout<<"Enter String:";
cin>>carr;

Bubble_sort<char>(carr,5);
Display<char>(carr);
break;

case 4:
break;
}

cout<<"\n Do You Want to continue:";
```

```
 cin>>choice;

 }while(choice=='Y' || choice=='y');
 getch();
 }
```

=====

## OUTPUT

=====

-----  
1.Int

2.Float

3.char

-----

Enter Your Choice:1

Enter Value:4

Enter Value:5

Enter Value:2

Enter Value:7

Enter Value:1

Sorted Completed.

---

Value is:1

Value is:2

Value is:4

Value is:5

Value is:7

Do You Want to continue:y

Enter Your Choice:2

Enter Value:12.4

Enter Value:7.8

Enter Value:0.1

Enter Value:2.3

Enter Value:78.20

Sorted Completed.

---

Value is:0.1

Value is:2.3

Value is:7.8

Value is:12.4

Value is:78.2

Do You Want to continue:y

Enter Your Choice:3

Enter String:users

Sorted Completed.

---

Value is:e

Value is:r

Value is:s

Value is:s

Value is:u

Do You Want to continue:n

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :6) WAP to create base class Book having int id and char name as data members

and respective functionality, show following types of inheritance and display

the details of each kind of books, also calculate the total no of each type

of books in proper format. Simple inheritance with derived class Sales

Hierarchical inheritance with derived classes academics and thrillers Show use

of constructor and destructor in above examples of inheritance.

=====

=====

=====

```
#include<iostream>
```

```
#include<conio.h>
```

```
#include<string>
using namespace std;
ostream & line(ostream &);

class Book
{
protected:
 int id;
 char *name,*book_type;
 float price;

public:
 Book(int id,char *name,float price,char *book_type)
 {
 cout<<line<<"\n Book Class Constructor is call";
 this->id=id;
 this->name=name;
 this->book_type=book_type;
 this->price=price;
 }
 ~Book(){
 cout<<line<<"\n Book Class Destructor is call";
 delete []name;
 delete []book_type;
 }
}
```

```
};
```

```
class Sales : public Book
```

```
{
```

```
protected:
```

```
 int qty;
```

```
public:
```

```
 Sales(int id,char *name,float price,char *book_type,int
 qty):Book(id,name,price,book_type)
```

```
{
```

```
 cout<<line<<"\n Sales Class Constructor is call";
```

```
 this->id=id;
```

```
 this->name=name;
```

```
 this->price=price;
```

```
 this->book_type=book_type;
```

```
 this->qty=qty;
```

```
}
```

```
 void display_books()
```

```
{
```

```
 cout<<"\n"<<line<<"\n Book Details \n"<<line;
```

```
 cout<<"\n Id is:"<<this->id;
```

```
 cout<<"\n Name is:"<<this->name;
```

```
 cout<<"\n Price is:"<<this->price;
```

```
 cout<<"\n Type:"<<this->book_type;
```

```
 cout<<"\n Qty is:"<<this->qty;
```

```
 cout<<"\n"<<line;
```

```
}

~Sales()

{

 cout<<line<<"\n Sales Class Destructor is call";

 delete []name;

 delete []book_type;

}

};

class Academics : public Book

{

public:

 Academics(int id,char *name,float price):Book(id
 ,name,price,"Academics")

 {

 cout<<"\n Academics Calss Constructor is call";

 }

~Academics(){

 cout<<"\n Academics Class Destructor is call";

 delete []name;

 delete []book_type;

}

};
```

```
class Thrillers : public Book
{
public:
 Thrillers(int id,char *name,float price):Book(id
 ,name,price,"Thrillers")
 {
 cout<<line<<"\n Thrillers Class Constructor is call";
 }

 ~Thrillers(){
 cout<<line<<"\n Thrillers Class Destructor is call";
 delete []name;
 delete []book_type;
 }
};

void main()
{
 Book b1(101,"Tom & Jarry",320,"Story");
 Sales s1(101,"Book 1",450,"Programming",5);
 Sales s2(102,"Book 5",560,"Opps",6);
 s1.display_books();
 Academics a1(103,"Book 2",400);
 Thrillers t1(104,"Book 3",780);
```

```
//s1.display_books();
getch();
}

ostream & line(ostream &obj)
{
 cout<<"\n";
 for(int i=0;i<50;i++)
 {
 obj<<"-";
 }
 return obj;
}
```

=====

OUTPUT

=====

---

-----  
Book Class Constructor is call

---

-----  
Book Class Constructor is call

---

-----  
Sales Class Constructor is call

---

-----  
Book Class Constructor is call

-----  
Sales Class Constructor is call

-----  
Book Details

-----  
Id is:101

Name is:Book 1

Price is:450

Type:Programming

Qty is:5

-----  
-----  
Book Class Constructor is call

Academics Calss Constructor is call

-----  
-----  
Book Class Constructor is call

Thrillers Class Constructor is call

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :7) WAP to create student having data members  
(rollno, name, stream) as base

class. Derive class subject with marks of 5 subjects and  
apply respective

functionality. Calculate final result and display details of  
each student

from derived class. (multilevel inheritance)

=====

=====

=====

```
#include<iostream>
#include<conio.h>
#include<string>
using namespace std;

class Student
{
 int rno;
 char name[20];
 string stream;

public:
 void get_stud()
 {
 cout<<"\n Enter Rollno:";
 cin>>rno;
 cout<<"\n Enter Name:";
 cin>>name;
 cout<<"\n Enter stream:";
 cin>>stream;
 }

 void display_stud()
 {
 cout<<"\n Rollno is:"<<rno;
 cout<<"\n Name is:"<<name;
```

```
 cout<<"\n stream is:"<<stream;
}

};

class Subject :public Student
{
public:
int n,marks[5];

void get_data()
{
 get_stud();
 cout<<"\n Enter 5 subject marks:";

 for(int i=0;i<5;i++)
 {
 cout<<"\n Enter marks:";
 cin>>marks[i];
 }
}

void display()
{
 display_stud();
}
```

```
};
```

```
class Result : public Subject
```

```
{
```

```
 Subject sub;
```

```
 int sum;
```

```
 float res;
```

```
public:
```

```
 void insert_stud()
```

```
{
```

```
 sub.get_data();
```

```
}
```

```
 void display_stud()
```

```
{
```

```
 sub.display();
```

```
}
```

```
 void result()
```

```
{
```

```
 sum=0;
```

```
 for(int i=0;i<5;i++)
```

```
{
```

```
 sum=sum+sub.marks[i];
```

```
}
```

```
 res=sum/5;
 cout<<"\n Result is:"<<res;
}

};

void main()
{
 int n;
 Result r1[5];

 cout<<"\n Enter the total student:";
 cin>>n;
 for(int i=0;i<n;i++)
 {
 cout<<"\n-----"<<i+1<<"-----";
 r1[i].insert_stud();
 cout<<"\n-----\n";
 }

 for(int i=0;i<n;i++)
 {
 cout<<"\n-----"<<i+1<<"-----";
 r1[i].display_stud();
 r1[i].result();
 }
}
```

```
 getch();
}

=====
```

OUTPUT

```
=====
```

Enter the total student:3

```
-----1-----
```

Enter Rollno:1

Enter Name:shivangi

Enter stream:science

Enter 5 subject marks:

Enter marks:78

Enter marks:89

Enter marks:88

Enter marks:78

Enter marks:65

---

-----2-----

Enter Rollno:2

Enter Name:riya

Enter stream:commerce

Enter 5 subject marks:

Enter marks:78

Enter marks:89

Enter marks:88

Enter marks:65

Enter marks:45

---

-----3-----

Enter Rollno:3

Enter Name:priya

Enter stream:commerce

Enter 5 subject marks:

Enter marks:78

Enter marks:88

Enter marks:87

Enter marks:84

Enter marks:82

-----1-----

Rollno is:1

Name is:shivangi

stream is:science

Result is:79

-----2-----

Rollno is:2

Name is:riya

stream is:commerce

Result is:73

-----3-----

Rollno is:3

Name is:priya

stream is:commerce

Result is:83

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :8) An educational institution wishes to maintain a database of its employees.

The database is divided into a number of classes whose hierarchical

relationships are shown in fig-1. The figure also shows the minimum information

required for each class. Specify all the classes and define function to create

the database and retrieve individual information as and when required. Write

parameterized constructor for each class in the hierarchy.

```
=====
```

```
=====
```

```
=====
```

```
#include<iostream>
#include<conio.h>
#include<string>
using namespace std;

class Staff
{
public:
 int code;
 string name,designation;

public:
 Staff()
```

```
{
}
Staff::Staff(int v_code,string v_name,string
v_designation)
{
 code=v_code;
 name=v_name;
}

virtual void printdata(){}
friend class StaffCollection;
};

class Teacher :virtual public Staff
{
public:
 string sub, pub;

public:
 Teacher(int v_code, string v_name, string v_sub, string
v_pub):Staff(v_code,v_name,"Teacher")
{
 sub=v_sub;
 pub=v_pub;
}

void printdata()
```

```
{
 cout<<"Designation is:"<<this->designation<<endl;
 cout<<"-----"<<endl;
 cout<<"code is:"<<code<<endl;
 cout<<"name is:"<<name<<endl;
 cout<<"Designation is:"<<this->designation<<endl;
 cout<<"Subject is:"<<sub<<endl;
 cout<<"Publication is:"<<pub<<endl;
}

};

class Officer : public Staff
{
public:
 char grade;

public:
 Officer(int v_code,string v_name,int
v_grade):Staff(v_code,v_name,"Officer")
 {
 grade=v_grade;
 }

void printdata()
{
 cout<<"Designation is:"<<this->designation<<endl;
```

```
 cout<<"-----"<<endl;
 cout<<"code is:"<<code<<endl;
 cout<<"name is:"<<name<<endl;
 cout<<"Designation is:"<<this->designation<<endl;
 cout<<"grade is:"<<grade<<endl;
 }

};

class Typist :public Staff
{
public:
 int speed;
 string typist_type;
public:
 Typist(int v_code,string v_name,int v_speed):Staff(v_code,v_name,"Typist")
 {
 speed=v_speed;
 }
 virtual void printdata()=0;
};

class Casual : public Typist
{
public:
```

```
float daily_wages;

public:

 Casual(int v_code,string v_name,int v_speed,float
v_wages):Typist(v_code,v_name,v_speed)
{
 typist_type="Casual";
 daily_wages=v_wages;
}

void printdata()
{
 cout<<"Designation is:"<<designation<<endl;
 cout<<"-----"<<endl;
 cout<<"code is:"<<code<<endl;
 cout<<"name is:"<<name<<endl;
 cout<<"Designation is:"<<this->designation<<endl;
 cout<<"Speed is:"<<speed<<endl;
 cout<<"Type is:"<<typist_type<<endl;
 cout<<"Daily wages is:"<<daily_wages<<endl;
}

};

class Reguler : public Typist
{
```

```
public:
 Reguler(int v_code,string v_name,int
v_speed):Typist(v_code,v_name,v_speed)
{
 typist_type="reguler";

}
void printdata()
{
 cout<<"Designation is:"<<this->designation<<endl;
 cout<<"-----"
 cout<<"code is:"<<code<<endl;
 cout<<"name is:"<<name<<endl;
 cout<<"Designation is:"<<this->designation<<endl;
 cout<<"Speed is:"<<speed<<endl;
 cout<<"Type is:"<<typist_type<<endl;
}
};

class StaffCollection
{
 Staff *list[20];
 int count,size;
public:
 StaffCollection::StaffCollection(int n){
 *list = new Staff[n];
```

```
this->count=0;
this->size=n;

}
int getCount() { return this->count; }

void addStaff(Staff *s) {
 if (this->count >= this->size) {
 cout << "Staff is full!" << endl;
 return;
 }
 list[this->count] = s;
 this->count++;
}

void displaysaff()
{
 for(int i=0;i<this->count;i++)
 {
 list[i]->printdata();
 }
}

Staff *readstaff()
{
 Staff *member=NULL;
```

```
string name,subject,publication;
int speed,op,code;
float daily_wages;
char grade;
cout<<"\n-----";
cout<<"\n 1.Teacher \n 2.Officer \n 3.Reguler Typist
\n 4.Casual Typist";
cout<<"\n-----";
cout<<"\n enter your choice:";
cin>>op;

cout<<"\n Enter staff code and name:";
cin>>code>>name;
switch(op)
{
case 1:
 cout<<"\n Subject is:";
 cin>>subject;
 cout<<"\n Publication is:";
 cin>>publication;
 member = new
Teacher(code,name,subject,publication);
 break;
case 2:
 cout<<"\n Enter grade:";
 cin>>grade;
 member = new Officer(code,name,grade);
```

```
 break;

 case 3:
 cout<<"\n Enter Speed:";
 cin>>speed;
 member = new
Reguler(code,name,speed);
 break;

 case 4:cout<<"\n Enter Speed:";
 cin>>speed;
 cout<<"\n Enter daily wages:";
 cin>>daily_wages;
 member = new
Casual(code,name,speed,daily_wages);
 break;
 }

 return member;
}

};

int main()
{
 int n, op = 1;

 cout << "\n Enter total number of staff members: ";
 cin >> n;
```

```
StaffCollection coll(n);
Staff *member;
string code;

while (op) {
 cout << "\n1.Add Staff Member\n2.Display
Staff\n3.Exit \n";
 cout<<"\n Enter choice:";
 cin >> op;
 switch (op) {
 case 1: member = coll.readstaff();
 coll.addStaff(member);
 break;
 case 2: coll.displaysaff(); break;
 case 3:exit(0);
 default: cout << "Wrong choice!" << endl;
 }
}

 getch();
return 0;
}
```

=====

OUTPUT

=====

Enter total number of staff members: 2

- 1.Add Staff Member
- 2.Display Staff

Enter choice:1

- 
- 1.Teacher
  - 2.Officer
  - 3.Reguler Typist
  - 4.Casual Typist
- 

enter your choice:2

Enter staff code and name:101 shivangi

Enter grade:45

- 1.Add Staff Member
- 2.Display Staff

Enter choice:Wrong choice!

- 1.Add Staff Member

2.Display Staff

Enter choice:1

---

1.Teacher

2.Officer

3.Reguler Typist

4.Casual Typist

---

enter your choice:4

Enter staff code and name:102 riya

Enter Speed:78

Enter daily wages:4500

1.Add Staff Member

2.Display Staff

Enter choice:2

Designation is:

---

code is:101

name is:shivangi

Designation is:

grade is:4

Designation is:

-----

code is:102

name is:riya

Designation is:

Speed is:78

Type is:Casual

Daily wages is:4500

1.Add Staff Member

2.Display Staff

3.Exit

Enter choice:3

Roll No :30

Name :Ajinkya Rathod

Class :MCA-2

Subject :Object Oriented Concepts & Programming

=====

=====

=====

Definition :9) Consider a class network of fig 2. The class master derives information

from both account and admin classes which in turn derived derive information

from the class person. Define all the four classes and write a program to

create, update and display the information contained in master objects.

=====

=====

=====

```
#include<iostream>
```

```
#include<conio.h>
```

```
#include<string>
```

```
using namespace std;
```

```
class Person
{
public:
 int code;
 string name;
};

class Admin :virtual public Person
{
protected:
 int exp;
};

class Account :virtual public Person
{
protected:
 int pay;
};

class Master :public Admin, public Account
{
public:
```

```
int get_code()
{
 return code;
}

void getdata()
{
 cout<<"\n Enter Code:";
 cin>>code;
 cout<<"\n Enter Name:";
 cin>>name;
 cout<<"\n Enter Experience:";
 cin>>exp;
 cout<<"\n Enter Pay:";
 cin>>pay;
}

void update_data()
{
 cout<<"\n Enter Name:";
 cin>>name;
 cout<<"\n Enter Experience:";
 cin>>exp;
 cout<<"\n Enter Pay:";
 cin>>pay;
}

void display()
{
 cout<<"\n-----";
```

```
 cout<<"\n Code is:"<<code;
 cout<<"\n Name is:"<<name;
 cout<<"\n Experience is:"<<exp;
 cout<<"\n Pay is:"<<pay;
 cout<<"\n-----";
 }
};
```

```
void main()
{
 Master m1[5];
 int i,ch,cnt=-1,v_code,res,index=0,flag=0;
 char cont='n';
```

```
 cout<<"\n -----";
 cout<<"\n 1.Add";
 cout<<"\n 2.Update";
 cout<<"\n 3.display";
 cout<<"\n-----";
 do
 {
```

```
 cout<<"\n Enter Your Choice:";
 cin>>ch;
```

```
 switch(ch)
 {
```

case 1:

```
cnt++;
m1[cnt].getdata();
break;
```

case 2:

```
cout<<"\n Enter code:";
cin>>v_code;
for(i=0;i<=cnt;i++)
{
 res=m1[i].get_code();
 if(res==v_code)
 {
 index=i;
 flag=1;
 break;
 }
}
if(flag==1)
{
 m1[index].update_data();
}
else
{
 cout<<"\n Invalid Code";
}
```

```
 break;
 case 3:
 for(i=0;i<=cnt;i++)
 {
 m1[i].display();
 }
 break;
 }

 cout<<"\n Do You Want to continue:";
 cin>>cont;
}while(cont=='y'|| cont=='Y');

}
```

=====

OUTPUT

=====

- 1.Add  
2.Update  
3.display
- 

Enter Your Choice:1

Enter Code:101

Enter Name:raj

Enter Experience:4

Enter Pay:7800

Do You Want to continue:y

Enter Your Choice:1

Enter Code:102

Enter Name:jinal

Enter Experience:8

Enter Pay:5680

Do You Want to continue:y

Enter Your Choice:3

---

Code is:101

Name is:raj

Experience is:4

Pay is:7800

-----

-----

Code is:102

Name is:jinal

Experience is:8

Pay is:5680

-----

Do You Want to continue:y

Enter Your Choice:2

Enter code:102

Enter Name:jinal

Enter Experience:2

Enter Pay:5600

Do You Want to continue:y

Enter Your Choice:3

Code is:101

Name is:raj

Experience is:4

Pay is:7800

Code is:102

Name is:jinal

Experience is:2

Pay is:5600

Do You Want to continue:n