

**Department of Computer Science &
Engineering**

LAB MANUAL

**SUBJECT:-OBJECT ORIENTED PRGRAMMING USING
C++ [BTCS-309]**

**B.Tech–III Semester [2012-13]
(Branch: CSE & ECE)**



**KCT COLLEGE OF ENGG & TECH.,
FATEHGARH**

Punjab Technical University

INDEX

Sr. No	Name of Experiment
1	Write a program to demonstrate the use of basic data types
2	WAP to show the use of loops and conditional Statements like if-else, switch case etc.
3	WAP to show the use of storage classes-auto ,static, register and extern
4	WAP using goto and continue ,break statements.
5	WAP to demonstrate basic use of arrays ,one dimensional and wo dimensional arrays
6	WAP to multiply two two-dimensional arrays.
7	Write a program which uses pre-processors, like #pragma.
8	WAP to show use of functions, give proper explanation of control transfer involved in function calls.
9	WAP to demonstrates the classes concept.
10	WAP to show constructor in c++.
11	Write a program to demonstrate all types of constructors
12	WAP to demonstrate the concept of function overloading.
13	WAP to derive a class from other class, i.e. to demo the concept of inheritance
14	How we implement data abstraction and data encapsulation ,data hiding using classes in inheritance or various types of inheritance.
15	WAP to overload a unary and binary operators .for example to overload ‘++’ and ‘+’ operators.
16	Wap to demonstrates polymorphism.
17	Wap to demonstrates inline function

EXPERIMENT NO:1

Title: -- Program to demonstrate different data types

Objective:--To learn the use of basic data types and checking their memory requirement.

S/W Requirement:--Turbo C++

H/W Requirement:-- An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components you require.

A CD-ROM disk drive

Theory:--To store different types of data c++ provides different data types (e.g. Integers, characters, float, etc.).To allocate memory to a particular variable we need to mention its data type .eg int indicates that variable mentioned after it will get **2 bytes** of memory

Program:--

```
#include< iostream .h >
#include< conio. h >
#include< stdio. h >
void main()
{
int a;
char b;
float c;
double d;
long int e;

clrscr();
```

```
cout<<"Enter an integer ";
cin>>a;
cout<<"\nYou entered-- "<<a<<"\n\n it's size in memory is"<<sizeof(a)<<"byte(s)";
```

```
////////////////////////////////////
```

```
cout<<"\n\n Enter a character ";
cin>>b;
cout<<"\nYou entered-- "<<b<<"\n\nIt's size in memory is"<<sizeof(b)<<"byte(s)";
```

```
////////////////////////////////////
```

```
cout<<"\n\n Enter a float ";
cin>>c;
cout<<"\nYou entered "<<c<<"\n\nIt's size in memory is---"<<sizeof(c)<<"byte(s)";
```

```
////////////////////////////////////
```

```
cout<<"\n\n Enter an double ";
cin>>d;
cout<<"\nyou entered "<<d<<"\n\nit size in memory is"<<sizeof(d)<<"bytes" ;
```

```
////////////////////////////////////
```

```
getchar();
```

```
}
```

Output:-

Enter an integer 2

You entered-- 2

Its size in memory is--2byte(s)

Enter a character d

You entered-- d

Its size in memory is--1byte(s)

Enter a float 456.23

You entered 456.230011

Its size in memory is---4byte(s)

Enter an double 234324532.3

You entered 2.343245e+08

It size in memory is8bytes

VIVA QUESTIONS

1q:-What is data types?

2q:-How many data types are used for memory?

3q:-How many bytes does int used to take for memory location.

4q:-Why we use sizeof function?

5q:- How many bytes does char used to take for memory location.

EXPERIMENT NO: 2

Title: -- WAP to show the use of loops and conditional Statements like if-else, switch Case etc.

Objective: -- To print the following pattern

```
1
12
123
1234
12345
```

S/W Requirement: -- Turbo C++

H/W Requirement:-- An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

You require.

A CD-ROM disk drive

Theory:-- we used the for loop for iteration required for displaying the above pattern.

ON each iteration of loop loop counter of inner loop is printed .

Program:--

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

void main()

{
    clrscr();
    int i , j;

    for( i=1;i<=5;i++)
        {
            for(j=1;j<=i ;j++)
                {
                    cout<<j;
                }
        }
}
```

```
        cout<<"\n";
    }
    getch();
}
```

Output:-:

1

12

123

1234

12345

VIVA QUESTIONS

1q:-What is looping?

2q:-How many types of looping are used?

3q:-write syntax of For loop.

4q:-what is the difference between do-while and while loop?

5q:- what is recursion.

EXPERIMENT NO. 3

Title: -- WAP to show the use of storage classes-auto, static, register and extern

Objective:- To check the effect of auto storage class

S/W Requirement: -- Turbo C++

H/W Requirement:--IBM PC compatible

An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

You require.

A CD-ROM disk drive

Theory:- Following program shows how an automatic storage class variable is declared, and the fact that if the variable is not initialized it contains a garbage value.

Program:--

```
#include<iostream.h>
#include<conio.h>
void main( )
{
    clrscr();
    auto int i=1
    {
        {
            cout << i ;
        }
        cout << i ;
    }
}
```

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

Output:-:

111

VIVA QUESTIONS

1q:-What is a storage class?

2q:-How many types of storage class are their?

3q:-What do you mean by static?

4q:- What do you mean by auto?

5q:- What do you mean by register?

EXPERIMENT NO. --4

Title: WAP using goto and continue , break statements,

Objective:- Program to compare two numbers making use of goto statement

S/W Requirement: -- Turbo C++

H/W Requirement:--IBM PC compatible

An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

You require.

A CD-ROM disk drive

Theory: **goto** statement causes the control to jump to a particular label mentioned in goto statement, **break** statement causes the control to come out of one loop, **continue** statement is used inside the loops ,it makes the control to return to starting of loop.

Program:-

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
void main()
{
    clrscr();
    int a,b;
    cout<<"Enter two integers \n\n" ;
    cin>>a>>b;
    if(a>b)
```

```
goto aaaa;
else
cout<<"second integer is bigger";
goto bbbb;
aaaa:
{
cout<<"firsrt one is bigger" ;
getch();
}
bbbb:
getch();
}
```

Output:-**Enter two integers****4****7****Second integer is bigger**

VIVA QUESTIONS

1q:-What is a control statements?

2q:-Why we use break statement?

3q:- In which statement we use label?

4q:- What do you mean by continue statement?

5q:- What do you mean by goto statement?

EXPERIMENT NO. --5

Title: -- WAP to demonstrate basic use of arrays, one dimensional and two dimensional arrays

Objective :-- one dimensional array-to reverse an integer array

Two dimensional array-to add two two-dimensional integer arrays

S/W Requirement: -- Turbo C++

H/W Requirement:--IBM PC compatible

An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

You require.

A CD-ROM disk drive

Theory:- An array is a series of elements of the same type placed in contiguous memory locations that can be individually referenced by adding an index to a unique identifier. we can access the value of any of its elements individually as if it was a normal variable, thus being able to both read and modify its value.

Data Structures Used :-- Arrays

Program:--

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

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

```
void main()
```

```
{
```

```
clrscr();
```

```
int a[10];
```



```
char ch;
cout<<"enetr 10 integers\n\n";
for(int i=0;i<10;i++)
    cin>>a[i];

cout<<"want to reverse it(y/n)?";
cin>>ch;
if(ch=='y')
{
    for(i=9;i>=0;i--)
    {
        cout<< a[i]<<"\n";
    }
}
else
{
    cout<<" you entered\n ";
    for(i=0;i<10;i++)
        cout<< a[i]<<"\n";
}
getch();
}
```

Output:-**enetr 10 integers**

1
2
3
4

5

6

7

8

9

10

Want to reverse it(y/n)? y

10

9

8

7

6

5

4

3

2

1

Two dimensional array-to add two two-dimensional integer arrays

```
#include<iomanip.h>
#include<iostream.h>
main()

{
int a[3][3],b[3][3],c[3][3],i,j;
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
cin>>a[i][j];
}
cout<<'\n';
}

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

```
        for(j=0;j<3;j++)
        {
            cin>>b[i][j];
        }
        cout<<'\n';
    }

    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            c[i][j]=a[i][j]+b[i][j];
        }
        cout<<'\n';
    }

    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            cout<<setw(3)<<c[i][j];
        }
        cout<<'\n';
    }
}
```

VIVA QUESTIONS

1q:-What do you mean by arrays?

2q:-How many dimensional array is used for memory?

3q:- Is Array stores data in Non-continues location?

4q:- What are the advantages and disadvantages of array?

5q:- How we store data in 1-dimensional array?

EXPERIMENT NO. - 6

Title :- WAP to multiply two two-dimensional arrays.

Objective:-To learn how to operate on two dimensional arrays

S/W Requirement: -- Turbo C++

H/W Requirement:--IBM PC compatible

An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

You require.

A CD-ROM disk drive

Theory: --Two-dimensional arrays are declared by specifying the number of rows then the number of columns. Unless specified, all initial values of arrays are *garbage*. You can specify initial values by enclosing each row in curly braces like this: Write subscripts as $x[\text{row}][\text{col}]$. Passing over all elements of a two-dimensional array is usually done with two *nested* for loops.

Data Structures Used (Optional):--Arrays

Program:-

```
#include<iomanip.h>
```

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

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

```
void main()
```

```
{
```

```
clrscr();
```

```
int a[3][3],k,b[3][3],c[3][3],i,j;
```

```
cout<<"enter a matrix of size 3x3\n\n";
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        cin>>a[i][j];
    }
    cout<<"\n";
}
cout<<"you entered\n\n";
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        cout<<a[i][j];
        cout<<" ";
    }
    cout<<"\n";
}
cout<<"you entered first matrix ,now enter second matrix ";
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        cin>>b[i][j];
    }
    cout<<"\n";
}
cout<<"you entered \n\n";
for(i=0;i<3;i++)
{
```

```
        for(j=0;j<3;j++)
        {
            cout<<b[i][j];
            cout<<" ";
        }
        cout<<'\n';
    }
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        c[i][j]=0;
    }
    cout<<'\n';
}
cout<<"multiplication going on.....\n\n\n" ;
cout<<"result is\n\n";
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        c[i][j]=0;
        for(k=0;k<3;k++)
        {
            c[i][j]=c[i][j]+(a[i][k]*b[k][j]);
        }
        cout<<setw(3)<<c[i][j];
    }
    cout<<'\n';
}
getch();
```

```
} //end of main
```

Output:-

enter a matrix of size 3x3

1

1

1

2

2

2

3

3

3

you entered

1 1 1

2 2 2

3 3 3

you entered first matrix ,now enter second matrix

4

4

4

5

5

5

6

6

6

you entered

4 4 4

5 5 5

6 6 6

multiplication going on.....

result is

15 15 15

30 30 30

45 45 45

VIVA QUESTIONS

1q:-What do you mean by 2 dim arrays?

2q:-Where we use 2-dimensional array?

3q:- what do you mean by Matrices?

4q:- what do you mean multi-dimensional array?

5q:- How we store data in 2-dimensional array?

EXPERIMENT NO.7

Title:-- Write a program which uses pre-processors directives like #pragma

Objective:-- To learn the use of preprocessors

S/W Requirement: -- Turbo C++

H/W Requirement:--IBM PC compatible

An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

You require.

A CD-ROM disk drive

Theory: -- It is a program that processes our source program before it is passed to the compiler. Preprocessor commands (often known as directives) form what can almost be considered a language within C language.

For example **#pragma startup** and **#pragma exit**: These directives allow us to specify functions that are called upon program startup (before main ()) or program exit (just before the program terminates). Their usage is as follows:

For following program (Note that the functions fun1 () and fun2 () should neither receive nor return any value. If we want two functions to get executed at startup then their pragmas should be defined in the reverse order in which you want to get them called.)

Program:--

```
void fun1( );
```

```
void fun2( );
```

```
#pragma startup fun1
```

```
#pragma exit fun2
```

```
Void main( )
```

```
{
```

```
printf ( "\nInside main" );  
}  
void fun1()  
{  
printf ( "\nInside fun1" );  
}  
void fun2()  
{  
printf ( "\nInside fun2" );  
getch();  
}
```

Output:-**Inside fun1****Inside main****Inside fun2**

VIVA QUESTIONS

1q:- What do you mean by preprocessor?

2q:- Define **#pragma startup** and **#pragma exit**?

3q:- Is function is called outside in preprocessor?

4q:- What are the advantages and disadvantages of directives preprocessor?

5q:- Where preprocessor directives are used?

EXPERIMENT NO:-.8

Title: --WAP to show use of functions, give proper explanation of control transfer involved in function calls.

Objective:--To learn how control transfers in function calls

S/W Requirement: -- Turbo C++

H/W Requirement:--IBM PC compatible

An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

You require.

A CD-ROM disk drive

Program:--

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

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

```
int Add(int,int); //function prototype
```

```
int a,b,c;
```

```
void main()
```

```
{
```

```
clrscr();
```

```
cout<<"enter two numbers\n";
```

```
cin>>a;
cin>>b;
c=Add(a,b); //function call
cout <<"\n\nsum of above two numbers ="<<c << endl;
fetch();

}
```

```
int Add(int a, int b) //function definition
{
c=a+b;
return c;
}
```

Output:-**enter two numbers****3****4****sum of above two numbers =7**

VIVA QUESTIONS

1q:- What do you mean by function?

2q:- Define call by value and call by reference?

3q:- Is function is called outside in call by value?

4q:- What are the advantages and disadvantages of function?

5q:- what do you mean by return type and argument?

EXPERIMENT NO:9

Title: -- WAP to demonstrates the classes concept.

Objective:--To learn the use of classes declaration.

S/W Requirement:--Turbo C++

H/W Requirement:-- An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

you require.

A CD-ROM disk drive

Theory:--To store different types of data c++ provides different data types (e.g. Integers, characters, float, etc.).in classes used.

Program:--

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

```
#include<string.h>
```

```
class student
```

```
{private:char reg_no[10];
```

```
    char name[30];
```

```
    int age;
```

```
    char address[25];
```

```
public:void int_data(void)
```

```
    {cout<<"!!!!!!enter the data of student!!!!!!";
```

```
    cout<<"Registration no.=";
```

```
    gets(reg_no);
```

```
    cout<<"Name----";
```

```
    gets(name);
```

```
cout<<"Age=";  
cin>>age;  
cout<<"Address----";  
gets(address);  
}  
};
```

```
void main()  
{student s1;  
s1.int_data();  
}
```

Output: !!!!!enter the data of student!!!!!!!

Registration no.=123

Name----abc

Age=21

Address----city..... state....

VIVA QUESTIONS

1q:- What do you mean by class?

2q:- Define object?

3q:- Is member function called outside in class? If yes? how?

4q:- What are access member used in class?

5q:- What is objective of classes in c++?

EXPERIMENT NO:10

Title: -- WAP to show constructor in c++.

Objective:--To learn the use of constructor.

S/W Requirement:--Turbo C++

H/W Requirement:-- An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components
you require.

A CD-ROM disk drive

Theory:- Constructor are used to initialize memory.how it used to initialize it.

Program:

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

```
class ash
{int a,b;
public:ash();
    void sumdata();
    };
```

```
ash::ash()
{a=3;
b=4;
}
```

```
void ash::sumdata()
{cout<<"sum is="<<a+b;
}
```

```
void main()
{ash a;
a.sumdata();
}
```

Output: sum is= 7

VIVA QUESTIONS

1q:- What do you mean by constructor?

2q:- Define properties of constructor?

3q:- Is constructor used as same name of class? If yes?why?

4q:- What are access member used in constructor?

5q:- What is destructor ?

EXPERIMENT NO:11

Title: -- Write a program to demonstrate all types of constructors

Objective:--To learn the type of constructor.

S/W Requirement:--Turbo C++

H/W Requirement:-- An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

you require.

A CD-ROM disk drive

Theory:- Constructor are used to initialize memory.how it used to initialize it .The various types of constructors which are used to be declared.

Program's :

/*Example of DEFAULT Constructor*/

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

```
class ash
{int a,b;
public:ash();
    void sumdata();
    };
```

```
ash::ash()
{a=3;
b=4;
}
```

```
void ash::sumdata()
```

```
{cout<<"sum is="<<a+b;
}
```

```
void main()
{ash a;
a.sumdata();
}
```

Output: sum is= 7

/* Example of PARAMETERIZED Constructor */

```
#include<iostream.h>
class ash
{private:int a,b;
public:ash(int c,int d);
      void sumdata();
      };
```

```
ash::ash(int c,int d)
{a=c;
b=d;
}
```

```
void ash::sumdata()
{cout<<"sum is="<<a+b;
}
```

```
void main()
{ash a(4,5);
a.sumdata();
}
```

Output: sum is= 9

/*Example of COPY Constructor */

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

```
class ash
```

```
{private:int a,b;
```

```
public:ash(int c,int d)
```

```
{a=c;
```

```
  b=d;
```

```
}
```

```
  ash(ash&i)
```

```
{a=i.a;
```

```
  b=i.b;
```

```
}
```

```
  void sumdata();
```

```
};
```

```
void ash::sumdata()
```

```
{cout<<"sum is="<<a+b;
```

```
}
```

```
void main()
```

```
{ash a(4,5);
```

```
ash b(a);
```

```
b.sumdata();
```

```
}
```

Output: sum is= 9

/*Example of OVERLOADED Constructor */

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

```
class ash
```

```
{private:int m,n;
```

```
public:ash()
```

```
{m=0;
```

```
  n=0;
```

```
}
```

```
  ash(int a,int b)
```

```
{m=a;
```

```
  n=b;
```

```
}
```

```
  ash(ash&i)
```

```
{m=i.m;  
n=i.n;  
}  
void sumdata();  
};
```

```
void ash::sumdata()  
{cout<<"sum is="<<m+n;  
}
```

```
void main()  
{ash a();  
ash aa(4,5);  
ash bb(aa);  
bb.sumdata();  
}
```

Output: sum is= 9

VIVA QUESTIONS

1q:- How many types of constructors are there?

2q:- Define default constructor?

3q:- Define copy constructor

4q:- Define overloaded constructor?

5q:- Define Parameterized constructor?

EXPERIMENT NO:12

Title: -- Write a program to demonstrate overloading operator's

Objective:--To learn the overloading concept..

S/W Requirement:--Turbo C++

H/W Requirement:-- An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

you require.

A CD-ROM disk drive

Theory:- operator overloading is used to load various operation for performing different functions with same operator.

Program's :

```
/*Overloading of a BINARY OPERATOR */  
  
#include<iostream.h>  
  
class complex  
{float x,y;  
public:complex()  
    {}  
    complex(float real,float img)  
    {x=real;  
    y=img;  
    }  
    complex operator+(complex);  
    void display(void);  
};  
  
complex complex::operator+(complex c)  
{complex temp;  
temp.x=x+c.x;
```

```
temp.y=y+c.y;
return(temp);
}

void complex::display(void)
{
cout<<x<<"+"j"<<y<<"\n";
}

void main()
{
complex c1,c2,c3;
c1=complex(1.2,4.5);
c2=complex(3.4,9.5);
c3=c1+c2;
cout<<"c1= ";
c1.display();
cout<<"c2= ";
c2.display();
cout<<"c3= ";
c3.display();
}
```

Output: c1=1.2+j4.5
c2=3.4+j9.5
c3=4.6+j14.0

VIVA QUESTIONS

- 1q:- Define operator overloading?
- 2q:- Define binary operator overloading?
- 3q:- Define unary operator overloading
- 4q:- Which operators are not used as for overloading.
- 5q:- For what purpose ++ operator used?

EXPERIMENT NO:13

Title: -- Write a program to demonstrate inheritance

Objective:--To learn the inheritance concept..

S/W Requirement:--Turbo C++

H/W Requirement:-- An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

you require.

A CD-ROM disk drive

Theory:- inheritance is used to derived the properties of one class to another class.

Program:

/* Example of SINGLE Inheritance */

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

class a{
int a;
public: int b;
    void show_a(void)
    {cout<<"a=";
    cin>>a;
    }
    int get_a(void)
    {return a;
    }};

class b:public a
{int c;
public:void get_b(void)
    {cout<<"b=";
```

```
        cin>>b;
    }

void mul(void)
{c=b*get_a();
cout<<"a*b="<<c;
}};

void main()
{
b n;
n.show_a();
n.get_b();
n.mul();
getch();
}
```

Output: a=5
b=7
a*b=35

VIVA QUESTIONS

1q:- Define inheritance.

2q:- What do you mean by derived class.

3q:- What do you mean by base class.

4q:- Define properties of inheritance.

5q:- why we use inheritance?

EXPERIMENT NO:14

Title: -- Write a program to demonstrate various types of inheritance

Objective:--To learn the inheritance concept by using its type..

S/W Requirement:--Turbo C++

H/W Requirement:-- An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

you require.

A CD-ROM disk drive

Theory:- inheritance is used to derived the properties of one class to another class and learn its various type.

Program:

/ Example of SINGLE Inheritance */*

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

class a{
int a;
public: int b;
void show_a(void)
{cout<<"a=";
cin>>a;
}
int get_a(void)
{return a;
}};

class b:public a
{int c;
```

```
public: void get_b(void)
    {cout<<"b=";
    cin>>b;
    }
```

```
void mul(void)
{c=b*get_a();
cout<<"a*b="<<c;
}};
```

```
void main()
{
b n;
n.show_a();
n.get_b();
n.mul();
getch();
}
```

Output: a=5
b=7
a*b=35

/*Example of MULTILEVEL Inheritance */

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

```
class student
{protected:int roll_no;
public: void print(void)
    {cout<<"Enter roll no.=";
    cin>>roll_no;
    }
};
```

```
class test:public student
{protected:float s1,s2;
public: void sub(void)
    {cout<<"enter the marks of subject1=";
```

```
    cin>>s1;
    cout<<"enter the marks of subject2=";
    cin>>s2;}
};
```

```
class result:public test
{protected:float total;
public:void total(void)
    {total=s1+s2;
    cout<<"total"<<total;
    }
    void display(void)
    {void print(void);
    void sub(void);
    }
};
```

```
void main()
{result a1;
a1.display();
a1.total();
}
```

Output:

```
Enter roll no.=123
enter the marks of subject1=34
enter the marks of subject2=45
total 79
```

/*Example of MULTIPLE Inheritance */

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

class student
{protected:int roll_no;
public:void print(void)
    {cout<<"Enter roll no.=";
    cin>>roll_no;
    }
};
```

```
class test
{protected:float s1,s2;
public:void sub(void)
    {cout<<"enter the marks of subject1=";
    cin>>s1;
    cout<<"enter the marks of subject2=";
    cin>>s2;}
};
```

```
class result:public student,public test
{protected:float total;
public:void total(void)
    {total=s1+s2;
    cout<<"total"<<total;
    }

    void display(void)
    {void print(void);
    void sub(void);
    }
};
```

```
void main()
{result a1;
a1.display();
a1.total();
}
```

Output: Enter roll no.=123
enter the marks of subject1=34
enter the marks of subject2=45
total 79

/* Example of HIERARCHICAL Inheritance */

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

```
class A
{protected:int a,b;
    float x,y;

};
```

```
class B:public A
{ public:void sum(void)
  {cout<<"a=";
   cin>>a;
   cout<<"b=";
   cin>>b;
   cout<<"sum="<<a+b;
  }};
```

```
class C:public A
{public:void pro(void)
  {cout<<"\nx=";
   cin>>x;
   cout<<"y=";
   cin>>y;
   cout<<"product="<<x*y;}
};
```

```
void main()
{B a1;
 C a2;
 a1.sum();
 a2.pro();
}
```

Output: a=8

b=6

sum=14

x=5

y=6

product=30

/* Example of HYBRID Inheritance */

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

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

```
class student
```

```
{protected:int roll_no;
```

```
public:void print(void)
```

```
  {cout<<"Enter roll no.=";
```

```
   cin>>roll_no;
```

```
  }
```

```
};
```

```
class test:public student
```

```
{protected:float s1,s2;
```

```
public:void sub(void)
```

```
        {cout<<"enter the marks of subject1=";  
        cin>>s1;  
        cout<<"enter the marks of subject2=";  
        cin>>s2;}  
};  
class sports  
{protected:float score;  
public:void put(void)  
    {cout<<"sports wt.=";  
    cin>>score;  
    }  
};  
class result:public test,public sports  
{protected:float total;  
public:void display(void)  
    {print();  
    sub();  
    }  
    void total1(void)  
    {total=s1+s2+score;  
    cout<<"total="<<total;  
    }  
};  
  
void main()  
{result a1;  
a1.display();  
a1.put();  
a1.total1();  
}
```

Output: Enter roll no.=123
enter the marks of subject1=34
enter the marks of subject2= 45
sports wt.=20
total=99

VIVA QUESTIONS

1q:- How many types of inheritance are their.

2q:- Define single inheritance.

3q:- Define multiple inheritance

4q:- Define hybrid inheritance.

5q:- Define multilevel inheritance

EXPERIMENT NO:15

Title: -- Write a program to demonstrate various operator unary and binary

Objective:--To learn the concept of operators such as unary(++ and binary(+)..

S/W Requirement:--Turbo C++

H/W Requirement:-- An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

you require.

A CD-ROM disk drive

Theory:- operators such as unary(++ and binary(+ description.

Program:

```
/*Overloading of a UNARY OPERATOR */
```

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

```
class unary  
{int x,y,z;  
public:void get_data(int a,int b,int c)  
    {x=a;  
    y=b;  
    z=c;  
    }  
void display(void);  
void operator-();  
};
```

```
void unary::display(void)  
{cout<<"\nx="<<x;  
cout<<"\ny="<<y;  
cout<<"\nz="<<z;  
}
```

```
void unary::operator-()
```

```
{x=-x;  
y=-y;  
z=-z;  
}
```

```
void main()  
{unary AKP;  
AKP.get_data(-5,9,-7);  
cout<<"S: ";  
AKP.display();  
-AKP;  
cout<<"\nNow the - operator will work on the above given values of S " ;  
cout<<"\nS: ";  
AKP.display();  
}
```

/*Overloading of a BINARY OPERATOR */

```
#include<iostream.h>  
  
class complex  
{float x,y;  
public:complex()  
    {}  
complex(float real,float img)  
{x=real;  
y=img;  
}  
complex operator+(complex);  
void display(void);  
};  
  
complex complex::operator+(complex c)  
{complex temp;  
temp.x=x+c.x;  
temp.y=y+c.y;  
return(temp);  
}  
  
void complex::display(void)  
{  
cout<<x<<" +j"<<y<<"\n";  
}
```

```
void main()
{
complex c1,c2,c3;
c1=complex(1.2,4.5);
c2=complex(3.4,9.5);
c3=c1+c2;
cout<<"c1= ";
c1.display();
cout<<"c2= ";
c2.display();
cout<<"c3= ";
c3.display();
}
```

Output: c1=1.2+j4.5
c2=3.4+j9.5
c3=4.6+j14.0

VIVA QUESTIONS

1q:- How binary operator are useful.

2q:- How unary operator are useful.

3q:- what is the importance of unary operator.

4q:- what is the importance of binary operator.

5q:- which operator is used as binary operator.

EXPERIMENT NO:16

Title: -- Write a program to demonstrate polymorphism concept.

Objective:--To learn the concept of polymorphism.

S/W Requirement:--Turbo C++

H/W Requirement:-- An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

you require.

A CD-ROM disk drive

Theory:- Polymorphism means many form.it implement same function in many form.

Program:

. Example of POLYMORPHISM-----Volume

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

```
int vol(int x,int y,int z)
{int v=x*y*z;
return v;
}
```

```
float vol(float x,float y,float z)
{float v=x*y*z;
return v;
}
```

```
void main()
{int a,b,c;
float l,m,n;
cout<<"Enter the side of rectangle (in integer values)";
cout<<"\na=";
```

```
cin>>a;
cout<<"b=";
cin>>b;
cout<<"c=";
cin>>c;
cout<<"volume(in integer)="<<vol(a,b,c);
cout<<"\nEnter the side of rectangle (in float values)";
cout<<"\n";
cin>>l;
cout<<"m=";
cin>>m;
cout<<"n=";
cin>>n;
cout<<"volume(in float)="<<vol(l,m,n);
}
```

Output:

Enter the side of rectangle (in integer values)

a=4

b=3

c=7

volume(in integer)=84

Enter the side of rectangle (in float values)

l=2

m=3

n=4

volume(in float)=24

VIVA QUESTIONS

1q:- Define polymorphism.

2q:- What do you mean by late binding.

3q:- What do you mean by early binding.

4q:- What is virtual function mean..

5q:- What is pure virtual function.

EXPERIMENT NO:17

Title: -- Write a program to demonstrate inline concept.

Objective:--To learn the concept of inline.

S/W Requirement:--Turbo C++

H/W Requirement:-- An 80386 or higher processor

8 MB of memory

Hard disk with enough space available to install the components

you require.

A CD-ROM disk drive

Theory:- Polymorphism means many form.it implement same function in many form.

program:

```
/*Example of INLINE FUNCTION */
```

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

```
inline float mul(float x,float y)
{return(x*y);
}
```

```
inline double div(float p,float q)
{return(p/q);
}
```

```
void main()
{float a,b;
cout<<"a=";
cin>>a;
cout<<"b=";
cin>>b;
cout<<"\na*b="<<mul(a,b);
cout<<"\na/b="<<div(a,b);
}
```


VIVA QUESTIONS

1q:- Define inline function.

2q:- Why we use inline function.

3q:- What is the importance of inline function.

4q:- What is friend function.

5q:- What is private access method means..