

Department of Computer Science
Gujarat University



Certificate

Roll No: 36

Seat No: _____

This is to certify that Mr./Ms. PREKSHA K. SHETH 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 CONCEPTS AND PROGRAMMING towards partial fulfillment of his/her Degree of Masters in Computer Applications.

Date of Submission

1st - JULY - 2020

Internal Faculty

Head of Department

Department Of Computer Science
 Rollwala Computer Centre
 Gujarat University

MCA - II

Subject: - C++ (Object oriented Concepts & Programming)

Name: - Sheth Preksha

Roll No.: - 36 **Exam Seat No.:** - _____

Sr. No.	Contents	Pg. No	Date	Signature
1.	Assignment - 1 POP vs OOP vs C++	1 to 23	5/21/20	✓
2.	C vs C++	"		
3.	merit, Demerit of oop			
4.	Scope resolution operator			
5.	inline function			
6.	static data members			
7.	private & public member function			
8.	function overloading			
9.	friend function			
10.	Inline vs Normal function			
	Assignment - 2 * Operator overloading, Console I/O & manipulators)	1 to 34	11/11/20	✓ 11/11/20 6/2/2020
	Assignment - 3 * C Template, Inheritance, Files, Namespaces & STL).	1 to 17	11/11/20	

Assignment 2.

Q1 What is Constructor? Explain explicit & parameterized Constructor in detail.

- A Constructor is a 'special' member function whose task is to initialize the objects of its class.
- It is a special because its name is the same as the class name.
- The constructor is invoked whenever an object of its associated class is created.
- It is called Constructor because it constructs the values of data members of the class.

Syntax:

Class integer

{

 int m, n;

 Public :

 integer(void); // declared.

}

 integer :: integer(void) // define.

 { m>0; n>0;

}

Teacher's Signature:

Explicit Constructor:

→ In most case, The automatic conversion adds to readability. It is, therefore, very useful. At times automatic conversion is not required, we can easily omit the creation of this conversion operator by using keyword explicit before the class name while defining the object.

Suppose we want like,

brother fourthbrother ("Rikipinting");
There we need explicit constructor.

#Class brother

{

String name;

public :

explicit brother (String fourthname)

{ name = fourthname;

}

};

main()

{

brother firstbrother = brother ("steve");

brother secondbrother = brother ("hetan");

}

Parameterized Constructor:

- When the constructor contains single or multiple arguments then its called parameterized constructor.
- This is useful when we need constructor for creating objects with assigned data values.

Class Point

{

int x, y;

public :

Point (int a, int b)

{

x = a;

y = b;

}

void display()

{

cout << "A : " << a << "B : " << b;

}

};

main()

{

Point P(1, 1);

P.display();

},

Q2 Explain Copy Constructor with example

- A constructor which is used to initialize the members of a newly created object, by copying the members of an already existing object into the members of newly created object is known as 'copy constructor'
- It's defined explicitly by the programmer

Class code

```

int id;
public :
    void init(int x)
    {
        id=x;
    }
    void disp();
{
    cout<<id<<endl;
}
Code C() { }
Code (Code fa)
{
    id=a.id;
}
main()
{
}
  
```

```

Code obj;
obj.init();
obj.dispc();
Code obj2 = obj;
obj2.dispc();
}

```

O/P :

5 5

Q.3 Explain MIL with example.

- Member wise initialization list or member initialization list is the method for initialization the members of a class using the constructor function.
- It provides an alternative for providing initializations outside constructor. today
- MIL appears in b/w the body of constructor & header.
- It starts with : & has all initialization separated by a (,) Comma.
- The initialization has a normal syntax of initializations,
- i.e. Variable (value).

Class Time

{

Public :

int h;

int m;

int s;

void Showtime()

{

cout << "Time is " << h << m << s;

}

Time (int t_h, int t_m, int t_s)

: h(t_h),

m(t_m),

s(t_s);

{ } // the empty body.

};

void main()

{

Time time1(12, 15, 20);

time1.Showtime();

}

Q4

What is destructor? Why we need it?



A destructor is a special member function of a class, whose name is same as the class name preceded with a tilde character (~), which is involved the automatic destruction of the objects created by the constructor.

Class A



```
int z;
public:
    A (int);
    ~A (void);
```



```
A :: A (int z)
{
```

z = 2;

Called "constructor called" (Ans.)



```
A :: ~A ()
```



Called "Destructor Called" (Ans.)

```
int A :: get () const
```



return z;



name)

{

A a1(5);

cout << a1.get()

A a2(10);

cout << a2.get()

5

~~Q5~~

Explain Unary & Binary operator overloading with example.

→ Unary operators:

→ Let us consider the unary minus operator. A minus operator when used as a unary, takes just one operand.

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

→ The unary minus when applied to an object should change the sign of each of its data items.

Class Space

E

int x, y, z;

Public :

void getdata (int a, int b, int c);

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

Teacher's Signature _____

```
void display();
void operator -();
```

{}

```
void space :: getdata (int a, int b, int c)
```

{}

```
x = a;
```

```
y = b;
```

```
z = c;
```

{}

```
void space :: display()
```

{}

```
cout << "x = " << x;
```

```
cout << "y = " << y;
```

```
cout << "z = " << z;
```

{}

```
void space :: operator -()
```

{}

```
x = -x;
```

```
y = -y;
```

```
z = -z;
```

{}

```
main () {
```

```
    space s;
```

```
    s.getdata (10, -20, 30);
```

```
    cout << "s = ";
```

```
    s.display();
```

```
- s;
```

```
    cout << "-s = ";
```

```
    s.display();
```

{}

Binary operator:

→ The same mechanism can be used to overload a binary operator is unary operator overloading.

class Complex {

 float x, y;

 public :

 Complex (float real, float img)

}

 x = real;

 y = img;

}

 Complex operator + (Complex);

 void display () ;

};

complex Complex :: operator + (Complex);

{

 Complex = temp;

 temp.x = x + c.x;

 temp.y = y + c.y;

 return temp;

}

void Complex :: display

{

 cout << x << " + " << y

}

main()

{

Complex C1, C2, C3;

C1 = Complex (2.5, 3.5);

C2 = Complex (3.6, 2.7);

C3 = C1 + C2;

cout << C1.display();

cout << C2.display();

cout << C3.display();

}

Q.6

Explain the need of friend as operator function in overloading an operator.

- As mentioned earlier, we can use friend as operator functions.
- We seen that using friend is a violation of principle of information hiding & is to be avoided as far as possible.
- We know that member functions can't change the 1st arg. & it is always the invoking objects.
- At time, it is a friday. There are two cases where it is really important to use friend function.
- Suppose, we want an operation of multiplication in a matrix class.

- we have matrix m1 & we would like to have s * m1 to indicate multiplication of salary name s to matrix m1
- It is an operation of multiplying s to all the elements of m1. But it is not possible using member operator because the first argument is not an object.
- Instead if we want to provide (m1 * s) we can
- If we are friend, Both cases can be entertained

```
class Matrix {
```

```
    int element[3][3];
```

```
public :
```

```
    Matrix();
```

```
    Matrix(int tempmatrix[3][3])
```

```
{
```

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

```
        for (int j = 0; j < 3; j++)
```

```
            Element[i][j] = tempmatrix[i][j];
```

```
}
```

```
    void read()
```

```
{
```

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

```
        for (int j = 0; j < 3; j++)
```

```
            cin >> Element[i][j];
```

```
}
```

```

void display()
{
    for (int i=0; i<3; i++)
        for (int j=0; j<3; j++)
    {
        cout << element[i][j] << " ";
    }
    cout << endl;
}

friend matrix operator * (matrix, int)
friend matrix operator * (int, matrix)
{
    matrix operator * (matrix Tempmatrix, int multiplication)
    {
        for (int i=0; i<3; i++)
            for (int j=0; j<3; j++)
        {
            Tempmatrix.element[i][j] = multiplier * Tempmatrix.element[i][j];
        }
        return (Tempmatrix);
    }

    matrix operator * (int multiplier, matrix Tempmatrix)
    {
        for (int i=0; i<3; i++)
            for (int j=0; j<3; j++)
        {
            Tempmatrix.element[i][j] = multiplier * Tempmatrix.element[i][j];
        }
        return (matrix(Tempmatrix));
    }
}

```

void main()

```

    {
        int ArrayOfInt1[8] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
        int ArrayOfInt2[8] = {4, 5, 6, 7, 8, 9, 1, 2, 3};
        matrix M1(ArrayOfInt1);
        matrix M2(ArrayOfInt2);
        matrix M3, M4;
        M1.display();
        M3 = M1 * 5;
        M3.display();
        M2.display();
        M4 = 5 * M2;
        M4.display();
    }

```

Q. 8. Why we need to user defined conversion & explain 4 different type cases where user defined conversion are needed.

Ans:- Object assignment is simple when both object involve are of same type. But when they are different type then we need to define user defined conversion.

→ As the objects are different & the members are not same, the compiler cannot go far member by member copy.

Teacher's Signature : _____

- User define conversions are needed in 4 different cases :
- Conversion from Built-in Data Type to object :
 - one specific method to solve the problem of dissimilar objects assignment is to use Constructor.

Example:

Suppose we define a class length & would like to have constructor as follows:

```
class Length
{
    int L;
    ...
    public:
        Length (int TempLength = 0)
    {
        L = TempLength;
    }
}
```

- Object to Built-in Data Type.

→ suppose we have loggedInUser Class

With contents such as Name, Token.
 In the function we may need to
 print error message. In that case
 we write cout << LIU.Name where
 LIU is object.

- when we write a function such as
 Printname () we need to write
 LIU.Printname () .

Ex:

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

```
Class LoggedInUsers
{
```

String Name;

int TokenNo;

Static int TotalLoggedIn;

Public :

LoggedInUsers ()

{

TotalLoggedIn +=;

cout << "in you are user No" << TotalLoggedIn;

Void InsertName ()

{

cout << "Insert Name of new user:"

cin >> Name;

}

Teacher's Signature : _____

```

String()
{
    return Name;
}

int LoggedInUser :: TotalLoggedIn;

void main()
{
    LoggedInUser * ArrayofUser[100];
    int index = 0;
    int choice;

    while (true)
    {
        cout << "1. New User";
        cout << "2. List Users";
        cout << "3. Exit";
        cout << "Enter your choice";
        cin >> choice;
        if (choice == 1)
        {
            if (index == 100)
                cout << "Too many users";
            cout << endl;
            ArrayofUsers[index] = new LoggedInUser();
            ArrayofUsers[index] -> InsertName();
            index++;
        }
    }
}

```

else if (choice == 2)

{

for (int i = 0; i < index; i++)

{

String NameUser = *Arrayof50s[i];

CountNameUser;

{

}

else

{

exist(0);

}

}

3) Object type Using Constructor :

→ It is possible to convert from one type of object into another using either Constructor or Conversion functions.

→ There are 2 cases in such Conversions.

- They are conversions from foreign object into a native one & vice versa.
- Conversion from foreign object to native one is done using Constructor.

Ex:

#include <iostream>

#include <string>

Class Polar

{

double Radius;

double Angle;

public:

polar (double Radius=0, double Angle=0)

{

Radius = radius;

Angle = angle;

{

double getRadius()

{

return Radius;

{

double getAngle()

{

return Angle;

{

Class Cartesian

{

double x;

double y;

Public :

Cartesian (double TempX=0, double TempY=0)

{

x = TempX;

y = TempY;

{

Cartesian (Polar PolarPoint)

{

```

double TRadius = PolarPoint.getRadius();
double TAngle = PolarPoint.getAngle();
x = TRadius * Cos(TAngle);
y = TRadius * Sin(TAngle);

```

}

void show()

{

```

cout << "x : " << x << ", " << y << endl;

```

}

void main()

{

Cartesian Point(10,10);

Polar Point(10,45);

Polar point1;

Cartesian point2;

(Point2 = Point1;

Point2.show());

}

4) Object Type Using Conversion Function:

When a native object needs to be converted into foreign object, operator functions are used. They can be used in some manner as used while converting to a basic type from an object.

```
#include <iostream>
```

```
#include <string>
```

```
#include <cmath>
```

```
class cartesian:
```

```
class Polar
```

```
{
```

```
    double Radius;
```

```
    double Angle;
```

```
public:
```

```
    Polar(double TRadius=0, double Tangle=0)
```

```
{
```

```
    Radius = TRadius; Angle = Tangle;
```

```
{
```

```
    double getRadius {
```

```
        return Radius;
```

```
{
```

```
    double getAngle {
```

```
        return Angle;
```

```
{
```

```
    void show()
```

```
{
```

```
        cout << "C" << Radius << ", " << Angle << endl;
```

```
{
```

```
}
```

```
class cartesian
```

```
{
```

```
    double x, y;
```

```
public:
```

Cartesian (double Tx=0; double Ty=0)

{

$x = Tx$; $y = Ty$;

{

Cartesian (Polar polarpoint) {

double rRadius = PolarPoint.getRadius();

double rAngle = PolarPoint.getAngle();

$x = rRadius * \cos(rAngle);$

$y = rRadius * \sin(rAngle);$

{

operator Polar ()

{

double rAngle = area(x, y);

double rRadius = sqrt(x*x + y*y);

return Polar(rRadius, rAngle);

void show()

{

cout << "x" << x << ", " << y << "m";"

{

void main()

{

Cartesian CPoint1(10,10);

Polar PPoint2(10, double cos());

Polar PPoint1, Cartesian CPoint2;

CPoint2 = PPoint2;

PPoint1 = CPoint1;

CPoint1.show();

PPoint1.show();

{

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

Q.8 What is template function? Explain non generic parameter in template function with example.

→ Function template are generic function that works for any data type that is passed to them. The data type is not specified while writing the function.

→ While using that function, we pass the data type & get the required functionality.

* Non generic parameters in function template:

→ We can also pass non generic argument to a function.

→ However pass the array size as well to the generic function as shown in the following code.

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

```
template <typename Type>
```

```
void generatearray (Type temparray[], int size)
```

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

```
    for (int j=i+1; j<size; j++)
```

{ }

Teacher's Signature : _____

```

if (TempArray[i] > TempArray[j])
{

```

```

    type TempGeneric = TempArray[i];
    TempArray[i] = TempArray[j];
    TempArray[j] = TempGeneric;
}
}

```

```
void main()
```

```

int Array2[] = {1, 8, 9, 5, 4, 12, 11, 45, 23, 89, 65, 34, 12, 44, 65, 22};
CountBubbleSort (Array2, 16);
for (int i = 0; i < 16; i++)
{

```

```
    cout << " " << Array2[i] << endl;
}
}

```

Q.9 What is class template? Explain classes with multiple generic data types.

- A class created from a class template is called a template class.
- The syntax for defining an object of template class is:

```
Classname<Type> Objectname (arglist);
```

#include <iostream>

Using namespace std;

template < class T1, class T2 >

Class Test

{

T1 a;

T2 b;

Public:

Test (T1 x, T2 y)

{

a = x;

b = y;

}

void show()

{

cout << "and" << b << endl;

}

};

int main()

{

cout << "Initialzing the class template as
test, with filout & int ob." << "m";

Test < float, int > test1(1.23, 123);

test1.show();

Test < int, char > test2(100, 'w');

test2.show();

}

O/P:

Initialising the class template as follows with
float & int dt.

1.23 and 123

100 and 4.

~~Q.10~~

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

- The Template class can also have static data members. The static variable will have one instance for one initialization of the template class.
- Thus, there will be two diff. static members for Static <int> & for Stack (char).
- for all objects of a single class there is only one instance of static member. The way a static member is defined is analogous to the way member functions are defined outside the template class.

Ex:

```
#include <iostream>
```

```
#include <string>
```

```
template <typename Elementtype>
```

```
class Stack
```

{

```
int Stackpointer;
ElementType StackArray[10];
Public:
```

```
Static int TotalStacks,
Stacks();
```

{

```
Stackpointer = 0;
```

```
TotalStacks = 0;
```

}

```
void Push(ElementType);
```

```
ElementType Pop();
```

};

```
template <typename ElementType>
```

```
int Stack(ElementType) :: TotalStacks;
```

```
template <typename ElementType>
```

```
void Stack <ElementType> :: Push (ElementType value)
```

E

```
if (Stackpointer > 9)
```

{

```
cout << "Stack overflowed";
```

}

```
else
```

{

```
StackArray[Stackpointer] = value;
```

```
Stackpointer++;
```

}

{

```

template<typename Elementtype>
Elementtype* Stack<Elementtype>::pop()
{
    if (Stackpointer == 0)
    {
        cout << "stack underflow";
        return NULL;
    }
    Stackpointer--;
    return stackArray[Stackpointer];
}

void main()
{
    Stack<int> mystack;
    cout << "int stack element" << mystack.TotalStack();
    Stack<char> yourstack;
    Stack<int> mystack2;
    Stack<char> yourstack2;
    getch();
}

```

⇒ Export Keyword:

- The keyword `export` is useful when a template function is defined at a single place & the declaration are used at other place.

Export template <typename Type>

void bubblesort (Type TempIntArray[])

{

// Body of function

}

Q.11. Explain the Advantages of C++ I/O over C I/O.

- C++ I/O is object-oriented. objects represent streams in C++. cout is an object of the output stream class, whereas cin is an object of the input Stream class.
- C++ I/O Stream contains richer formatting option than C. It is possible to have the programmers own format operators known as user-defined manipulators in C++.
- Though apparent at first glance, C++ I/O is much easier to use. we can take input using cin without the '&' operator unlike scanf(). we also use overloaded << and >> operators for I/O. for file manipulation C++ provides constructor function - reading & writing from text file. sum can be performed using >> and << operators.

Teacher's Signature :

Q2

Explain ios member functions for formatting.

- The built in ios functions is also called ios member functions.
- The following are the list of ios member functions:

width()

precision()

fill()

setf()

unsetf()

- The prototype for all the following function is :
(old value of stream) fun-name (< specified new val >)

1) width(): specifies the minimum field width for display. The width() reacts itself after the first O/P after statement. The O/P of the statement.

cout.width(10);

cout<<"eff";

2) precision(): specifies precision, that is, the number of digits to be displayed after the decimal point. The default value of precision() is six.

→ The precision function is important while displaying numbers in scientific notation.

3) `fill()`: The function `fill(char)` fills the subsequent empty positions of fields by the fill characters specified.

`Cout.fill('*');`

`Cout.width(10);`

`Cout << " C lang";`

O/P: `*****C lang.`

4) `setf()`: Specifies the format flags that control O/P display such as left or right justification, scientific notation display & displaying base of the number.

5) `unsetf()`: provides undo operation for the operations with `setf()`.

Q.13 What is manipulator? Explain different formatting manipulators also give brief description about user defined manipulator.

→ Manipulators are special functions for formatting they can do all the formatting, that is done by the i/o.

member function

1) **setw()** : used to set field width to 'width'.

Ex:

```
cout << setw(5) << 30 << endl;
cout << setw(5) << 30 << endl;
```

2) **setprecision()** : used to set the floating point precision to decimal.

Ex:

```
cout << setprecision(3) << sqrt(3) << endl;
cout << setprecision(4) << sqrt(4) << endl;
```

3) **setfill()** : used to fill empty column obtained after using the manipulator 'setw()' by character.

Ex

```
cout << setfill('9') << setw(6) << 10 << endl;
```

4) **setiosflags()** : sets the format flags to flags

→ flags include 'ios::showpoint', 'ios::showpos' etc.

5) **resetiosflags()** : clear the format flag specified by flags.

6) **endl** : used to end line in program & flush stream

* Userdefined manipulators :

→ manipulators can also be defined to suit particular requirements. The following program will demonstrate how we can write & use our own manipulator.

Ex:

```
#include <iostream>
```

```
#include <iomanip>
```

// 1st manipulator

ostream & Printheading(ostream & tempoint)

{

```
tempoint << setw(80) << setiosflags(ios::left);
tempoint << "First Higher Secondary" << endl <<
setw(20) << "Standard XII" << endl;
return tempoint;
```

}

// 2nd manipulator

ostream & Printmarksheet(ostream & tempoint)

{

```
tempoint << setw(15) << setiosflags(ios::left) <<
setiosflags(ios::fixed) << setprecision(2) <<
setiosflags(ios::showpoint);
```

```
tempoint << "Roll No" << setw(15) << "Name" << setw(10) <<
```

```
setprecision(2) << "marks" << endl;
```

```
return tempoint;
```

}

Teacher's Signature :

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

11.32) manipulator :

OStream & Printline (ostream & Tempout)

{

Tempout << "....";

Tempout << endl;

return Tempout;

{

int main()

{

Cout << printline;

Cout << PrintHeading << printline << PrintMarksheetHeading;

Cout << printline;

Cout << setw(15) << 1 << setw(15) << "Lara" << setw(10) <<
setprecision(2) << 355.50 << endl;

Cout << setw(15) << 2 << setw(15) << "Sathish" << setw(10) <<
setprecision(2) << 190.75 << endl;

Cout << printline;

{

Assignment - 3

Q1 What is inheritance? Example multiple inheritance & discuss the problem with multiple inheritance & how to solve that?

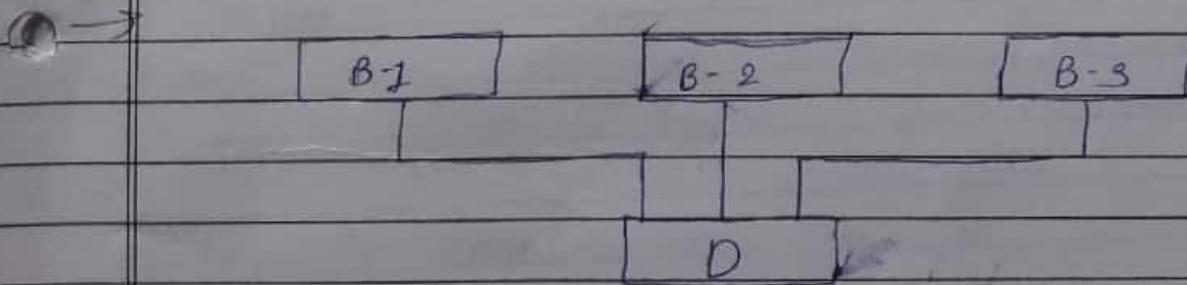
→ The mechanism of deriving a new class from an old class is called inheritance.

→ The derived class inherits some or all the traits from the base class.

→ Inheritance & one with several base classes is called multiple inheritance.

* Multiple Inheritance:

→ A class can inherit the attribute of two or more classes as shown in fig. known as multiple inheritance.



→ Occasionally, we may face a problem in using multiple inheritance, when a function with the same name appears in more than one base class. Consider the following classes.

Class m

Public :

void display()

{

cout << "class m" << endl;

}

};

Class N

{

Public :

void display()

{

cout << "class N" << endl;

}

};

Q2 Which display function is used by the derived class when we inherit these two classes?

→ class P : public m, public N

{

Public :

void display()

{

};

m::display(); } // overrides display() of m & N

int main()

I

P p;

P::display();

}

→ we can solve this problem by defining a named instance within the derived class using the scope resolution operator with function as shown upper like:

m::display();

Q.3 what is the difference b/w Is-a & Part-of relationship.

Part of

Is-a

→ defines a strong-coupled relationship b/w two entities where the one entity is part of another & both need each other for their existence.

→ Define a weak-coupled relationship b/w few entities where one entities could be part of another but either can exist without the other.

→ eg engine is part of vehicle

→ eg. car is a variable.

- Part of relationship implies real ownership of its components. → Is a relationship does not necessarily own any of its aggregation.
- part of relationship a stronger bond of its components → Is a relationship has weaker or looser bonds with aggregates.

Q.4 Explain derivation using different access modifiers

→ Public derivation :

- Fig shows the first case where a class is derived using a public access specifier from the base class. The private member of base class is copied to the derived class but it is not visible.
- The public members of the base class are copied as public in derived class.

BasePrivate :

int privateBaseInt;

Public :

int publicBaseInt;

void setPrivateBaseInt(6);

Derived

Not directly accessible

int privateBaseInt;

Public :

int publicBaseInt;

void setPrivateBaseInt();

Private :

int privateDerivedInt;

void setPrivateDerivedInt();

10

Private Derivation:

- The public members of the base class become the private members of the derived class. Again private member can not inherited. Though the private members of the base class are copied to the derived class, they are Not Visible.

Base :Private :

int Private A;

Public :

int Public A;

void setPrivate();

Derived :

Not accessible:

int Private A;

int Public A;

void setPrivate();

Private :

int Derived A;

void setDerivedA();

→ Protected Derivation:

→ In case of Protected derivation the protected members of the base class become protected members of the derived class.

→ The public members of base class also become the protected members of the derived class.

Q.5 Explain Runtime polymorphism.

* Runtime polymorphism :

→ This type of polymorphism is achieved by function overriding.

* Function Overriding :

→ On the other hand occurs when a derived class has a definition for one of member function is base class. That base function is said to be overridden.

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

```
Class base
{
```

```
Public:
```

```
virtual void print()
```

```
[
```

```
cout << "Print base class" << endl;
```

```
]
```

```
void show()
```

```
{
```

```
cout << "Base class" << endl;
```

```
]
```

```
};
```

```
Class derived : Public base.
```

```
{
```

```
Public:
```

```
void print()
```

```
[
```

```
cout << "Print derived class" << endl;
```

```
]
```

```
void show()
```

```
{
```

```
cout << "Show derived class" << endl;
```

```
]
```

```
};
```

```

int main()
{
    base *bptr;
    derived d;
    ptr = &d;
}

```

```

bptr-> Print();
bptr-> Show();
return 0;
}

```

Q.6 Describe the use of Pointer with eg.

→ Pointer is a derived data type that refers to another data variable by storing the variable's memory address rather than data.

→ Declaring & Initializing pointer:

Declaration:

```
datatype *pointer - variable;
```

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

```

int main()
{
}
```

```
int a = 10;  
int *ptr;  
ptr = &a;  
cout << "The value of a is " << *ptr;  
*ptr = *ptr + 1;  
cout << "The revised value of a  
is : " << *ptr;  
getch();  
return 0;
```

3

Q.7 Explain following functions:

1) fseek()

→ used to move file pointer associated with given file to a specific position.

Syntax:

```
int fseek(FILE *pointer, long int offset, int position)
```

2) feof()

→ int feof(FILE *stream)

→ takes a file stream as argument
8. returns an integer value which

Specifies if the end of file has been reached.

3) fread()

→ The fread() function in C++ reads a specific number of characters from the given input stream.

(void *buffer, size_t size, size_t count,
file *stream)

4) fopen()

→ The fopen() function opens a file indicated by fname & returns a Stream associated with that file.

e.g. FILE *fopen(const char *filename, const char *mode);

f 5) fclose():

→ int fclose(FILE *fstream)

→ The fclose() function takes a single argument, a file stream which is to be closed.

Q.8 Explain Namespace in details.

- Namespaces can define with only variables inside or with function & class also as part of it.
- Namespace declaration don't have access specifies like public, private etc.

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

```
namespace ns1
{
    int no;
    string name;
}
```

```
int main()
{
    cin >> ns1::no;
    cin >> ns1::name;
    cout << ns1::no << endl;
    cout << ns1::name << endl;
}
```

Q.9 Explain IO modes in file.

IO mode

Effect

`ios::in` file opens in input mode

`ios::out` file opens in output mode

`ios::app` file opens in append mode.

`ios::ate` when file opened the file pointer moves at the end of file.

`ios::trunc` when the file is opened the contents are erased.

`ios::noremove` checks if the file exists;

`ios::binary` the file would be opened in binary rather than default text mode.

Q10 What are the advantages of saving the data in binary form?

- In binary files, the numbers are stored as binary numbers. 0 is stored as binary zero.
- In text file storage & retrieval may require character conversions.
- In binary file this type conversions is not required.
- Example:

Class Institute

Protected :

```
int m_id;
string type;
string I_name;
string M_name;
```

Public :

Institute()

{

Iname = "RollaIII";

}

void addNumber

{

cout << "id";

cin >> M_Id

cout << "Name:";

cin >> M_Name

{

void dispC()

{

cout << "Id:" << m_Id;

cout << "Name:" << M_Name;

cout << "Type" << type;

{

class student

{

public :

student();

{

member type "student";

{

{};

class Teacher

{

public :

Teacher();

{

member type "Teacher";

{

{};

void main()

{

Teacher T[10];

Student S[10];

Institute I;

int choice1, choice2, i;

static int count = 0;

do

{

cout << "1. Add member "

" 2. Display "

<< Exit;

cin >> choice1;

switch (choice1)

{

case 1 :

cout << "1. Add stud : " << " 2. Add teacher : ";

cin >> choice2;

if (choice2 == 1)

{

S[count].add_member();

}

else

{

T[count].add_member();

}

count++;

break;

Case 2 :

cout << "disp. stud." << " & disp. Teacher";

choice2;

if (choice2 == 1)

{

for (i=0; i<count; i++)

{

s[i].disp();

}

}

else

{

for (i=0; i<count; i++)

{

t[i].disp();

}

break;

case 3 :

exit(0);

default :

cout << "Enter Proper choice : " << endl; } }

Q.11 Explain STL in detail

→ STL is generic software components & algo. It is a generalized library & so, its components are parameterized.

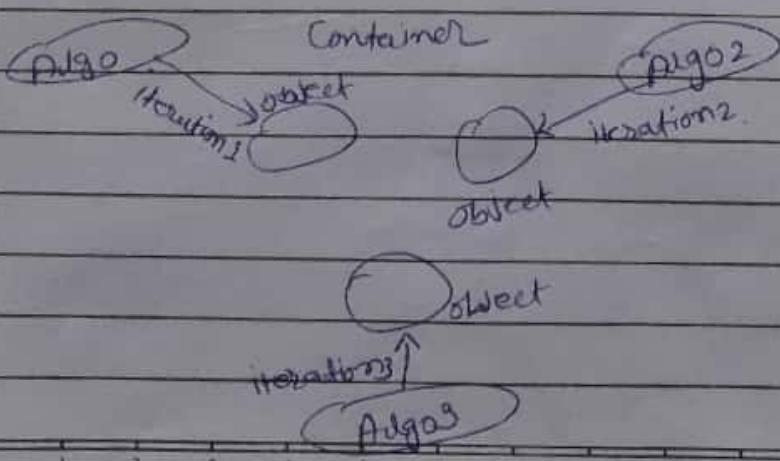
* Components of STL:

1) Container : A container is an object that actually stores data. It is a way data is organized in memory.

2) Algorithm : An algo is a procedure that is used to process the data contained in the containers.

3) Iteration : An iteration is an object that points to an element in a container.

→ Relationship b/w three component:



Teacher's Signature :

**DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. - II**

ROLL NO : 36
NAME : Preksha K. Sheth
S U B J E C T : Object oriented programming

NO.	TITLE	PAGE NO.	DATE	SIGN
	***** ASSIGNMENT : 1 *****	58 to 104		
1.	Program to demonstrate the use of static variables in class		1/7/2020	
2.	Define a class to represent a bank account.		1/7/2020	
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.		1/7/2020	
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 than 20.0.		1/7/2020	
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.		1/7/2020	
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.		1/7/2020	

**DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. - II**

ROLL NO : 36

NAME : Preksha K. Sheth

S U B J E C T : Object oriented programming

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 Car Collection. Car Collection contains array of cars. Car Collection class should contain member functions as Add, delete, modify and replace. Car Collection is to be defined as friend of Car class. Write C++ programs to test your classes.		1/7/2020	
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.		1/7/2020	
9.	Program to maintain the inventory of the books that are being sold at the shop.		1/7/2020	
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) .		1/7/2020	
	***** ASSIGNMENT : 2 *****	104 to 135		
1.	WAP to use binary operator + add two object of class Numbers having num1 and num2 as its data members and display result.		1/7/2020	
2.	WAP to overload operator * which multiply a number to each element of an array within a class array Container and display the result.		1/7/2020	
3.	WAP to Overload the *, +,-, ==, != and = operators for the complex class.		1/7/2020	
4.	WAP to define an object m1 of matrix class, use m1<cout		1/7/2020	

**DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. - II**

ROLL NO : 36

NAME : Preksha K. Sheth

S U B J E C T : Object oriented programming

5.	WAP to define a matrix class and overload the * operator to multiply a number with matrix.		1/7/2020	
6.	WAP to define a class Date with properties int month; int day; int year; overload the following operators.		1/7/2020	
7.	WAP to define a class Time with properties int hour; int minute; int second; overload the following operators.		1/7/2020	
8.	Write a menu driven program that can perform the following functions on strings.		1/7/2020	
Console I/O Manipulators:				
9.	Design a manipulator to provide the following output specifications for printing float values		1/7/2020	

10.	Program to print the marksheets of the student.		1/7/2020	
	***** ASSIGNMENT : 3 *****	135 to 188		
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		1/7/2020	
2.	Write an object oriented program to implement a generic Stack. Incorporate all the possible operation on Stack in the program.		1/7/2020	
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.		1/7/2020	
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.		1/7/2020	
5.	WAP Implement template sort with a non type size.		1/7/2020	
	INHERITANCE :			

**DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. - II**

ROLL NO : 36

NAME : Preksha K. Sheth

S U B J E C T : Object oriented programming

6.	Program of inheritance having Base class Book.		1/7/2020	
7.	Program to calculate subject marks with class Student.		1/7/2020	
	Program : The class master derives information from both account and admin classes which in turn derived derive information from the class person.		1/7/2020	
10.	Program create a class student from which the classes test and sports are derived,		1/7/2020	
11.	Write a Program to perform following operation on text file.		1/7/2020	
12.	Write a program to create a file student to store name and marks of 5 students and then display them.		1/7/2020	
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.		1/7/2020	
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.		1/7/2020	
15.	WAP that stores and displays the records of the customer from a file.		1/7/2020	
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.		1/7/2020	
17.	WAP to define a vector and use for the student class to store and display the information.		1/7/2020	

```
*****
*****  
Roll No      : 36  
Name         : Preksha Sheth  
Class        : MCA-II  
Subject      : OOPC  
*****  
*****
```

ASSIGNMENT - 1

```
*****  
*****  
Q 1 : Demonstrate the use of static variables in a class by using it to  
count the number of objects created in the program.
```

```
*****  
*****  
#include<iostream>  
using namespace std;  
class counter  
{  
    private:  
    static int c;//static data member  
    public:  
    counter()//counstructor  
    {  
        c++;  
    }  
    void disp_count()  
    {  
        cout<<"Total Objects : "<<c<<"\n";  
    }  
};  
int counter::c=0;//initialization of var  
void main()  
{  
    counter c1,c2,c3,c4,c5,c6,c7;  
    c1.disp_count();  
}
```

Output:

```
Total Objects : 7  
Press any key to continue . . .
```

```
*****  
*****  
Q 2 : Define a class to represent a bank account. Include the following  
members :
```

DATA MEMBERS MEMBER FUNCTIONS

```
-----  
Name of depositor  
(1) To assign initial valuesAccount Number  
(2) To Deposit the amountType 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.

```
*****  
*****  
#include<iostream>  
#include<string>  
using namespace std;  
  
class bank  
{  
  
public :  
    string name;  
    int acc,amount;  
    string type;  
    void input()  
    {  
        int ty=0;  
        cout<<"Creating New Bank Account"<<endl;  
        cout<<"Enter Account Number : "<<endl;  
        cin>>acc;  
        cout<<"Enter Name of The Accountant : "<<endl;  
        cin.ignore();  
        getline(cin,name);  
        while (ty<3)  
        {  
  
            ty=0;  
            cout<<"Press 1 for Saving Account \nPress 2 for Current  
Account"<<endl;  
            cin>>ty;  
            if(ty==1)  
            {  
                type="Savings Account";  
                ty=4;  
            }  
            else if(ty==2)  
            {  
                type="Current Account";  
                ty=4;  
            }  
            else  
            {  
                cout<<"Try Again Invalid Input!!"<<endl;  
                ty=1;  
            }  
        }  
        ty=0;  
        amount=0;  
        while(ty<2)
```

```

    {
        ty=0;
        cout<<"Enter Amount : "<<endl;
        cin>>amount;
        if(amount<5000)

        {
            cout<<"Insufficient Amount Try Again"<<endl;
            ty=1;
        }
        else
        {
            ty=3;
        }
    }

}

void deposit()
{
    int amt1;
    cout<<"Enter The Amount to Be Deposited : "<<endl;
    cin>>amt1;
    amount +=amt1;
    cout<<"Amount Successfully Deposited In Account"<<acc<<endl;
}
void withdraw()
{
    int amt1;
    cout<<"Enter The Amount to Be Withdrawn : "<<endl;
    cin>>amt1;
    if(amt1<=amount)
    {
        amount -=amt1;
        cout<<"Amount Successfully Withdrawn from Account"<<acc<<endl;
    }
    else
    {
        cout<<"Unavailable Balance!!!"<<endl;
    }
}

void display()
{
    cout<<"Bank Details Of Account Number :- "<<acc<<endl;
    cout<<"Accountant Name :- "<<name<<endl;
    cout<<"Type Of Account :- "<<type<<endl;
    cout<<"Amount           :- "<<amount<<endl;
}
};

void main()

```

```

{
bank a[9];
int ck=0;

for(int i=1;i<=100;i++)
{
    cout<<"\nPress 1 for Creating New Account\nPress 2 for Displaying
Account Details \nPress 3 Deposit\nPress 4 Withdraw"<<endl;
    cin>>ch;
    switch(ch)
    {
        case 1:
        {
            a[i].input();
            ck=i;
            break;
        }

        case 2:
        {
            int acc1,flag=0;
            if(ck<1)
            {
                cout<<"There Are No Bank Account"<<endl;
                break;
            }
            cout<<"Enter The Account Number You want view"<<endl;
            cin>>acc1;
            for(int j=1;j<=ck;j++)
            {
                if(acc1==a[j].acc)
                {
                    a[j].display();
                    flag=1;
                }
            }

            if(flag==0)
            {
                cout<<"There is no bank account like that!!!"<<endl;
            }
            break;
        }

        case 3:
        {
            int acc1,flag=0;
            if(ck<1)
            {
                cout<<"There Are No Bank Account"<<endl;
                break;
            }
            cout<<"Enter The Account Number You want view"<<endl;
            cin>>acc1;
            for(int j=1;j<=ck;j++)

```

```

        {
            if(acc1==a[j].acc)
            {
                a[j].deposit();
                flag=1;
            }
        }

        if(flag==0)
        {
            cout<<"There is no bank account like that!!!!"<<endl;
        }
        break;
    }
case 4:
{
    int acc1,flag=0;
    if(ck<1)
    {
        cout<<"There Are No Bank Account"<<endl;
        break;
    }
    cout<<"Enter The Account Number You want view"<<endl;
    cin>>acc1;
    for(int j=1;j<=ck;j++)
    {
        if(acc1==a[j].acc)
        {
            a[j].withdraw();
            flag=1;
        }
    }

    if(flag==0)
    {
        cout<<"There is no bank account like that!!!!"<<endl;
    }
    break;
}
default:
{
    cout<<"Invalid Input !!!"<<endl;
}
}

}
*****  

*****  

Output:

```

```
Press 1 for Creating New Account
Press 2 for Displaying Account Details
Press 3 Deposit
Press 4 Withdraw
1
Creating New Bank Account
Enter Account Number :
234
Enter Name of The Accountant :
preksha
Press 1 for Saving Account
Press 2 for Current Account
1
Enter Amount :
5000

Press 1 for Creating New Account
Press 2 for Displaying Account Details
Press 3 Deposit
Press 4 Withdraw
2
Enter The Account Number You want view
234
Bank Details Of Account Number :- 234
Accountant Name :- preksha
Type Of Account :- Savings Account
Amount      :- 5000

Press 1 for Creating New Account
Press 2 for Displaying Account Details
Press 3 Deposit
Press 4 Withdraw
3
Enter The Account Number You want view
234
Enter The Amount to Be Deposited :
5000
Amount Successfully Deposited In Account234

Press 1 for Creating New Account
Press 2 for Displaying Account Details
Press 3 Deposit
Press 4 Withdraw
2
Enter The Account Number You want view
234
Bank Details Of Account Number :- 234
Accountant Name :- preksha
Type Of Account :- Savings Account
Amount      :- 10000

Press 1 for Creating New Account
Press 2 for Displaying Account Details
Press 3 Deposit
```

```

Press 4 Withdraw
4
Enter The Account Number You want view
234
Enter The Amount to Be Withdrawn :
5000
Amount Successfully Withdrawn from Account234

Press 1 for Creating New Account
Press 2 for Displaying Account Details
Press 3 Deposit
Press 4 Withdraw

*****
*****Q 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.

*****
*****#include<iostream>
#include<string>
using namespace std;
class student
{
    int rollno;
    string name;
    int age;
    int sub1,sub2,sub3;
public:
student()
{
    sub1 = 25;
    sub2 = 25;
    sub3 = 25;
}
//public:
void get_data()
{
    cout<<endl<<"Enter RollNo : ";
    cin>>rollno;
    cout<<endl<<"Enter Name :";
    cin>>name;
    cout<<endl<<"Enter Age :";
    cin>>age;
    cout<<endl<<"Enter Marks of sub1 : ";
    cin>>sub1;
    cout<<endl<<"Enter Marks of sub2 : ";
    cin>>sub2;
    cout<<endl<<"Enter Marks of sub3 : ";
    cin>>sub3;
}

```


Enter How Many Student Data you want :2

=====
Enter RollNo : 36

Enter Name :preksha

Enter Age :21

Enter Marks of sub1 : 44

Enter Marks of sub2 : 45

Enter Marks of sub3 : 47

=====
=====

Enter RollNo : 32

Enter Name :push

Enter Age :21

Enter Marks of sub1 : 45

```
Enter Marks of sub2 : 43
```

```
Enter Marks of sub3 : 46
```

```
=====
```

```
===== **** DISPLAY DATA **** =====
```

```
Student1:
```

```
Roll No. = 36  
Name = preksha  
Age = 21  
Marks of Subject 1 = 44  
Marks of Subject 2 = 45  
Marks of Subject 3 = 47
```

```
=====
```

```
Student2:
```

```
Roll No. = 32  
Name = pushti  
Age = 21  
Marks of Subject 1 = 45  
Marks of Subject 2 = 43  
Marks of Subject 3 = 46
```

```
=====
```

```
===== Enter Which Roll No. You want to Change : 32
```

```
===== sorry, Your Roll NO. not Matched..!Please Enter Correct Roll No..  
Enter New Marks for Subject 1 :44
```

```
Enter New Marks for Subject 2 :46
```

```
Enter New Marks for Subject 3 :43
```

```
===== **** AFTER MODIFYING DATA **** =====
```

```
Student1:
```

```
Roll No. = 36  
Name = preksha  
Age = 21  
Marks of Subject 1 = 44  
Marks of Subject 2 = 45  
Marks of Subject 3 = 47
```

```
=====
```

```
Student2:
```

```

Roll No. = 32
Name = pushti
Age = 21
Marks of Subject 1 = 44
Marks of Subject 2 = 46
Marks of Subject 3 = 43
=====
Press any key to continue . . .
*****
***** Q 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 than 20.0.

*****
***** #include<iostream>
***** #include<set>
using namespace std;
class rectangle
{
public:
    float width,length;
    rectangle()
    {
        width=1.0;
        length=1.0;
    }
    rectangle(float len,float wid)
    {
        length=len;
        width=wid;
    }
    float setlen(float len)
    {
        if(len>=0.0 && len <=20.0)
        {
            length = len;
        }
        else
        {
            len=1.0;
            length=1.0;
        }
    }
    float setwid(float wid)
    {

```

```

        if(wid>0.0 && wid <20.0)
        {
            width=wid;
        }
        else
        {
            wid=1.0;
            width=1.0;
        }

    }
float getlen()
{
    return length;
}
float getwid()
{
    return width;
}

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

void main()
{
    rectangle a;
    a.setlen(15);
    a.setwid(10);
    cout<<"Length :- "<<a.getlen()<<endl;
    cout<<"Width :- "<<a.getwid()<<endl;
    cout<<"Perimeter :-"<<a.perimeter()<<endl;
}
*****
*****
Output:

```

```

Length :- 15
Width :- 10
Perimeter :-50

```

```

Process returned 0 (0x0)    execution time : 0.192 s
Press any key to continue.

```

```

*****
*****
Q 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.

```

```
*****
*****
#include<iostream>
#include<string>
using namespace std;
static int count;
class supplier
{
public:
    string name,item1[3];
    int sup_no,id;
    void input()
    {
        id=count+1;
        cout<<"Enter The Supplier Name :- "<<endl;
        cin.ignore();
        getline(cin,name);
        cout<<"Enter Supplier No :- "<<endl;
        cin>>sup_no;
        cout<<"Enter Items :- "<<endl;
        for(int i=0;i<3;i++)
        {
            cin.ignore();
            getline(cin,item1[i]);
        }
        cout<<"Auto Mated Id generated :- "<<id<<endl;
    }
    void display()
    {
        cout<<"Supplier Name :- "<<name<<endl;
        cout<<"Supplier Number :- "<<sup_no<<endl;
        for(int i=0;i<3;i++)
        {
            cout<<"Items :- "<<item1[i]<<endl;
        }
    }
    friend class item;
};
class item
{
public:
    supplier* s;
    int ch;
    item()
    {
        cout<<"Enter How Many Supplier Are There :- "<<endl;
        cin>>ch;
        s = new supplier[ch];
    }
    void add()
    {
        if(ch==count)
        {
            cout<<"You cannot add more "<<endl;
        }
    }
}
```

```

        }
    else
    {
        s[count].input();
        count++;
    }

}

void show()
{
    int id,flag=0;
    cout<<"Enter Supplier ID :- "<<endl;
    cin>>id;
    for(int i=0;i<count;i++)
    {
        if(id==s[i].id)
        {
            s[i].display();
            flag =1;
            break;
        }
    }
    if(flag == 0)
    {
        cout<<"Id Not Found"<<endl;
    }
}

};

void main()
{
    item a;
    int ch;
    for(;;)
    {
        cout<<"Press 1 for Adding Supplier\nPress 2 for Display
Supplier"<<endl;
        cin>>ch;
        switch(ch)
        {
        case 1:
            a.add();
            break;
        case 2:
            a.show();
            break;
        default:
            cout<<"Invalid Input"<<endl;
        }
    }
}

```

```
*****  
*****  
*****  
Output:
```

```
Enter How Many Supplier Are There :-
```

```
2
```

```
Press 1 for Adding Supplier
```

```
Press 2 for Display Supplier
```

```
1
```

```
Enter The Supplier Name :-
```

```
preksha
```

```
Enter Supplier No :-
```

```
270
```

```
Enter Items :-
```

```
Fruits
```

```
Vegetables
```

```
Animal Product
```

```
Auto Mated Id generated :- 1
```

```
Press 1 for Adding Supplier
```

```
Press 2 for Display Supplier
```

```
2
```

```
Enter Supplier ID :-
```

```
1
```

```
Supplier Name :- preksha
```

```
Supplier Number :- 270
```

```
Items :- Fruits
```

```
Items :- egetables
```

```
Items :- nimal Product
```

```
Press 1 for Adding Supplier
```

```
Press 2 for Display Supplier
```

```
*****  
*****
```

```
Q 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.
```

```
*****  
*****
```

```
#include<iostream>  
#include<cstring>  
using namespace std;  
static int count;  
class student  
{  
public:  
    int rollno,marks_obtained,max_marks,perc;  
    string name;  
    void input()  
    {  
        cout<<"Enter Your Name :- "<<endl;
```

```

    cin.ignore();
    getline(cin,name);
    cout<<"Enter Your Roll No :- "<<endl;
    cin>>rollno;
    cout<<"Enter Your Marks Obtained :- "<<endl;
    cin>>marks_obtained;
    lump:
        max_marks=0;
        cout<<"Enter Your Max_Marks :- "<<endl;
        cin>>max_marks;
        if(max_marks > 100 && max_marks>=marks_obtained)
        {
            cout<<"Invalid Input cannot be possible !!!!\nTry
Again"<<endl;
            goto lump;
        }
        perc=marks_obtained/5;
    }
    int calc()
    {
        return marks_obtained/5;
    }
    void display()
    {

        cout<<"Your Name :- "<<name<<endl;
        cout<<"Your RollNo :- "<<rollno<<endl;
        cout<<"Marks Obtained :- "<<marks_obtained<<endl;
        cout<<"Max Marks :- "<<max_marks<<endl;
        cout<<"Percentage :- "<<perc<<endl;
    }
    friend class MCA_II;
};

class MCA_II
{
public:
    student *s;
    int ch;
    MCA_II()
    {
        cout<<"Enter How many student :- "<<endl;
        cin>>ch;
        s = new student[ch];
    }
    void add()
    {
        int ck=0;
        if(ch==count)
        {
            cout<<"You cannot add more student"<<endl;
        }
        else
        {
            // lump:

```

```

        s[count].input();
        count++;

    }

}

void delete1()
{
    int roll;
    int flag = 0;
    cout<<"Enter The roll no you want delete : "<<endl;
    cin>>roll;
    for(int i = 0 ; i<count;i++)
    {
        if(roll==s[i].rollno)
        {

            s[i].rollno=0;
            s[i].name='NULL';
            s[i].marks_obtained=0;
            s[i].max_marks=0;
            s[i].perc=0;
            flag=1;
        }
    }
    if(flag==0)
    {
        cout<<"There is no Student roll number like that"<<endl;
        return;
    }

    cout<<"Successfully Deleted !!!"<<endl;
}

void modify()
{
    lump1:
    int roll,ch1,flag=0 ;
    int ck=0;

    cout<<"Enter The Student roll you want to modify :- "<<endl;
    cin>>roll;
    for(int i =0 ; i<count;i++)
    {
        if(roll == s[i].rollno)
        {
            ck=i;
            flag = 1;
            break;
        }
    }
    if(flag==0)
    {
        cout<<"No roll number exist like that !!!"<<endl;
        cout <<"Try Again !!!"<<endl;
    }
}

```

```

        goto lump1;
    }
    cout<<"\nPress 1 for Name\nPress 2 for Roll Number\nPress 3 for
Marks Obtained\nPress 4 for Max Marks"<<endl;
    cin>>ch1;
    switch(ch1)
    {
    case 1:
    {
        string name;
        cout<<"Enter Your Name :- "<<endl;
        cin.ignore();
        getline(cin,name);
        s[ck].name=name;
        cout<<"Name Changed !!"<<endl;
        break;

    }
    case 2:
    {
        lump3:
        int roll,flag=0;

        cout<<"Enter Your New Roll number :- "<<endl;
        cin>>roll;
        for(int i=0;i<count;i++)
        {
            if(roll==s[i].rollno)
            {
                flag=1;
            }
        }
        if(flag == 0)
        {
            cout<<"Roll Number with this Number Already Exist !!!
"<<endl;
            cout<<"Try Again"<<endl;
            goto lump3;
        }
        else
        {
            s[ck].rollno = roll;
            cout<<"Roll Number Changed Successfully !!! "<<endl;
            break;
        }
    }
    case 3:
    {
        int mark,per;
        cout<<"Enter Your New Marks Obtained :- "<<endl;
        cin>>mark;
        s[ck].marks_obtained = mark;
        s[ck].perc=s[ck].calc();
    }
}

```

```

        cout<<"Marks Successfully Changed Your Percentage is
changed Accordingly !!!"<<endl;
            break;
        }
    case 4:
    {
        int mark;
        lump2:
        cout<<"Enter Your New Max Marks :- "<<endl;
        cin>>mark;
        s[ck].max_marks = mark;
        if(s[ck].max_marks > 100 &&
s[ck].max_marks>=s[ck].marks_obtained)
        {
            cout<<"Your Max marks cannot be greater than 100 and
it cannot be greater than your total marks obtained\nTry Again !!!
"<<endl;
            goto lump2;
        }
        cout<<"Max Marks Successfully Changed !!!"<<endl;
        break;
    }
    default :
        cout<<"Invalid Input !!! "<<endl;
        break;
    }
}
void replace1()
{
    lump1:
    int roll,flag =0;
    int ck=0;
    int mark,per;
    string name;
    cout<<"Enter The Student roll you want to replace :- "<<endl;
    cin>>roll;
    for(int i =0 ; i<count;i++)
    {
        if(roll == s[i].rollno)
        {
            flag =1;
            ck=i;
            break;
        }
    }
    if(flag == 0)
    {
        cout<<"There is no Roll number like that !!! "<<endl;
        cout<<"Try Again !!! "<<endl;
        goto lump1;
    }

    cout<<"Enter Name :- "<<endl;
    cin.ignore();
}

```

```

        getline(cin,name);
        s[ck].name=name;
        cout<<"Name Changed Successfully" << endl;

        cout<<"Enter Your New Marks Obtained :- " << endl;
        cin>>mark;
        s[ck].marks_obtained = mark;
        s[ck].perc=s[ck].calc();
        cout<<"Marks Successfully Changed Your Percentage is
changed Accordingly !!!" << endl;
        mark=0;

        lump:
        cout<<"Enter Your New Max Marks :- " << endl;
        cin>>mark;
        s[ck].max_marks = mark;
        if(s[ck].max_marks > 100 &&
s[ck].max_marks>=s[ck].marks_obtained)
        {
            cout<<"Your Max marks cannot be greater than 100 and
it cannot be greater than your total marks obtained\nTry Again !!!
" << endl;
            goto lump;
        }
        else
        {
            cout<<"Your Max Marks Successfully Changed" << endl;
        }

    }

void show()
{
    int roll=0;
    cout<<"Enter Roll Number You Want View :- " << endl;
    cin>>roll;
    for(int i=0;i<count;i++)
    {
        if(roll == s[i].rollno)
        {
            s[i].display();
        }
    }
}

void main()
{
    MCA_II m;
    int ch;
    for(;;)
    {
        cout<<"Press 1 to Add new Student\nPress 2 to Delete a Student
Details\nPress 3 to modify a Student detail\nPress 4 to replace a Student
Details\nPress 5 to Show Student details" << endl;
    }
}

```

```

        cin>>ch;
        switch(ch)
        {
        case 1:
            m.add();
            break;
        case 2:
            m.delete1();
            break;
        case 3:
            m.modify();
            break;
        case 4:
            m.replace1();
            break;
        case 5:
            m.show();
            break;
        default:
            cout<<"Invalid Input !!"<<endl;
            break;
        }
    }
}

*****
*****
Output:

```

```

Enter How many student :-
2
Press 1 to Add new Student
Press 2 to Delete a Student Details
Press 3 to modify a Student detail
Press 4 to replace a Student Details
Press 5 to Show Student details
1
Enter Your Name :-
preksha
Enter Your Roll No :-
36
Enter Your Marks Obtained :-
500
Enter Your Max_Marks :-
100
Press 1 to Add new Student
Press 2 to Delete a Student Details
Press 3 to modify a Student detail
Press 4 to replace a Student Details
Press 5 to Show Student details
1
Enter Your Name :-
prerak

```

```
Enter Your Roll No :-  
26  
Enter Your Marks Obtained :-  
400  
Enter Your Max_Marks :-  
95  
Press 1 to Add new Student  
Press 2 to Delete a Student Details  
Press 3 to modify a Student detail  
Press 4 to replace a Student Details  
Press 5 to Show Student details  
2  
Enter The roll no you want delete :  
26  
Successfully Deleted !!!  
Press 1 to Add new Student  
Press 2 to Delete a Student Details  
Press 3 to modify a Student detail  
Press 4 to replace a Student Details  
Press 5 to Show Student details  
5  
Enter Roll Number You Want View :-  
27  
Your Name :- preksha  
Your RollNo :- 36  
Marks Obtained :- 500  
Max Marks :- 100  
Percentage :- 100  
Press 1 to Add new Student  
Press 2 to Delete a Student Details  
Press 3 to modify a Student detail  
Press 4 to replace a Student Details  
Press 5 to Show Student details  
3  
Enter The Student roll you want to modify :-  
36  
  
Press 1 for Name  
Press 2 for Roll Number  
Press 3 for Marks Obtained  
Press 4 for Max Marks  
3  
Enter Your New Marks Obtained :-  
400  
Marks Successfully Changed Your Percentage is changed Accordingly !!!  
Press 1 to Add new Student  
Press 2 to Delete a Student Details  
Press 3 to modify a Student detail  
Press 4 to replace a Student Details  
Press 5 to Show Student details  
5  
Enter Roll Number You Want View :-  
36  
Your Name :- preksha
```

```

Your RollNo :- 36
Marks Obtained :- 400
Max Marks :- 100
Percentage :- 80
Press 1 to Add new Student
Press 2 to Delete a Student Details
Press 3 to modify a Student detail
Press 4 to replace a Student Details
Press 5 to Show Student details
4
Enter The Student roll you want to replace :-
36
Enter Name :-
preksha sheth
Name Changed Successfully
Enter Your New Marks Obtained :-
500
Marks Successfully Changed Your Percentage is changed Accordingly !!!
Enter Your New Max Marks :-
100
Your Max Marks Successfully Changed
Press 1 to Add new Student
Press 2 to Delete a Student Details
Press 3 to modify a Student detail
Press 4 to replace a Student Details
Press 5 to Show Student details
5
Enter Roll Number You Want View :-
36
Your Name :- preksha sheth
Your RollNo :- 36
Marks Obtained :- 500
Max Marks :- 100
Percentage :- 100
Press 1 to Add new Student
Press 2 to Delete a Student Details
Press 3 to modify a Student detail
Press 4 to replace a Student Details
Press 5 to Show Student details

```

```
*****
*****
Q 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.
```

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

```

static int count;
class car
{
public:
    string name,color;
    int size1,cost,id;
    void input()
    {
        id=count+1;
        cout<<"Enter the Maker Name :- "<<endl;
        cin.ignore();
        getline(cin,name);
        cout<<"Enter Car color :- "<<endl;
        cin.ignore();
        getline(cin,color);
        cout<<"Enter Car Size :- "<<endl;
        cin>>size1;
        cout<<"Enter Car Cost :- "<<endl;
        cin>>cost;
        cout<<"Car Id Generated is :- "<<id<<endl;
    }
    void display()
    {
        cout<<"Car Id :- "<<id<<endl;
        cout<<"Car Maker Name :- "<<name<<endl;
        cout<<"Car Color :- "<<color<<endl;
        cout<<"Car Size :- "<<size1<<endl;
        cout<<"Car Cost :- "<<cost<<endl;
    }
    friend class collection;
};
class collection
{
public:
    car* c;
    int ch;

collection()
{
    cout<<"Enter How many Car details you want to enter :- "<<endl;
    cin>>ch;
    c = new car[ch];
}
void add()
{
    if(count==ch)
    {
        cout<<"You Cannot add more car"<<endl;
    }
    else
    {
        c[count].input();
        count++;
    }
}

```

```

        }
    }
void delete1()
{
    int id1;
    cout<<"Enter ID of the car you want to delete :- "<<endl;
    cin>>id1;
    for(int i=0;i<count;i++)
    {
        if(id1==c[i].id)
        {
            c[i].id=0;
            c[i].name='NULL';
            c[i].color='NULL';
            c[i].size1=0;
            c[i].cost=0;
        }
    }
    cout<<"Successfully Deleted !!! "<<endl;
}
void modify()
{
    int id1,ch,flag=0;
    cout<<"Enter ID of the car you want to modify :- "<<endl;
    cin>>id1;
    for(int i=0;i<count;i++)
    {
        if(id1==c[i].id)
        {
            flag=1;
            cout<<"Press 1 to change the name\nPress 2 to change the
color\nPress 3 to change the size\nPress 4 to change the cost "<<endl;
            cin>>ch;
            switch(ch)
            {
                case 1:
                {
                    string name1;
                    cout<<"Enter The new name of car :- "<<endl;
                    cin.ignore();
                    getline(cin,name1);
                    c[i].name=name1;
                    cout<<"Successfully Changed !!! "<<endl;
                    break;
                }
                case 2:
                {
                    string color1;
                    cout<<"Enter The new color of car :- "<<endl;
                    cin.ignore();
                    getline(cin,color1);
                    c[i].color=color1;
                    cout<<"Successfully Changed !!! "<<endl;
                    break;
                }
            }
        }
    }
}
```

```

        }
    case 3:
    {
        int size1;
        cout<<"Enter the New size of the car :- "<<endl;
        cin>>size1;
        c[i].size1=size1;
        cout<<"Successfully Changed !!! "<<endl;
        break;
    }
    case 4:
    {
        int cost1;
        cout<<"Enter the New cost of the car :- "<<endl;
        cin>>cost1;
        c[i].cost=cost1;
        cout<<"Successfully Changed !!! "<<endl;
        break;
    }
    default:
        cout<<"Invalid input"<<endl;
    }

}

if(flag == 0)
{
    cout<<"Id does not exist !!!"<<endl;
    return;
}
void replace1()
{
    int id1,flag =0,size1,cost;
    string name,color;
    cout<<"Enter The id you want to replace :- "<<endl;
    cin>>id1;
    for(int i=0;i<count;i++)
    {
        if(id1==c[i].id)
        {
            cout<<"Enter Car Name :- "<<endl;
            cin.ignore();
            getline(cin,name);
            c[i].name=name;
            cout<<"Enter Car Color :- "<<endl;
            cin.ignore();
            getline(cin,color);
            c[i].color=color;
            cout<<"Enter Size of Car :- "<<endl;
            cin>>size1;
            c[i].size1=size1;
            cout<<"Enter Cost of Car :- "<<endl;

```

```

        cin>>cost;
        c[i].cost=cost;

    }
}
void show()
{
    int id1;
    cout<<"Enter Car id You want view :- "<<endl;
    cin>>id1;
    for(int i=0;i<count;i++)
    {
        if(id1==c[i].id)
        {
            c[i].display();
        }
    }
};

void main()
{
    int ch;
    collection c;
    for(;;)
    {
        cout<<"Press 1 to add a car\nPress 2 to delete a car\nPress 3 to
modify a car\nPress 4 to replace a car\nPress 5 to Show details of a
car"<<endl;
        cin>>ch;
        switch(ch)
        {
        case 1:
            c.add();
            break;
        case 2:
            c.delete1();
            break;
        case 3:
            c.modify();
            break;
        case 4:
            c.replace1();
            break;
        case 5:
            c.show();
            break;
        default:
            cout<<"Invalid Input !!!"<<endl;
        }
    }
}

```

```
*****
*****
Output:
```

```
Enter How many Car details you want to enter :-
```

```
2
```

```
Press 1 to add a car
```

```
Press 2 to delete a car
```

```
Press 3 to modify a car
```

```
Press 4 to replace a car
```

```
Press 5 to Show details of a car
```

```
1
```

```
Enter the Maker Name :-
```

```
preksha Production
```

```
Enter Car color :-
```

```
Black
```

```
Enter Car Size :-
```

```
100
```

```
Enter Car Cost :-
```

```
10000
```

```
Car Id Generated is :- 1
```

```
Press 1 to add a car
```

```
Press 2 to delete a car
```

```
Press 3 to modify a car
```

```
Press 4 to replace a car
```

```
Press 5 to Show details of a car
```

```
5
```

```
Enter Car id You want view :-
```

```
1
```

```
Car Id :- 1
```

```
Car Maker Name :- preksha Production
```

```
Car Color :- black
```

```
Car Size :- 100
```

```
Car Cost :- 10000
```

```
Press 1 to add a car
```

```
Press 2 to delete a car
```

```
Press 3 to modify a car
```

```
Press 4 to replace a car
```

```
Press 5 to Show details of a car
```

```
1
```

```
Enter the Maker Name :-
```

```
preksha Production
```

```
Enter Car color :-
```

```
yellow
```

```
Enter Car Size :-
```

```
100
```

```
Enter Car Cost :-
```

```
10000
```

```
Car Id Generated is :- 2
```

```
Press 1 to add a car
```

```
Press 2 to delete a car
```

```
Press 3 to modify a car
```

```
Press 4 to replace a car
```

```
Press 5 to Show details of a car
```

```
2
Enter ID of the car you want to delete :-
2
Successfully Deleted !!!
Press 1 to add a car
Press 2 to delete a car
Press 3 to modify a car
Press 4 to replace a car
Press 5 to Show details of a car
3
Enter ID of the car you want to modify :-
1
Press 1 to change the name
Press 2 to change the color
Press 3 to change the size
Press 4 to change the cost
1
Enter The new name of car :-
prerak Production
Successfully Changed !!!
Press 1 to add a car
Press 2 to delete a car
Press 3 to modify a car
Press 4 to replace a car
Press 5 to Show details of a car
5
Enter Car id You want view :-
1
Car Id :- 1
Car Maker Name :- prerak Production
Car Color :- lack
Car Size :- 100
Car Cost :- 10000
Press 1 to add a car
Press 2 to delete a car
Press 3 to modify a car
Press 4 to replace a car
Press 5 to Show details of a car
4
Enter The id you want to replace :-
1
Enter Car Name :-
preksha Production
Enter Car Color :-
Black
Enter Size of Car :-
100
Enter Cost of Car :-
100000
Press 1 to add a car
Press 2 to delete a car
Press 3 to modify a car
Press 4 to replace a car
Press 5 to Show details of a car
```

```

5
Enter Car id You want view :-
1
Car Id :- 1
Car Maker Name :- preksha Production
Car Color :- lack
Car Size :- 100
Car Cost :- 100000
Press 1 to add a car
Press 2 to delete a car
Press 3 to modify a car
Press 4 to replace a car
Press 5 to Show details of a car

*****
*****Q(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.

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

using namespace std;

class Employee
{
    string name,dept;
    int id;
public:
    string designation;
    void input()
    {
        cout<<"\n-----\n";
        cout<<"Enter Employee ID: ";
        cin>>id;
        cout<<"Enter Employee Name: ";
        cin>>name;
        cout<<"Enter Department of employee: ";
        cin>>dept;
        cout<<"Enter Designation Of Employee:";
        cin>>designation;
    }

    void output()
    {

```

```

        cout<<"\n-----\n";
        cout<<"Employee ID: "<<id<<endl;
        cout<<"Employee Name: "<<name<<endl;
        cout<<"Employee Department: "<<dept<<endl;
        cout<<"Employee Designation: "<<designation<<endl;

    }

};

class EmpCollection
{
    Employee e[10];
    int limit;
public:
    void input()
    {
        cout<<"Enter number of Employees: ";
        cin>>limit;
        for(int i=0;i<limit;i++)
            e[i].input();
    }

    void output()
    {
        cout<<"\n=====\\n";
        for(int i=0;i<limit;i++)
        {
            e[i].output();
        }
    }

    friend EmpCollection GetSubordinates(EmpCollection);
};

EmpCollection GetSubordinates(EmpCollection e1)
{
    int i,j=0;
    EmpCollection e2;
    for(i=0;i<e1.limit;i++)
    {
        if(e1.e[i].designation != "Manager")
        {
            e2.e[j]=e1.e[i];
            j++;
        }
    }
    e2.limit=j;
    return e2;
}

void main()
{
    EmpCollection e1,e3;
    e1.input();
}

```

```
    e1.output();
    e3=GetSubordinates(e1);
    cout<<"\n-----Details of subordinates of
Manager :"<<endl;
    e3.output();
    getch();
}

*****
*****
output:
ENTER number of Employees: 5

-----
Enter Employee ID: 36
Enter Employee Name: preksha
Enter Department of employee: IT
Enter Designation Of Employee:Supervisor

-----
Enter Employee ID: 27
Enter Employee Name: vraj
Enter Department of employee: IT
Enter Designation Of Employee:Manager

-----
Enter Employee ID: 26
Enter Employee Name: rutwik
Enter Department of employee: IT
Enter Designation Of Employee:Assistant

-----
Enter Employee ID: 25
Enter Employee Name: Maahi
Enter Department of employee: IT
Enter Designation Of Employee:Manager

-----
Enter Employee ID: 31
Enter Employee Name: prerak
Enter Department of employee: IT
Enter Designation Of Employee:Peon

=====
-----
Employee ID: 36
Employee Name: preksha
Employee Department: IT
Employee Designation: Supervisor

-----
Employee ID: 27
Employee Name: vraj
Employee Department: IT
```

Employee Designation: Manager

Employee ID: 26
Employee Name: rutwik
Employee Department: IT
Employee Designation: Assistant

Employee ID: 25
Employee Name: Maahi
Employee Department: IT
Employee Designation: Manager

Employee ID: 31
Employee Name: prerak
Employee Department: IT
Employee Designation: Peon

-----Details of subordinates of Manager :

Employee ID: 36
Employee Name: preksha
Employee Department: IT
Employee Designation: Supervisor

Employee ID: 26
Employee Name: rutwik
Employee Department: IT
Employee Designation: Assistant

Employee ID: 31
Employee Name: prerak
Employee Department: IT
Employee Designation: Peon

Q 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 "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.

```
*****  
*****
```

```
#include<iostream>
#include<string.h>
#include<stdio.h>
#include<vector>
using namespace std;
static int count;
static int trans;
class book
{
public:
    string fname, t_name, p_name;
    int price, stock, id;
    void input()
    {
        id = count + 1;
        cout << "Enter The Author Name :- " << endl;
        cin.ignore();
        getline(cin, fname);
        cout << "Enter The Title Name :- " << endl;
        getline(cin, t_name);
        cout << "Enter The Price of Book :- " << endl;
        cin >> price;
        cout << "Enter The Publisher Name :- " << endl;
        cin.ignore();
        getline(cin, p_name);
        cout << "Enter The Stock of The Book :- " << endl;
        cin >> stock;
        cout << "ID of the Book Generated :- " << id << endl;
    }
    void display()
    {
        cout << "ID of The Book :- " << id << endl;
        cout << "Book Name :- " << t_name << endl;
        cout << "Author Name :- " << fname << endl;
        cout << "Price of Book :- " << price << endl;
        cout << "Publisher of Book :- " << p_name << endl;
        cout << "Stock of Book :- " << stock << endl;
    }
}
```

```

        }
        friend class person;
    };
class person
{
public:
book* b;
int ch;
int t[20];
vector<string> transl;
person()
{
    cout<<"How Many Book Are There "<<endl;
    cin>>ch;
    b = new book[ch];
}
void add()
{
    if(ch == count)
    {
        cout<<"You cannot add more number of books"<<endl;
    }
    else
    {
        b[count].input();
        trans++;
        transl.push_back("--Book Added of Name :- ");
        transl.push_back(b[count].t_name);
        count++;
    }
}
void search1()
{
    string t_name,a_name;
    int flag =0,cp,cost1,id,res1=1,res2;
    cout<<"Enter The Title name :- "<<endl;
    cin.ignore();
    getline(cin,t_name);

    for(int i=0;i<count;i++)
    {
        if(t_name == b[i].t_name)
        {
            res1 = 0;
        }
        if(res1 == 0 )
        {
            flag = 1;
            trans++;
            transl.push_back("--Book Found of Name :- ");
            transl.push_back(b[i].t_name);
        }
    }
}

```

```

        res2=i;
        if(b[i].stock==0)
        {
            flag = 2;
            break;
        }
        b[i].display();
        cout<<"Enter The number of Copies :- "<<endl;
        cin>>cp;
        if(b[i].stock<cp)
        {
            flag = 3;
            break;
        }
        else
        {
            cost1=b[i].price * cp;
            b[i].stock = b[i].stock - cp;
            cout<<"Total Cost :- "<<cost1<<endl;
            cout<<"Number Copies ordered :- "<<cp<<endl;
            cout<<"Number copies left :- "<<b[i].stock<<endl;
            trans++;
            transl.push_back("--Books Purchased of Name :- ");
            transl.push_back(b[i].t_name);
            break;
        }
    }
    if(flag == 0)
    {
        cout<<"There is no Book Like That !!!"<<endl;
        trans++;
        transl.push_back("--no Book Found of Name :- ");
        transl.push_back(t_name);
        return;
    }
    else if(flag == 2)
    {
        cout<<"There is no stock available !!!"<<endl;
        trans++;
        transl.push_back("--Stock Not Available of Book :- ");
        transl.push_back(b[res2].t_name);
        return;
    }
    else if(flag == 3)
    {
        cout<<"Required copies not in Stock !!!"<<endl;
        trans++;
        transl.push_back("--Required Stock Not Available of Book :- ");
        transl.push_back(b[res2].t_name);
        return;
    }
}

```

```

void show()
{
    int id,flag = 0;
    cout<<"Enter The Book Id :- "<<endl;
    cin>>id;
    for(int i=0;i<count;i++)
    {
        if(id==b[i].id)
        {
            b[i].display();
            flag = 1;
        }
    }
    if(flag == 0)
    {
        cout<<"There is no Book id like that "<<endl;
        return;
    }
}
private:
void update(void)
{
    int id,flag=0;
    cout<<"Enter the Book Id :- "<<endl;
    cin>>id;
    for(int i =0;i<count;i++)
    {
        if(id==b[i].id)
        {
            flag = 1;
            trans++;
            transl.push_back("--Book Found of Name For Updation of
Price");
            transl.push_back(b[i].t_name);
            id=0;
            cout<<"Enter The New Price for Book Id"<<b[i].id<<" :-
"<<endl;
            cin>>id;
            b[i].price=id;
            trans++;
            transl.push_back("--price Updated of Book Name");
            transl.push_back(b[i].t_name);
            break;
        }
    }
    if(flag == 0)
    {
        cout<<"Book Not Found !!"<<endl;
        trans++;
        transl.push_back("--Book Not Found For Price Updation");
    }
}
public:
    void update1()

```

```

    {
        update();
    }
void transaction()
{
    cout<<"--Total Transaction Number :- "<<trans<<endl;
    for (int i=0;i<trans1.size();i++)
    {
        cout<<trans1[i]<<endl;
    }
}
};

void main()
{
    person p;
    int ch;
    for(;;)
    {
        cout<<"Press 1 to Add a Book\nPress 2 to Place an Order\nPress 3 to
show Book Details\nPress 4 to Show Transaction Details\nPress 5 to Update
Price of Book"<<endl;
        cin>>ch;
        switch(ch)
        {
        case 1:
            p.add();
            break;
        case 2:
            p.search1();
            break;
        case 3:
            p.show();
            break;
        case 4:
            p.transaction();
            break;
        case 5:
            p.update1();
            break;
        default:
            cout<<"Invalid Input "<<endl;
        }
    }
}
}

*****
*****
```

Output:

```

How Many Book Are There
2
Press 1 to Add a Book
Press 2 to Place an Order
Press 3 to show Book Details
Press 4 to Show Transaction Details
```

```
Press 5 to Update Price of Book
1
Enter The Author Name :-
balaguru
Enter The Title Name :-
c++
Enter The Price of Book :-
1000
Enter The Publisher Name :-
ansi Publication
Enter The Stock of The Book :-
5
ID of the Book Generated :- 1
Press 1 to Add a Book
Press 2 to Place an Order
Press 3 to show Book Details
Press 4 to Show Transaction Details
Press 5 to Update Price of Book
2
Enter The Title name :-
c++
ID of The Book :- 1
Book Name :- c++
Author Name :- balaguru
Price of Book :-1000
Publisher of Book :-ansi Publication
Stock of Book :-5
Enter The number of Copies :-
4
Total Cost :- 4000
Number Copies ordered :- 4
Number copies left :- 1
Press 1 to Add a Book
Press 2 to Place an Order
Press 3 to show Book Details
Press 4 to Show Transaction Details
Press 5 to Update Price of Book
3
Enter The Book Id :-
1
ID of The Book :- 1
Book Name :- c++
Author Name :- balaguru
Price of Book :-1000
Publisher of Book :-ansi Publication
Stock of Book :-1
Press 1 to Add a Book
Press 2 to Place an Order
Press 3 to show Book Details
Press 4 to Show Transaction Details
Press 5 to Update Price of Book
5
Enter the Book Id :-
1
```

```

Enter The New Price for Book Id1 :-
2000
Press 1 to Add a Book
Press 2 to Place an Order
Press 3 to show Book Details
Press 4 to Show Transaction Details
Press 5 to Update Price of Book
4
--Total Transaction Number :- 5
--Book Added of Name :-
c++
--Book Found of Name :-
c++
--Books Purchased of Name  :-
c++
--Book Found of Name For Updation of Price
c++
--price Updated of Book Name
c++
Press 1 to Add a Book
Press 2 to Place an Order
Press 3 to show Book Details
Press 4 to Show Transaction Details
Press 5 to Update Price of Book

```

```

*****
*****
Q 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).

```

```

*****
*****
#include<iostream>
using namespace std;
static int count;
class search
{
public:
int* a;
int x,ch;
search()
{
    cout<<"How Many Number You want in array :- ";
    cin>>ch;
    a = new int[ch];
}
void input()
{
    if(ch==count)
    {
        cout<<"You Cannot Add More Number "<<endl;
    }
}
```

```

        return;
    }
    else
    {
        cout<<"Enter The Number You Want to add :- "<<endl;
        cin>>x;
        a[count]=x;
        count++;
        cout<<"Successfully Added"<<endl;
    }
}
void output()
{
    cout<<"Elements in The Array :- "<<endl;
    for(int i =0;i<count;i++)
    {
        cout<<a[i]<<endl;
    }
}
void search1(int position)
{
    if(position>count)
    {
        cout<<"Position exceeds the array ";
        return;
    }
    else
    {
        cout<<"Value on Position "<<position<<" is :-
"<<a[position]<<endl;
    }
}

void add(int value)
{
    if(ch==count)
    {
        cout<<"You Cannot Add More Number"<<endl;
        return;
    }
    else
    {
        a[count]=value;
        count++;
    }
}
};

void main()
{
    search s;
    int ch;

    for(;;)
    {

```

```

        cout<<"Press 1 for Output\nPress 2 for Search\nPress 3 for add a
value\nPress 4 for input"<<endl;
        cin>>ch;
        switch(ch)
        {
        case 1:
            s.output();
            break;
        case 2:
            {
            int pos;
            //pos = new int;
            cout<<"Enter The Position You Want to Search :-"<<endl;
            cin>>pos;
            s.search1(pos);
            // delete pos;
            break;
            }
        case 3:
            {
            int val;
            // val = new int;
            cout<<"Enter The Value You Want to Add :-"<<endl;
            cin>>val;
            s.add(val);
            // delete val;
            break;
            }
        case 4:
            {
            s.input();
            break;
            }
        default:
            {
            cout<<"Invalid Input !!!"<<endl;
            }
        }
    }
}

```


Output:

How Many Number You want in array :- 5

Press 1 for Output

Press 2 for Search

Press 3 for add a value

Press 4 for input

4

Enter The Number You Want to add :-

4

Successfully Added

Press 1 for Output

```
Press 2 for Search
Press 3 for add a value
Press 4 for input
4
Enter The Number You Want to add :-
3
Successfully Added
Press 1 for Output
Press 2 for Search
Press 3 for add a value
Press 4 for input
4
Enter The Number You Want to add :-
2
Successfully Added
Press 1 for Output
Press 2 for Search
Press 3 for add a value
Press 4 for input
4
Enter The Number You Want to add :-
1
Successfully Added
Press 1 for Output
Press 2 for Search
Press 3 for add a value
Press 4 for input
1
Elements in The Array :-
4
3
2
1
Press 1 for Output
Press 2 for Search
Press 3 for add a value
Press 4 for input
2
Enter The Position You Want to Search :-
3
Value on Position 3 is :- 1
Press 1 for Output
Press 2 for Search
Press 3 for add a value
Press 4 for input
3
Enter The Value You Want to Add :-
5
Press 1 for Output
Press 2 for Search
Press 3 for add a value
Press 4 for input
1
Elements in The Array :-
```

```

4
3
2
1
5
Press 1 for Output
Press 2 for Search
Press 3 for add a value
Press 4 for input

*****
***** ASSIGNMENT - 2 *****

Q 1 : WAP to use binary operator + add two object of class Numbers
having num1
and num2 as its data members and display result.

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

class overloads
{
    private :
        int a,b;

    public :

        void setdata(int x,int y)
        {
            a = x;
            b = y;
        }

        void operator +(overloads ob2)
        {
            cout<<"Addition of "<<a<<" and "<<ob2.a<<" is :
"<<a+ob2.a;
            cout<<"\n\nAddition of "<<b<<" and "<<ob2.b<<" is :
"<<b+ob2.b;
        }
};

void main()
{
    overloads ob1,ob2;
    ob1.setdata(50,50);
    ob2.setdata(20,30);
    ob1 + ob2;
}

```

```
*****
*****
Output:
```

```
Addition of 50 and 20 is : 70
```

```
Addition of 50 and 30 is : 80Press any key to continue . . .
```

```
*****
*****
```

```
Q 2 : WAP to overload operator * which multiply a number to each element  
of an
```

```
array within a class arrayContainer and display the result.
```

```
*****
*****
```

```
*****
*****
```

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

```
class arraycontainer  
{  
    private :  
        int arr[100],n;  
  
    public :  
  
        void getdata()  
        {  
            cout<<"How Many Elements you want to enter in Array List :";  
            cin>>n;  
            for(int i = 0; i < n; i++)  
            {  
                cout<<"\nEnter Number "<<i+1<<" : ";  
                cin>>arr[i];  
            }  
        }  
        void print()  
        {  
            for(int i = 0; i < n; i++)  
            {  
                cout<<"\nNumber "<<i+1<<" = "<<arr[i];  
            }  
        }  
        void operator *(int no)  
        {  
            for(int i = 0; i < n; i++)  
            {  
                cout<<"\nNumber "<<i+1<<" is "<<arr[i]*no;  
            }  
        }  
};  
void main()  
{  
    int no;
```

```

arraycontainer ob1;
ob1.getdata();
//ob2.getdata();
ob1.print();
cout<<"\nEnter the Number you want to be Multiplied :";
cin>>no;
cout<<"\n=====\\"n";
cout<<"\t\tAfter Multiplied no. with array ";
cout<<"\n=====\\"n";
ob1*no;
//ob2.print();
}
*****
*****Output:

```

How Many Elements you want to enter in Array List :3

Enter Number 1 : 10

Enter Number 2 : 15

Enter Number 3 : 20

Number 1 = 10

Number 2 = 15

Number 3 = 20

Enter the Number you want to be Multiplied :5

===== After Multiplied no. with array =====

Number 1 is 50

Number 2 is 75

Number 3 is 100Press any key to continue . . .

Q 3 :) WAP to Overload the *, +, -, ==, != and = operators for the complex class.

***** /

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

class complex

{

 int x;

 int y;

public :

 void input()

```

{
    cout<<"Enter real Number : ";
    cin>>x;
    cout<<"Enter Img Number : ";
    cin>>y;
}
void disp()
{
    cout<<"\n\nReal Number is "<<x;
    cout<<"\nImg Number is "<<y;
}
complex operator + (complex obj3)
{
    complex temp;
    temp.x = x + obj3.x;
    temp.y = y + obj3.y;
    return(temp);
}

complex operator * (complex obj3)
{
    complex temp;
    temp.x = x * obj3.x;
    temp.y = y * obj3.y;
    return(temp);
}
complex operator - (complex obj3)
{
    complex temp;
    temp.x = x - obj3.x;
    temp.y = y - obj3.y;
    return(temp);
}
bool operator ==(complex obj3)
{
    if((x == obj3.x) && (y == obj3.y))
    {
        return true;
    }
    else
    {
        return false;
    }
}

bool operator !=(complex obj3)
{
    if((x != obj3.x) && (y != obj3.y))
    {
        return true;
    }
    else
    {
        return false;
    }
}

```

```

        }
    }
};

void main()
{
    complex obj1,obj2,obj3;
    obj1.input();
    obj2.input();
    obj1.disp();
    obj2.disp();
    cout<<"\n\nADDITION\n";
    obj3 = obj1 + obj2;
    obj3.disp();
    cout<<"\n\nMULTIPLICATION\n";
    obj3 = obj1 * obj2;
    obj3.disp();
    cout<<"\n\nSUBTRACTION\n";
    obj3 = obj1 - obj2;
    obj3.disp();
    cout<<"\n\n == and != Operator\n";
    if(obj1 == obj2)
    {
        cout<<"\nboth objects are same";
    }
    else if(obj1 != obj2)
    {
        cout<<"\nboth objects are not same";
    }
    else
    {
        cout<<"none";
    }
}
*****
*****
*****
Output:
```

```

Enter real Number : 36
Enter Img Number : 30
Enter real Number : 30
Enter Img Number : 36
```

```

Real Number is 36
Img Number is 30
```

```

Real Number is 30
Img Number is 36
```

ADDITION

```

Real Number is 66
```

Img Number is 66

MUTIPLICATION

Real Number is 1080
Img Number is 1080

SUBTRACTION

Real Number is 6
Img Number is -6

== and != Operator

both objects are not samePress any key to continue . . .

Q 4 : WAP to define an object m1 of matrix class, use m1<<cout.

```
*****  
*****  
*****  
*****  
#include<iostream>  
using namespace std;  
class matrix  
{  
    int m[10][10],r,c;  
public:  
    void input()  
    {  
        int i,j;  
        cout<<"\n Enter number of rows:";  
        cin>>r;  
        cout<<"\n Enter number of cols:";  
        cin>>c;  
        for(i=0;i<r;i++)  
        {  
            for(j=0;j<c;j++)  
            {  
                cout<<"\n Enter ["<<i<<"] "<<" ["<<j<<"] :" ;  
                cin>>m[i][j];  
            }  
        }  
    }  
    //friend ostream & operator <<(ostream &dout,matrix temp);  
    friend ostream & operator <<(matrix temp,ostream &dout);  
};  
ostream & operator <<(matrix temp,ostream &dout)  
{  
    int i,j;  
    for(i=0;i<temp.r;i++)  
    {
```

```

        for(j=0;j<temp.c;j++)
    {
        dout<<"\n Enter ["<<i<<"] "<<" ["<<j<<"] :"<<temp.m[i][j];
    }
    return (dout);
}
void main()
{
    matrix m1;
    m1.input();
    m1<<cout;
}

```

```
*****
*****
Output:
```

Enter number of rows:3

Enter number of cols:3

Enter [0] [0]:1

Enter [0] [1]:2

Enter [0] [2]:3

Enter [1] [0]:4

Enter [1] [1]:5

Enter [1] [2]:6

Enter [2] [0]:7

Enter [2] [1]:8

Enter [2] [2]:9

Enter [0] [0]:1

Enter [0] [1]:2

Enter [0] [2]:3

Enter [1] [0]:4

Enter [1] [1]:5

Enter [1] [2]:6

Enter [2] [0]:7

Enter [2] [1]:8

Enter [2] [2]:9Press any key to continue . . .

```
*****
*****
Q 5 : WAP to define a matrix class and overload the * operator to
multiply a number with matrix (Example: 5*Matrix should be possible).
```

```
*****
*****
#include<iostream>
using namespace std;
class matrix
{
    int m[10][10],r,c;
public:
    void input()
    {
        int i,j;
        cout<<"\n Enter number of rows:";
        cin>>r;
        cout<<"\n Enter number of cols:";
        cin>>c;
        for(i=0;i<r;i++)
        {
            for(j=0;j<c;j++)
            {
                cout<<"\n Enter ["<<i<<"] "<<" ["<<j<<"] :" ;
                cin>>m[i][j];
            }
        }
    }
    void display()
    {
        cout<<"\n\t\t*** DISPLAY ***\n\n";
        int i,j;
        for(i=0;i<r;i++)
        {
            for(j=0;j<c;j++)
            {
                cout<<m[i][j]<<"\t";
            }
            cout<<"\n\n";
        }
    }
    friend matrix operator *(matrix obj1,int n);
};
matrix operator *(matrix obj1,int n)
{
    int i,j;
    matrix temp;
    temp.r=obj1.r;
    temp.c=obj1.c;
    for(i=0;i<obj1.r;i++)
    {
        for(j=0;j<obj1.c;j++)
        {
            temp.m[i][j]=obj1.m[i][j]*n;
        }
    }
    return temp;
}
```

```
}

void main()
{
    int num;
    matrix m1,rslt;
    m1.input();
    m1.display();
    cout<<"\n Enter Number to be Multiplied: ";
    cin>>num;
    rslt=m1*num;
    rslt.display();
}
```

```
*****
*****
Output:
```

```
Enter number of rows:2
```

```
Enter number of cols:3
```

```
Enter [0] [0]:1
```

```
Enter [0] [1]:2
```

```
Enter [0] [2]:3
```

```
Enter [1] [0]:4
```

```
Enter [1] [1]:5
```

```
Enter [1] [2]:7
```

```
*** DISPLAY ***
```

```
1      2      3
```

```
4      5      7
```

```
Enter Number to be Multiplied: 2
```

```
*** DISPLAY ***
```

```
2      4      6
```

```
8      10     14
```

```
Press any key to continue . . .
```

```
*****
***** Q(6): WAP to define a class Date with properties int month; int day;
int year;      overload the following operators.

5.1) + operator [a+b] (a is of date type and b is an integer), use
the assumption that all years a have 360 days and months 30 days
5.2) - operator [a-b(same as above)]
5.3) = operator
5.4) <, <=, >, >=
5.5) ++, --[post and pre both]

*****
#include<iostream>
using namespace std;
class date
{
    int month,day,year;
public:
    void input()
    {
        cout<<"\n Enter Date in proper format(dd mm yyyy):";
        cin>>day>>month>>year;
    }
    void display()
    {
        cout<<"\n Date is:"<<day<<"/"<<month<<"/"<<year;
    }
    const date operator =(const date obj2)
    {
        day=obj2.day;
        month=obj2.month;
        year=obj2.year;
        return *this;
    }
    bool operator >(date temp)
    {
        if( (day > temp.day) && (month>temp.month) &&
(year>temp.year) )
            return true;
        else
            return false;
    }
    bool operator >=(date temp)
    {
        if( (day >= temp.day) && (month>=temp.month) &&
(year>=temp.year) )
            return true;
        else
            return false;
    }
    bool operator <(date temp)
    {
```

```

        if( (day< temp.day) && (month<temp.month) && (year<temp.year)
    )
            return true;
        else
            return false;
    }
bool operator <=(date temp)
{
    if( (day <= temp.day) && (month<=temp.month) &&
(year<=temp.year) )
        return true;
    else
        return false;
}
date operator ++()
{
    ++day;
    if(day==31)
    {
        day=1;
        ++month;
    }
    if(month==13)
    {
        month=1;
        ++year;
    }
    return *this;
}
date operator ++(int dummy)
{
    ++day;
    if(day==31)
    {
        day=1;
        ++month;
    }
    if(month==13)
    {
        month=1;
        ++year;
    }
    return *this;
}
date operator --()
{
    --day;
    if(day==0)
    {
        day=30;
        --month;
    }
    if(month==0)
    {

```

```

        month=12;
        --year;
    }
    return *this;
}
date operator --(int dummy)
{
    --day;
    if(day==0)
    {
        day=30;
        --month;
    }
    if(month==0)
    {
        month=12;
        --year;
    }
    return *this;
}

friend date operator +(date obj2,int n);
friend date operator -(date obj2,int n);
};

date operator +(date obj2,int n)
{
    date temp;
    temp.year=obj2.year;
    temp.month=obj2.month;
    temp.day=(obj2.day)+n;
    while(temp.day>30)
    {
        temp.day=temp.day-30;
        temp.month++;
        if(temp.month>12)
        {
            temp.month=1;
            temp.year++;
        }
    }
    return temp;
}

date operator -(date obj2,int n)
{
    int flag=0;
    date temp; //32
    temp.year=obj2.year; //1999
    temp.month=obj2.month; //10
    temp.day=obj2.day; //30
    if(obj2.day>n)
        temp.day=(obj2.day)-n; //3-2=1
    else
    {

```

```

    if(n>30)
    {
        while(n>=30)
        {
            if(temp.day>30)
                temp.day=temp.day-30; //3-30

            if(temp.day==0)
                temp.day=30;

            temp.month--; //4

            if(temp.month==0)
            {
                temp.month=12;
                temp.year--;
            }
            n-=30; //1
        }
        if(n!=0) //if 2,3,4 etc days are remaining from 30,31,32
etc them it will come here
        {
            temp.day=temp.day-n; //3-1=2
            if(temp.day==0)
            {
                temp.day=30;
                temp.month--; //9
            }
            if(temp.month==0)
            {
                temp.month=12;
                temp.year--;
            }
        }
    }
else
{
    temp.day=temp.day-n;
    if(temp.day<0)
        temp.day=30+temp.day;
    else if(temp.day==0)
        temp.day=30;
    temp.month--; //9
    if(temp.month==0)
    {
        temp.month=12;
        temp.year--;
    }
}
return temp;
}
void main()
{

```

```

int num;
//int ch=1;

date obj1,obj2,rslt;
cout<<"\n Object1 input";
obj1.input();
cout<<"\n Object1 display";
obj1.display();
cout<<"\n Enter days to be added:";
cin>>num;
cout<<"\n-----After adding the days to date-----"
";
rslt=obj1+num;
rslt.display();
/*while(ch!=0)
{*/
    cout<<"\n Enter days to be subtracted:";
    cin>>num;
    cout<<"\n-----After subtracting the days to date----"
-----";
    rslt=obj1-num;
    rslt.display();
    /*printf("\n Enter ch(Press 0 to exit or else any number to
continue):::");
    scanf("%d",&ch);
}*/
printf("\n");
cout<<"\n Object2 input";
obj2.input();
cout<<"\n Object2 display";
obj2.display();
obj2=obj1;
cout<<"\n Object1 data is copied into object2\nAnd displaying
Object2 data";
obj2.display();

cout<<"\n";
cout<<"\n Please enter Object2 input again";
obj2.input();
cout<<"\n Object2 display";
obj2.display();

cout<<"\n\n\n";
if(obj1>obj2)
    cout<<"\n Object1 is greater than object2";
else if(obj2>obj1)
    cout<<"\n Object2 is greater than object1";
if(obj1<obj2)
    cout<<"\n Object1 is less than object2";
else if(obj2<obj1)
    cout<<"\n Object2 is less than object1";
if(obj1>=obj2)
    cout<<"\n Object1 is greater than equal to object2";
else if(obj2>=obj1)

```

```

        cout<<"\n Object2 is greater than equal to object1";
if(obj1<=obj2)
    cout<<"\n Object1 is less than equal to object2";
else if(obj2<=obj1)
    cout<<"\n Object2 is less than equal to object1";
cout<<"\n\n";
cout<<"\n wihtout Pre Incremented object1 data";
obj1.display();
cout<<"\n Pre Incremented object1 data";
++obj1;
obj1.display();
printf("\n");
cout<<"\n Post Incremented object1 data";
obj1++;
obj1.display();
cout<<"\n";
cout<<"\n wihtout Decrementated object2 data";
obj2.display();
cout<<"\n Pre Decrementated object2 data";
--obj2;
obj2.display();
printf("\n");
cout<<"\n Post Decrementated object2 data";
obj2--;
obj2.display();
printf("\n");
}

```

```
*****
*****
```

Output:

Object1 input
Enter Date in proper format(dd mm yyyy) :12 3 1999

Object1 display
Date is:12/3/1999
Enter days to be added:120

-----After adding the days to date-----
Date is:12/7/1999
Enter days to be subtracted:120

-----After subtracting the days to date-----
Date is:12/11/1998

Object2 input
Enter Date in proper format(dd mm yyyy) :30 12 2008

Object2 display
Date is:30/12/2008
Object1 data is copied into object2
And displaying Object2 data
Date is:12/3/1999

```
Please enter Object2 input again  
Enter Date in proper format(dd mm yyyy):30 12 2007
```

```
Object2 display  
Date is:30/12/2007
```

```
Object2 is greater than object1  
Object1 is less than object2  
Object2 is greater than equal to object1  
Object1 is less than equal to object2
```

```
without Pre Incremented object1 data  
Date is:12/3/1999  
Pre Incremented object1 data  
Date is:13/3/1999
```

```
Post Incremented object1 data  
Date is:14/3/1999
```

```
without Decrement object2 data  
Date is:30/12/2007  
Pre Decrement object2 data  
Date is:29/12/2007
```

```
Post Decrement object2 data  
Date is:28/12/2007  
Press any key to continue . . .
```

```
*****  
*****
```

```
Q(7): WAP to define a class Time with properties int hour;  
int minute; int second; overload the following operators.  
6.1) + operator [a+b] (a is of time type and b is an integer)  
6.2) - operator [a-b(same as above)]  
6.3) = operator  
6.4) <, <=, >, >=  
6.5) ++, --[post and pre both]
```

```
*****  
*****
```

```
#include<iostream>  
using namespace std;  
class time  
{  
    int hh,mm,ss;  
public:  
    void input()  
    {  
        cout<<"\n Enter Time in proper format(hh mm ss):";  
        cin>>hh>>mm>>ss;
```

```

}

void display()
{
    cout<<"\n time is:"<<hh<<":"<<mm<<":"<<ss<<"\n";
}
const time operator =(const time obj2)
{
    hh=obj2.hh;
    mm=obj2.mm;
    ss=obj2.ss;
    return *this;
}
bool operator >(time temp)
{
    if( (hh > temp.hh) && (mm>temp.mm) && (ss>temp.ss) )
        return true;
    else
        return false;
}
bool operator >=(time temp)
{
    if( (hh >= temp.hh) && (mm>=temp.mm) && (ss>=temp.ss) )
        return true;
    else
        return false;
}
bool operator <(time temp)
{
    if( (hh < temp.hh) && (mm<temp.mm) && (ss<temp.ss) )
        return true;
    else
        return false;
}
bool operator <=(time temp)
{
    if( (hh<=temp.hh) && (mm<=temp.mm) && (ss<=temp.ss) )
        return true;
    else
        return false;
}

time operator ++()
{
    ++mm;
    if(mm==60)
    {
        mm=0;
        ++hh;
    }
    if(hh==13)
        hh=1;
    return *this;
}
time operator ++(int dummy)

```

```

{
    ++mm;
    if (mm==60)
    {
        mm=0;
        ++hh;
    }
    if (hh==13)
        hh=1;
    return *this;
}
time operator --()
{
    --mm;
    if (mm===-1)
    {
        mm=59;
        --hh;
    }
    if (hh==0)
        hh=12;
    return *this;
}
time operator --(int dummy)
{
    --mm;
    if (mm===-1)
    {
        mm=59;
        --hh;
    }
    if (hh==0)
        hh=12;
    return *this;
}
friend time operator +(time obj2,int n);
friend time operator -(time obj2,int n);
};
time operator +(time obj2,int n)
{
    time temp;
    temp.hh=obj2.hh;
    temp.mm=obj2.mm+n; //50+9=59
    temp.ss=obj2.ss;
    while(temp.mm>=60)
    {
        temp.mm=temp.mm-60;
        temp.hh++;
        if(temp.hh==13)
            temp.hh=1;
    }
    return temp;
}
time operator -(time obj2,int n)

```

```

{
    time temp;
    temp.hh=obj2.hh;
    if(obj2.mm>=n)
        temp.mm=obj2.mm-n;
    temp.ss=obj2.ss;
    if(obj2.mm<n)
        //if value of subtract_minutes>minute then it will go to - if we
        will directly      subtract it from minutes.so it must not become - and
        minute will be      subtracted in this loop.
    {
        //first of all we will make minute to 0.
        //eg time->12:10 n->123.
        //Here n>mm so here time changes as follows.

        temp.mm=0;

        //Now time will remain same as time->12:0

        //now we are subtracting minute till 0(till 0 how many minute
        subtracted) and also decrementing these number of minute
        temp.minute.

        n-=obj2.mm; //n=123-10=113

        //now 113 minute are remaining to subtract.

        //if remaining subtraction minutes<60 then it will not go in
        and will directly subtract from temp.minute

        loop while(n>=60)
        {

            //in this loop we will subtract 60 minute in every
            iteration,decrement a hour and also decrement 60
            minute from total minute.

            temp.mm=n-60; //113-60=53
            //Now time will become as time->12:53

            //now subtracting 60 minute from total minute(n) and
            60 minute is subtracted from temp.minute

            n-=60;

            //decrementing hour as 60 minutes are subtracted

            temp.hh-=1;

            //Hour will become 12 if it will become 0

            if(temp.hh==0)
                temp.hh=12;
}

```

```

        }

        //remaing minutes which is <60 is subtracted here from
temp.minute.

        temp.mm=60-n;

        //decrementing hour as 60 minutes are subtracted

        temp.hh-=1;

        //Hour will become 12 if it will become 0

        if(temp.hh==0)
            temp.hh=12;
    }
    return temp;
}
void main()
{
    time obj1,rslt,obj2;
    int num,ch=1;
    cout<<"\n Object1 input";
    obj1.input();
    cout<<"\n Object1 display";
    obj1.display();

    cout<<"\n Enter minutes to add:";
    cin>>num;
    cout<<"\n-----After adding the minutes to time-----
--";
    rslt=obj1+num;
    rslt.display();

    cout<<"\n Enter minutes to subtract:";
    cin>>num;
    cout<<"\n-----After subtracting the minutes to time-----
-----";
    rslt=obj1-num;
    rslt.display();

    cout<<"\n----- = operator-----";
    cout<<"\n Object2 input";
    obj2.input();
    cout<<"\n Object2 display";
    obj2.display();

    obj2=obj1;
    cout<<"\n Object1 data is copied into object2\nAnd displaying
Object2 data";
    obj2.display();

    cout<<"\n";
    cout<<"\n Please enter Object2 input again";
}

```

```

obj2.input();
cout<<"\n Object2 display";
obj2.display();

cout<<"\n\n\n";

if(obj1>obj2)
    cout<<"\n Object1 is greater than object2";
else if(obj2>obj1)
    cout<<"\n Object2 is greater than object1";

if(obj1<obj2)
    cout<<"\n Object1 is less than object2";
else if(obj2<obj1)
    cout<<"\n Object2 is less than object1";

if(obj1>=obj2)
    cout<<"\n Object1 is greater than equal to object2";
else if(obj2>=obj1)
    cout<<"\n Object2 is greater than equal to object1";

if(obj1<=obj2)
    cout<<"\n Object1 is less than equal to object2";
else if(obj2<=obj1)
    cout<<"\n Object2 is less than equal to object1";
cout<<"\n\n\n";

cout<<"\n wihtout Pre Incremented object1 data";
obj1.display();
cout<<"\n Pre Incremented object1 data";
++obj1;
obj1.display();
printf("\n");
cout<<"\n Post Incremented object1 data";
obj1++;
obj1.display();
cout<<"\n";
cout<<"\n wihtout Decrementated object2 data";
obj2.display();
cout<<"\n Pre Decrementated object2 data";
--obj2;
obj2.display();
printf("\n");
cout<<"\n Post Decrementated object2 data";
obj2--;
obj2.display();
printf("\n");
}

*****
*****
*****
Output:

```

Object1 input

```
Enter Time in proper format(hh mm ss):2 3 4
```

```
Object1 display  
time is:2:3:4
```

```
Enter minutes to add:200
```

```
-----After adding the minutes to time-----  
time is:5:23:4
```

```
Enter minutes to subtract:10
```

```
-----After subtracting the minutes to time-----  
time is:1:53:4
```

```
----- = operator-----
```

```
Object2 input  
Enter Time in proper format(hh mm ss):4 5 6
```

```
Object2 display  
time is:4:5:6
```

```
Object1 data is copied into object2  
And displaying Object2 data  
time is:2:3:4
```

```
Please enter Object2 input again  
Enter Time in proper format(hh mm ss):2 6 7
```

```
Object2 display  
time is:2:6:7
```

```
Object2 is greater than equal to object1  
Object1 is less than equal to object2
```

```
wihtout Pre Incremented object1 data  
time is:2:3:4
```

```
Pre Incremented object1 data  
time is:2:4:4
```

```
Post Incremented object1 data  
time is:2:5:4
```

```
wihtout Decrement object2 data  
time is:2:6:7
```

```
Pre Decremented object2 data
time is:2:5:7
```

```
Post Decremented object2 data
time is:2:4:7
```

```
Press any key to continue . . .
```

```
*****  
*****
```

```
Q(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)

```
*****  
*****
```

```
#include<iostream>
using namespace std;
class Name
{
    char nm[100];
public:
    void input()
    {
        cout<<"\n Enter String:";
        cin>>nm;
    }
    const Name operator =(const Name obj2)
    {
        int i=0;
        while(obj2.nm[i]!='\0')
        {
            nm[i]=obj2.nm[i];
            i++;
        }
        nm[i]='\0';

        return *this;
    }
    bool operator ==(Name obj3)
    {
        int i=0,len1=0,len2=0;

        //first 2 loops calculate length
        while(obj3.nm[i]!='\0')
        {   len1++; i++; }
```

```

i=0;
while(nm[i]!='\0')
{   len2++; i++;}

if(len1!=len2)
    return false;
else
{
    i=0;
    while(nm[i]!='\0')
    {
        if(nm[i]!=obj3.nm[i])
        {   break; }
        i++;
    }
    cout<<"\n"<<len1;
    cout<<"\n"<<i;
    if(i==len1)
        return true;
    else
        return false;
}
}

bool operator <=(Name temp)
{
    int i=0,len1=0,len2=0;
    while(temp.nm[i]!='\0')
    {   len1++; i++;}
    i=0;
    while(nm[i]!='\0')
    {   len2++; i++;}

    if(len2<=len1)
        return true;
    else
        return false;
}

char operator [](int k)
{
    return nm[k];
}

void reverse()
{
    int len1=0,i=0,t;
    char temp;
    while(nm[i]!='\0')
    {   ++len1; i++;}
    len1--;
    t=len1;
    for(i=0;i<(len1/2)+1;i++)
    {
        temp=nm[i];
        nm[i]=nm[t];
        nm[t]=temp;
    }
}

```

```

        t--;
    }
    cout<<nm;
}
Name operator +(Name t1)
{
    int i=0,len1=0,len2=0;
    while(t1.nm[i]!='\0')
    {
        len2++; i++;
    }
    i=0;
    while(nm[i]!='\0')
    {
        len1++; i++;
    }
    i=0;
    while(t1.nm[i]!='\0')
    {
        nm[len1]=t1.nm[i];
        len1++;
        i++;
    }
    nm[len1]='\0';
    return *this;
}
void display()
{
    int i=0;
    while(nm[i]!='\0')
    {
        cout<<nm[i];
        i++;
    }
}
};

void main()
{
    Name obj1,obj2,obj3;
    int no=1,k,a[100];
    cout<<"\n Object1 input";
    obj1.input();
    cout<<"\n Object1 display:";
    obj1.display();
    cout<<"\n";
    cout<<"\n Object2 input";
    obj2.input();
    cout<<"\n";
    cout<<"\n Object2 display:";
    obj2.display();
    cout<<"\n";
    while(no!=0)
    {
        cout<<"\n 1.Check Equality";
        cout<<"\n 2.Check whether first string is smaller than the
second";
        cout<<"\n 3.Copy object2 string to object1 string";
        cout<<"\n 4.Extract Character";

```

```

cout<<"\n 5.Reverse String";
cout<<"\n 6.Concatenation of 2 strings";
cout<<"\n Enter number from above menu:";
cin>>no;
switch(no)
{
case 1:if(obj1==obj2)
           cout<<"\n Both objects are equal";
       else
           cout<<"\n Both objects are Not equal";
       break;
case 2:if(obj1<=obj2)
           cout<<"\n First string is less than equal to
second string";
       else
           cout<<"\n First string is not less than
equal to second string";
       break;
case 3:obj1=obj2;
           cout<<"\n After Copying obj2 string into obj1
string.\nDisplaying obj1 string:";
           obj1.display();
       break;
case 4:cout<<"\n Enter index. And i will Extract the
character:";
           cin>>k;
           cout<<obj1[k];
       break;
case 5:obj1.reverse();
       break;
case 6:obj3=obj1+obj2;
           obj3.display();
       break;
}
}
}

```


Output:

Object1 input
Enter String:pushti

Object1 display:pushti

Object2 input
Enter String:priya

Object2 display:priya

- 1.Check Equality
- 2.Check whether first string is smaller than the second
- 3.Copy object2 string to object1 string

```
4.Extract Character  
5.Reverse String  
6.Concat 2 strings  
Enter number from above menu:1
```

Both objects are Not equal

```
1.Check Equality  
2.Check whether first string is smaller than the second  
3.Copy object2 string to object1 string  
4.Extract Character  
5.Reverse String  
6.Concat 2 strings  
Enter number from above menu:2
```

First string is not less than equal to second string

```
1.Check Equality  
2.Check whether first string is smaller than the second  
3.Copy object2 string to object1 string  
4.Extract Character  
5.Reverse String  
6.Concat 2 strings  
Enter number from above menu:3
```

After Copying obj2 string into obj1 string.

```
Displaying obj1 stringpriya  
1.Check Equality  
2.Check whether first string is smaller than the second  
3.Copy object2 string to object1 string  
4.Extract Character  
5.Reverse String  
6.Concat 2 strings  
Enter number from above menu:4
```

Enter index.And i will Extract the character4

a

```
1.Check Equality  
2.Check whether first string is smaller than the second  
3.Copy object2 string to object1 string  
4.Extract Character  
5.Reverse String  
6.Concat 2 strings
```

```
Enter number from above menu:5  
ayirp
```

```
1.Check Equality  
2.Check whether first string is smaller than the second  
3.Copy object2 string to object1 string  
4.Extract Character  
5.Reverse String
```

6.Concat 2 strings

Enter number from above menu:6
ayirppriya

- 1.Check Equality
- 2.Check whether first string is smaller than the second
- 3.Copy object2 string to object1 string
- 4.Extract Character
- 5.Reverse String
- 6.Concat 2 strings

Enter number from above menu:0

Q(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 +


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

ostream & form(ostream & output)
{
    output<<setfill('+');
    output<<setiosflags(ios::showpoint);
    output<<setiosflags(ios::fixed);
    output<<setw(5);
    output<<setprecision(2);

    return output;
}

void main()
{
    float f;
    cout<<"\n Enter floating number:";
    cin>>f;
    cout<<form;
    cout<<f<<endl;
}
```


Output:

Enter floating number:1.2345
+1.23

Press any key to continue . . .

```
*****
*****
Q(10): Define a class marksheet. The class should contain a function
PrintMarkSheet such that it prints the marksheet 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).

*****
*****
#include<iostream>
#include<string>
#include<iomanip>
using namespace std;
namespace m1
{
    ostream & printheading(ostream & out)
    {
        out<<setw(80);
        out<<setiosflags(ios::left);
        out<<"Rollwala Computer
Center"<<endl<<endl<<setw(80)<<"Name:Preksha Sheth"<<endl<<endl;
        return out;
    }
    ostream & PrintLine(ostream & out)//userdefined Manipulator
    {
        out<<"-----";
        out<<endl;
        return out;
    }
}
class marksheet
{
    string s1,s2,s3;
    float a1[5],a2[5],a3[5];
public:
    void input()
    {
        cout<<"\n Enter Subject1 Name:";
        cin>>s1;
        for(int i=0;i<5;i++)
        {
            cout<<"\n Enter Mark"<<i+1<<" :";
            cin>>a1[i];
        }
        cout<<"\n Enter Subject2 Name:";
        cin>>s2;
        for(int i=0;i<5;i++)
        {
            cout<<"\n Enter Mark"<<i+1<<" :";
        }
    }
}
```

```

        cin>>a2[i];
    }
    cout<<"\n Enter Subject3 Name:";
    cin>>s3;
    for(int i=0;i<5;i++)
    {
        cout<<"\n Enter Mark"<<i+1<<" :";
        cin>>a3[i];
    }
}

void printmarksheet()
{
    cout<<m1::PrintLine;
    cout<<m1::printheading;
    cout<<m1::PrintLine;
    cout.width(15);
    cout.precision(2); //after point value will not show
    cout.setf(ios::showpoint);
    cout.setf(ios::fixed); //non scientific notation
    cout.setf(ios::left);
    cout<<"Subject";

    for(int i=0;i<5;i++)
    {
        cout.width(4);
        cout<<"Mark";
        cout.width(7);
        cout<<i+1;
    }
    cout<<endl;
    cout<<m1::PrintLine;
    cout<<"\n";
    cout.width(15);
    cout<<s1;
    for(int i=0;i<5;i++)
    {
        cout.width(11);
        cout<<a1[i];
    }
    cout<<endl;
    cout<<"\n";
    cout.width(15);
    cout<<s2;
    for(int i=0;i<5;i++)
    {
        cout.width(11);
        cout<<a2[i];
    }
    cout<<endl;
    cout<<"\n";
    cout.width(15);
    cout<<s3;
    for(int i=0;i<5;i++)
}

```

```

    {
        cout.width(11);
        cout<<a3[i];
    }
    cout<<endl;
    cout<<m1::PrintLine<<"\n\n\n";
}
};

void main()
{
    marksheet obj1;
    obj1.input();
    cout<<"\n\n"<<setw(43)<<"Marksheet";
    cout<<"\n";
    obj1.printmarksheet();
}

```

```
*****
*****
Output:
```

Enter Subject1 Name:fop

Enter Mark1:12.43333

Enter Mark2:12.33333

Enter Mark3:20.000001

Enter Mark4:20.01

Enter Mark5:5

Enter Subject2 Name:oocp

Enter Mark1:20.034

Enter Mark2:34.678

Enter Mark3:12.89034

Enter Mark4:12

Enter Mark5:34

Enter Subject3 Name:php

Enter Mark1:1

Enter Mark2:2

Enter Mark3:3

Enter Mark4:4

Enter Mark5:5

Marksheet

Rollwala Computer Center

Name:Preksha Sheth

Subject	Mark1	Mark2	Mark3	Mark4	Mark5
fop	12.43	12.33		20.00	20.01
	5.00				
oocp	20.03	34.68		12.89	12.00
	34.00				
php	1.00	2.00		3.00	4.00
	5.00				

Press any key to continue . . .

ASSIGNMENT - 3

Q.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.

```

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

```

```
*****  
*****
```

Q.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
```

```
*****  
*****
```

```
Q.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:7

Enter Value:8

Enter Value:4

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
```

```
*****  
*****  
Q.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:preksha

Enter Roll no:2

Enter Name:prerak

Total Student is:
-----
roll no is:1
Name is:preksha
roll no is:2
Name is:prerak
-----
Enter roll no which you want to find:4

rollno is not found

```


Q.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

*****
***** Q.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

*****  

*****  

Q.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:"<

```

```

        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:Preksha

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:kriya

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:Preksha
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:kriya
stream is:commerce

Result is:83

```
*****
*****Q.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<<"-----"<<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;
    }
};

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 preksha
```

```
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:preksha
```

```
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
```

```
*****  
*****
```

```
Q.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:preksha

Enter Experience:4

Enter Pay:7800

Do You Want to continue:y

Enter Your Choice:1

Enter Code:102

Enter Name:prerak

Enter Experience:8

Enter Pay:5680

Do You Want to continue:y

Enter Your Choice:3

```
-----  
Code is:101  
Name is:preksha  
Experience is:4  
Pay is:7800  
-----
```

```
-----  
Code is:102  
Name is:prerak  
Experience is:8  
Pay is:5680  
-----
```

Do You Want to continue:y

Enter Your Choice:2

Enter code:102

Enter Name:prerak

Enter Experience:2

Enter Pay:5600

Do You Want to continue:y

Enter Your Choice:3

```
-----  
Code is:101  
Name is:preksha  
Experience is:4  
Pay is:7800  
-----
```

```
-----  
Code is:102  
Name is:prerak  
Experience is:2
```

Pay is:5600

Do You Want to continue:n

Q.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:56  
  
rollno is:1  
name is:priya  
Internal marks is:45  
Internal marks is:89  
Internal marks is:78  
sports marks is:56
```

```
*****  
*****
```

Q.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);  
        // cout<<"Input:"<<endl;  
        while(true)  
        {  
            cin.getline(inputline,80);  
            if(!strcmp(inputline, "End"))  
        }  
    }  
}
```

```

        break;
    Entryfile<<"\n"<<inputline;
}

Entryfile.close();
}

void read_file(string file)
{
//    cout<<"Output"<<endl;
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;

fp.write_file("data.txt");
fp.read_file("data.txt");
fp.copy_file("details.txt","data.txt");
fp.read_file("details.txt");
fp.count_word_lines("data.txt");
getch();
*/
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
*****
*****Q.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:preksha  
  
Enter marks:89  
  
Enter marks:78  
  
Enter marks:56  
  
Do you want to continue:(y/n)y  
  
Enter Roll no:2  
  
Enter name:prerak  
  
Enter marks:45  
  
Enter marks:56  
  
Enter marks:78  
  
Do you want to continue(y/n):n
```

```
Output:
```

```
Roll no is:1  
Name is:preksha  
marks of sub 1 is:89  
marks of sub 2 is:78  
marks of sub 3 is:56
```

```
Roll no is:2  
Name is:prerak  
marks of sub 1 is:45  
marks of sub 2 is:56  
marks of sub 3 is:78
```

```
*****  
*****  
*****  
Q.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();

}

display.close();
getch();
}

```


Output :

Input Data:
Enter Roll no:1
Enter name:preksha
Enter marks:45
Enter marks:89
Enter marks:78
Do you want to continue:(y/n)y
Enter Roll no:2
Enter name:prerak
Enter marks:45

Enter marks:56

Enter marks:2

Do you want to continue: (y/n)

y

Enter Roll no:3

Enter name:rutwik

Enter marks:47

Enter marks:89

Enter marks:66

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

Enter Roll no:4

Enter name:dhruvin

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:abhi

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:jeet

Enter marks:56

Enter marks:65

Enter marks:54

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

Enter Roll no:9

Enter name:mahi

Enter marks:78

Enter marks:89

Enter marks:66

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

Enter Roll no:10

Enter name:kashish

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	preksha	70
2	prerak	34
3	rutwik	67
4	dhruvin	70
5	meet	26
6	abhi	51
7	aditi	85
8	jeet	58
9	mahi	77
10	kashish	84

```
*****  
*****
```

Q.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: preksha

Enter the Address:ahmedabad

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

Enter the emp id: 102

Enter the name: prerak

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: preksha

Address:ahmedabad

id: 102

Name: prerak

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: prerak

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: mahi
1. Add
2. Display
3. Update
4. Exit
2
1. Display Full List
2. Display Particular Detail
1
id: 101
Name: preksha
Address:ahmedabad
id: 102
Name: prerak
Address:rajkot
1. Add
2. Display
3. Update
4. Exit
4
*****
```


Q.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: preksha
Enter account type: savings
Enter the opening balance: 600
Do you want continue? (y/n): y
Enter account number: 2
Enter name of customer: prerak
Enter account type: currunt
Enter the opening balance: 25000
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: 2
Account Number: 2
Name: prerak
Balance: 25000

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

```

```

3
Enter account number: 2
1. Deposite
2. Withdraw
2
Enter the amount you want to withdraw: 100
Transaction completed successfully

1. Add Customer
2. Customer Details
3. Perform Transaction
4. Transaction History
5. Exit
2
Enter account number: 2
Account Number: 2
Name: prerak
Balance: 24900

1. Add Customer
2. Customer Details
3. Perform Transaction
4. Transaction History
5. Exit
4
Enter account number: 2
Account Numebr: 2
Transaction Date: 6/6/2020
Transaction type: Debit
Amount: 100
Balance: 24900

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

```

