Object Oriented Programming In C++ Lab – Manual

OBJECT ORIENTED PROGRAMMING USING C++

LIST OF EXPERIMENTS

1. Write a program to check whether a given number is prime or not ?
2. Write a program to find the largest value from the given
   (a) Two numbers, (b) Three numbers, (c) Four numbers.
3. Write a program to find sum of first 100 natural numbers
   by using (a) for loop (b) while loop (c) do-while loop
4. Write a program to find sum of the following series using function declaration .
   Sum = $x - x^3/3! + x^5/5! - \ldots + (-1)^{(n-1)} x^{2n-1}/(2n-1)!$
5. Write a program to read the elements of the given two matrices and perform the matrix multiplication.
6. Write a program to exchange the contents of two variables by using
   (a) call by value (b) call by reference.
7. Write a program to perform the following arithmetic operations on complex numbers using a structure
   (a) addition of two complex nos. (b) Subtraction of two complex nos. (c) multiplication of two complex nos. (d) Division of two complex nos.
8. Write a program to generate a series of Fibonacci nos. using the constructor where the constructor member function had been defined
   (a) as a scope of class definition itself (b) out of the class definitions using the scope resolution operator. Also make this program with the help of copy constructor .
9. Write a program to demonstrate how ambiguity is avoided using scope resolution operator in the following inheritance (a) single inheritance (b) multiple inheritance.
10. Write a program to perform swapping of two data items of integer , floating point number and character type with the help of function overloading.
11. Write a program to generate a Fibonacci series by overloading
    (a) Prefix operator (b) postfix operator
12. Write a program to define a class time that will represent a time period in minutes and seconds. Overload the following operators for the following :
    (a) ++operator that will increment the seconds by 1.
    (b) + operator that will add two objects of time class.
13. Write a program to overload the comma operator for a class such that for the instruction a=(b,c) the larger object of b and c is assigned to a.
14. Write a program to access the private data of a class by non-member function through friend function where the friend function is declared as (a) the location of public category (b) the location of private category (c) within the scope of a class definition itself (d) Defined with inline code substitution.
15. Write a program to define two classes alpha and beta containing an integer each as data members. Define a function sum( ) that will be a friend to both alpha and beta which will take one object from each class as argument and return the sum of the data members of the argument objects.

16. Write a program to demonstrate how a pure virtual function defined and invoked from the object of derived class through the pointer of the base class.
17. Write a program to define a class named base that will contain a protected integer data member and inherit this class in class called derived, so that it returns the factorial of the base class member.
18. Write a program that will ask the user to enter a file name and a line of text and transfer that line of text into the text file.
19. Write a program to read and write class objects using file handling.
20. Write a program to Bubble sort using template function.
21. Write a program for invoking function that generates and handle exception.
22. Create a class FLOAT that has one data member float. Perform all arithmetic & relational operation for FLOAT class.
23. Write a program to define three functions with the same name area( ) , taking one , two & three arguments respectively. The function taking one argument will consider it as the side of a square and calculate area, the function with two argument will consider it the sides of a rectangle and the function with three arguments will consider will calculate the area of a cuboid.
24. Write a program to define a function a function power that will take two arguments base & exponent and return the value of base raised to the power exponent. The function should have a default value of base as 10.
25. Write a program to define a class Complex that will simulate the Complex numbers of mathematics, containing real and imaginary as the data members. Define suitable constructors and a function that will display the data members.
26. Write a program that will ask the user to input a file name and copy the contents of that file into another file.
27. Write a program that will ask the user to enter the details of 5 employees & transfer the details into a binary file named as Emp.dat. Write another file that will read the details and print it.
28. Write a program to define an abstract class Person that will contain the essential Information like name, age and sex of a person. Now derive two classes Student & Employee both from the class Person.The class Student will contain the academic information such as roll number; school etc. and the class Employee will contain information such as department and salary. In the main function declare & array of Person pointers that can hold the address of either Student or Employee object. The program will ask the user to enter the details of Students/Employees,create dynamic objects of these classes using new operator & store them in the array. The program will then display the contents of these objects.
29. Write a program to define a class Complex that will simulate the Complex numbers of mathematics,containing real & imaginary as the data members. Define suitable constructors and a function that will display the data members.
30. Create a class for making a student's mark-sheet & include a static data member total Marks to calculate average marks of a particular class.

Experiment No. 1

Aim: Write a program to check whether the number is prime or not ?

Description :

Prime number is a number that is divisible by 1 and itself.

Algorithm:

```
Step 1:  Start
Step 2:  Input  any integer in variable n.
Step 3:  initialize d = 1
step 4:  if i>1 then go to step 5 otherwise go to step 12.
step 5:  initialize i=2
step 6:  Repeat Steps 7,8,9 till i is less than or equal to n/2 otherwise go to Step 10.
step 7:  if n%i = 0 then d=0 and go to step 10
step 8:  i = i + 1
step 9:  go to step 5
Step 10:  If  d=1 then print number is prime.
Step 11:  If it is not then print number is not prime.
Step 12: Stop.
```

Program:

```
#include<iostream.h>
#include<conio.h>
void main ( )
  {
        int i , n , d;
        char a = 'y';
        while (a=='y')
          {
                cout<<"Enter any no. to check";
                cin>>n;
                d=1;
                if(n>1)
                  {
                        for(i=2;i<=n/2;i++)
                        {
                                if(n%i==0)
                                d=0;
                                break;
                         }
                        if(d==1)
                                cout<<"\nThe number is prime";
                        else
                                cout<<"\nThe number is not prime";
                  }
                 else
                        cout<<"The number is not prime";
                cout<<"\nWhether you want to continue(y/n)";
                cin>>a;
          }
        getche( );
  }
```

Output:

```
Enter any number to check: 6
The number is not prime
Whether you want to continue(y/n) : n
```

Experiment No. 2

Aim: Write a program to find the largest value from the given
(a) Two numbers, (b) Three numbers, (c) Four numbers.

Algorithm:

Algorithm for three numbers :

```
Step 1: Start
Step 2: Input any three numbers( for eg. a,b,c)
Step 3: Check if a is greater than b. then goto step 4 otherwise go to step5.
Step 4: Check  if a is greater than c. then print 'a is greatest' and go to Step 7 otherwise
        go to step 6.
Step 5: Check if b is greater than c. then print 'b is greatest' and go to Step 7.
Step 6: Print 'c is greatest'.
Step 7: Stop
```

Program:

(a)
```
#include<iostream.h>
void main( )
{
        float x , y;
        cout<<"Enter any two numbers\n";
        cin>>x>>y;
        if(x>y)
                cout<<"largest value is"<<x<<endl;
        else
                cout<<"largest value is:"<<y<<endl;
        getche( );
}
```

Output:

```
Enter any two numbers : 20 40
Largest value is : 40
```

(b)
```
#include<iostream.h>
void main( )
  {
        float x,y,z;
        cout<<"Enter any three numbers\n";
        cin>>x>>y>>z;
        if(x>y)
        {
          if(x>z) cout<<"Largest value is:"<<x<<endl;
          else cout<<"Largest value is:"<<z<<endl;
        }
```

```
                else if(y>z)  cout<<"Largest value is:"<<y<<endl;
                else cout<<"Largest valueis"<<z<<endl;
                getche( );
        }
```

Output:

```
Enter any three numbers
10 40 50
Largest value is 50
```

(c)

```
#include<iostream.h>
void main( )
{       float a,b,c,d;
        cout<<"Enter any four values";
        cin>>a>>b>>c>>d;
        if(a>b)
        {     if(a>c)
                 {      if(a>d) cout<<"Largest="<<a<<endl;
                        else cout<<"Largest="<<d<<endl;
                 }
              else
                {                   if(c>d) cout<<"Largest="<<c<<endl;
                        else cout<<"Largest="<<d<<endl;}
                }
        else if(b>c)
                {
                          if(b>d)  cout<<"Largest="<<b<<endl;
                         else  cout<<"Largest="<<d<<endl;
                }
        else
                {
                     if(c>d) cout<<"Largest="<<c<<endl;
                    else  cout<<"Largest="<<d<<endl;
                }
 }
        getche( );
    }
```

Output:

```
Enter any four numbers
10 30 40 60
Largest=60
```

Experiment No. 3

Aim: Write a program to find sum of first 100 natural numbers by using (a) for loop (b) while loop (c) do-while loop

Description :

Natural numbers are those numbers which is greater than 0 i.e. 1,2,3 and so on.
The do-while statement :

the do-while is an exit-controlled loop based on a condition,the control is transferred back to a particular point in the program.

```
 do
  {
      action1;
  }while(condition is true);
```

The while statement:This is also a loop structure, but is an entry-controlled one.

```
 while(condition is true)
   {
      action1;
   }
```

Algorithm :

```
Step 1: Start
Step 2: initialize sum = 0 , digit = 1.
Step 3: Repeat step 4, 5, 6 till i is less than or equal to 100
Step 4: sum = sum + digit
Step 5: digit = digit + 1
Step 6: go to step 3
Step 6: Print sum is sum of all natural no.s from 1 to 100.
Step 7: Stop
```

Program:

(a)

```
#include<iostream.h>
void main( )
{
        int sum ,i , digit;
        sum =0;         digit=1;
        for(i=0;i<100;i++)
          {
```

```
                        sum+=digit;
                        digit++;
                 }
                cout<<"The sum="<<sum;
                 getche( );
   }
```

(b)
```
#include<iostream.h>
void main( )
  {
        int sum , digit;
        sum =0;       digit=1;
        while(digit<=100)
          {
                sum+=digit;
                digit++;
          }
         cout<<"The sum="<<sum;
         getche( );
   }
```

(c)
```
#include<iostream.h>
void main( )
  {
        int sum ,i , digit;
        sum =0;
        digit=1;
    do
        {
                sum+=digit;
                digit++;
        }
        while(digit<=100);
        cout<<"The sum="<<sum;
        getche( );
}
```

Output:

```
The sum=5050
```

Assignment:

1. Write a program to check whether given no. is palindrome or not .
2. Write a program to sort the given data in ascending order.
3. Write a program to display the number in reverse order.

Viva-Voce Questions

1. What is object oriented programming languages?
2. What is the difference between POP and OOP?
3. What are the various data types used in C++ ?
4. What is user defined data types ?
5. What is the difference between control statement and selection statement ?

Experiment No.4

Aim: Write a program to find sum of the following series using function declaration .
Sum $=x - x^3/3! + x^5/5! - \ldots\ldots\ldots.. (-1)^{(n-1)} x^{2n-1}/(2n-1)!$

Description:

Functions in C++:

Function is a set of statements which perform any specific task. It returns only one value at a time. Dividing a program into functions is one of the major principles of top-down structured programming. Another advantage of using function is that it is possible to reduce the size of a program by calling and using them at different places in the program.

Function prototype :It describe the function interface to the compiler by giving details such as the number and type of arguments and the type of return value. It is a declaration statement in the calling program and is of the following form :

return-type   function-name(argument-list);

eg.: float volume(int x, int y, float z);

Algorithm :

```
Step 1: Start
Step 2: Input  any integer in variable n.
Step 3: Input  any integer in variable x.
Step 4: initialize i=3, sum=x,sign=1.
Step 5: Repeat step 6, 7, 8, 9, 10, 11 till i is less than or equal to n
Step 6: factval = call fact function passing  i as argument which will return the
          Factorial value of  passing argument.
Step 7: pow = call power function passing x and n as argument which will return
          x to the power n.
Step 8: sign = (-1) * sign
Step 9: temp = sign * pow/factval;
Step 10: sum = sum + temp;
Step 11: i = i + 2
Setp 12: print the value of sum which is sum of the series
            $x - x^3/3! + x^5/5! - \ldots\ldots\ldots.. (-1)^{(n-1)} x^{2n-1}/(2n-1)!$
Step 13: Stop
```

Program:

```
#include<iostream.h>
void main( )
  {
          long int fact(int);
          float power(float , int);
          float sum , temp , x , pow;
          int sign , i , n ; long int factval;
          cout<<”Enter the value of n”<<endl;
          cin>>n;
          cout<<”Enter the value of x”<<endl;
          cin>>x;
          i=3;     sum=x;          sign=1;


          while(i<=n)
            {
                    factval=fact(i);
                    pow=power(x,i);
                    sign=(-1)*sign;
                    temp=sign*pow/factval;
                    sum=sum+temp;       i=i+2;
            }
          cout<<”sum=”<<sum;
          getche( );
  }

long int fact(int max)
   {
          long int value; value=1;
          for(int i=1;i<=max;i++)
            {
                    value=value*1;
            }
          return(value);
  }

float power(float x , int n)
  {
          float value2;   value2=1;
          for(int j=1;j<=n;j++)
                    value2=value2*x;
          return(value2);
  }
```

Output:

```
Enter the value of n
7
Enter the value of x
```

```
1
Sum=0.841468
```

Assignment:

(4) Write a program to print the sum of the series :

Sum = $x^3/3! + x^5/5! + x^7/7! + \ldots\ldots\ldots\ldots x^{(2n+1)} / (2n+1)!$ terms .

Viva-Voce Questions

1. What is object-based language ?
2. What do you understand by data hiding and encapsulation?
3. What do you understand by polymorphism .
4. What do you understand by static binding and late binding?
5. What is loop and its types ?
6. What is difference between while loop and do-while loop ?

Experiment No. 5

Aim: Write a program to read the element of the given two matrix and to perform the matrix multiplication.

Description :

Multidimensional arrays are defined in the same manner as one dimensional array except that a separate pair of square brackets are required for each subscript .

The general format of the multidimensional array is :
Storage_class data_type arrayname [expression_1][expression_2]….

Algorithm :

```
Step 1: Start
Step 2: print "Enter the elements of the first 3*3 matrix a".
Step 3: initialize i = 0
Step 4: initialize j = 0.
Step 5: Repeat step 6, 7, 8, 9 till i is  less than or equal to 2.
Step 6: Repeat step 7, 8 till j is less than or equal to 2.
Step 7: Enter a value from keyboard in position a[i][j].
Step 8: j = j + 1.
Step 9: i = i + 1.
Step 10: print "Enter the elements of the second 3*3 matrix b".
Step 11: initialize i = 0
Step 12: initialize j = 0.
Step 13: Repeat step 14, 15, 16, 17 till i is  less than or equal to 2.
Step 14: Repeat step 15, 16 till j is less than or equal to 2.
Step 15: Enter a value from keyboard in position b[i][j].
Step 16: j = j + 1.
Step 17: i = i + 1.
Step 18: call the function mul( ) with arguments matrix a and b.
Step 19: inside mul( ) function declare a third array c[3][3].
Step 20: Repeat step 21, 22, 23, 24, 25, 26, 27 till i is  less than or equal to 2.
Step 21: Repeat step 22, 23, 24, 25, 26 till j is less than or equal to 2.
Step 22: c[i][j] = 0
Step 23: Repeat step 24, 25 till k is less than or equal to 2.
Step 24: compute  c[i][j]=c[i][j]+a[i][k]*b[k][j];
Step 25: k = k + 1.
Step 26: j = j + 1.
Step 27: i = i + 1.
Step 28: print the resulting matrix c[3][3] calling output( ) function
Step 29: inside output( ) function
Step 30: initialize i = 0
Step 31: initialize j = 0.
Step 32: Repeat step 33, 34, 35, 36 till i is  less than or equal to 2.
```

Step 33: Repeat step 34, 35 till j is less than or equal to 2.
Step 34: Enter a value from keyboard in position b[i][j].
Step 35: j = j + 1.
Step 36: i = i + 1.
Step 37: Stop

Program:

```
        #define MAX 100
        #include<iostream.h>
        void main(void)
          {     void output(float a[MAX][MAX],int n);
                void mul(float a[MAX][MAX] , b[MAX][MAX] , int n);
                float a[MAX][MAX] , b[MAX][MAX];
                int i , j , n;
                cout<<"Order of A matrix"<<endl;
                cin>>n;
                cout<<"Enter the elements of A matrix"<<endl;
                for(i=0;i<=n-1;++i)
                  {       for(j=0;j<=n-1;++j)
                                  cin>>a[i][j];
                  }
                cout<<"Order of B matrix"<<endl;
                cin>>n;
                cout<<"Enter the elements of B matrix"<<endl;
                for(i=0;i<=n-1;++i)
                  {    for(j=0;j<=n-1;++j)
                          cin>>b[i][j];
                   }
                cout<<"Output A[i][j]"<<endl;
                output(a,n);
                cout<<endl;
                cout<<"Output B[i][j]"<<endl;
                output(b,n);
                mul(a,b,n);
            }
void mul(float a[MAX][MAX],float b[MAX][MAX] , int n)
                {
                        void output(float c[MAX][MAX] , int n);
                        float c[MAX][MAX];
                        int i , j , k;
                        for(i=0;i<=n-1;++i)
                          {
                            for(j=0;j<=n-1;++j)
                                  {
                                          c[i][j]=0.0;
                                        for(k=0;k<=n-1;++k)
                                           c[i][j]=c[i][j]+a[i][k]*b[k][j];
                                  }
                            }
                         cout<<endl;
                         cout<<"Output of C[i][j] matrix"<<endl;
                         output(c,n);
                }

void output(float x[MAX][MAX] , int n)
                {
                         int i , j;
                         for(i=0;i<=n-1;++i)
                           {     for(j=0;j<=n-1;++j)
                                           cout<<x[i][j]<<'\t';
                                 cout<<endl;
                           }
                }
```

Output:

```
        Order of A matrix
        3
        Enter the elements of A matrix
        3       3       3
        3       3       3
        3       3       3

        Order of  B matrix
        3

        Enter the elements of B matrix
        3       3       3
        3       3       3
        3       3       3

        Output A[i][j]
        3       3       3
        3       3       3
        3       3       3

        Output B[i][j]
        3       3       3
        3       3       3
        3       3       3

        Output of C[i][j] matrix
        27      27      27
        27      27      27
        27      27      27
```

Assignment :

1. Write a program to read the element of the given two matrix and to perform the matrix subtraction.

Viva-Voce Questions

1. What is an Array.
2. Describe some properties of an Array.
3. what is difference between one and two dimensional Array.

Experiment No. 6

Aim: Write a program to exchange the contents of two variable by using (a) call by value (b) call by reference

Description :

An argument can be passed to a function by the following ways :

1) call-by-value :an argument is passed to the function.
2) call-by-reference: only the address of the argument is passed implicitly to the function.

Algorithm :

Algorithm for "Call by value"

```
Step 1: Start
Step 2: Input any two values (a,b)
Step3: Call the function swap by passing the value of two variables.
Step 4: Inside the function
        a. Exchange the values of two variable using a third pointer variable.
Step 5:  Print the values of  original variables  whose values were passed.
Step 6:  Print the values after calling swap function.
Step 7: Stop
```

Algorithm for "Call by reference"

```
Step 1: Start
Step 2: Input any two values (a,b)
Step3: Call the function swap by passing the addresses of two variables.
Step 4: Inside the function
        a. Exchange the values of two addresses using a third variable.
Step 5:  Print the values of  original variables  whose addresses were passed.
Step 6:  Print the values after calling swap function.
Step 7:  Stop.
```

Program:

(a)

```
#include<iostream.h>
void main( )
  {     int x,y;
        void swap(int , int);
        x=100;
        y=20;
        cout<<"values before swap"<<endl;
        cout<<"x="<<x<<"and y="<<y<<endl;
        swap(x,y);
        cout<<"values after swap"<<endl;
        cout<<"x="<<x<<"and y="<<y<<endl;
  }
```

```
void swap(int x, int y)
  {       int temp;
          temp=x;
          x=y;
          y=temp;
  }
```

Output:

```
Values before swap
x=100 and y=20
values after swap
x=100 and y=20
```

(b)

```
#include<iostream.h>
void main( )
{
        int x,y;
        void swap(int *x, int *y);
        x=100;
        y=20;
        cout<<”values before swap”<<endl;
        cout<<”x=”<<x<<”and y=”<<y<<endl;
        swap(&x,&y);
        cout<<”values after swap”<<endl;
        cout<<”x=”<<x<<”and y=”<<y<<endl;
}

void swap(int *x, int *y)
{
        int temp;
        temp=*x;
        *x=*y;
        *y=temp;
}
```

Ouput:

```
Values before swap
x=100 and y=20
values after swap
x=20 and y=100
```

Assignment :

6. Write a program to find the distance between two points using pointer.

Viva-Voce Questions

1. What is call by value and call by reference ?
2. What is pointer variable ?
3. what are the advantages from call by reference ?

Experiment No. 7

Aim: Write a program to perform the following arithmetic operations of a complex number using a structure
(a) addition of two complex no.
(b) Subtraction of two matrix
(c) multiplication of two complex no.
(d) Division of two complex numbers.

Description:

Structure is user-defined data_type in which we declare different type of data members. A structure can be passed to a function as a single variable.

Program:

```
struct complex
  {
        float real;
        float imag;
  };

complex add(complex a , complex b);
complex sub(complex a , complex b);
complex mul(complex a , complex b);
complex div(complex a , complex b);
#include<iostream.h>
#include<stdio.h>

void main( )
  {
        complex a , b ,c ;
        int ch;
        void menu(void);
        cout<<"Enter the first complex number\n";
        cin>>a.real>>a.imag;
        cout<<"Enter the second complex number\n";
        cin>>b.real>>b.imag;
        menu( );


        while((ch=getchar( ))!='q')
          {
            switch(ch)
                {
                case 'a':
                        c = add(a,b);
```

```
                                        cout<<"Addition of two complex numbers\n";
                                        cout<<c.real<<"+i"<<c.imag<<endl;
                                        break;
                               case 's':
                                        c=sub(a,b);
                                        cout<<"Subtraction of two complex numbers\n";
                                        cout<<c.real<<"+i"<<c.imag<<endl;
                                        break;
                              case 'm':
                                        c=mul(a,b);
                                        cout<<"Multiplication of two complex numbers\n";
                                        cout<<c.real<<"+i"<<c.imag<<endl;
                                        break;
                              case 'd':
                                        c=mul(a,b);
                                        cout<<"Division of two complex numbers\n";
                                        cout<<c.real<<"+i"<<c.imag<<endl;
                                        break;
                    }
                 }
              }
          void menu(void)
             {
                    cout<<"complex number operations\n";
                    cout<<"menu( )\n";
                    cout<<"a—addition \n";
                    cout<<s—subtraction \n";
                    cout<<m—multiplication \n";
                    cout<<"d—division \n";
                    cout<<"q—quit \n";
                    cout<<"option, please ?\n";
            }

          complex add(struct complex a , struct complex b)
            {
                    complex c;
                    c.real=a.real+b.real;
                    c.imag=a.imag+b.real;
                    return(c);
            }

          complex sub(struct complex a , struct complex b)
            {
                    complex c;
                    c.real=a.real-b.real;
                    c.imag=a.imag-b.real;
                    return(c);
            }
```

```
complex mul(struct complex a , struct complex b)
  {
        complex c;
        c.real=(a.real*b.real)-(a.imag*b.imag);
        c.imag=(a.real*b.imag)-(a.imag+b.real);
        return(c);
  }

complex div(struct complex a , struct complex b)
  {
        complex c;
        float temp;
        temp=(b.real*b.real)+(b.imag*b.imag);
        c.real=((a.real*b.real)+(a.imag*b.imag))/temp;
        c.imag=((b.real*a.imag)-(a.real*b.imag))/temp;
        return(c);
  }
```

Output:

```
Enter first complex number
1       1
Enter second complex number
2       2
Complex number operations
Menu( )
a—addition
s—subtraction
m—multiplication
d—division
q—quit
option , please?
a
Addition of two complex numbers 3+i3
q
```

Experiment No. 8

Aim: Write a program to generate a series of Fibonacci nos. using the constructor where the constructor member function had been defined (a) is a scope of class definition itself (b) out of the class definitions using the scope resolution operator . Also make this program with the help of copy constructor .

Description:

A constructor is a special member function whose main operation is to allocate the required resources such as memory and initialize the objects of its class. A constructor is distinct from other member function of the class and it has the same name as its class.It is executed automatically when a class instantiated.

```
class classname
 {
    .......
  public :
    classname(); //construtor
                     // prototype
 };


classname :: classname()
     {
         //constructor body
     }
```

A constructor has the following characteristics :-

1) It has same name as that of the class

2) it is executed automatically whenever the class is instantiated.

3) It does not have any return type.

4) It is normally used to initialize the data memebers of a class.

5) constructor can not be virtual.

Parameterized Constructor :

constructor can be invoked with arguments, just as in the case of functions.

Default Constructor :
The default constructor is a special member function which is invoked by the c++ compiler without any argument for initializing the object of a class. It may be explicitly written in a program. In case, a default constructor is not defined in a program, the c++ compiler automatically generates it in a program.

Copy Constructor :
Copy constructor are always used when the compiler has to create temporary object of a class object. The copy constructor are used in the following situations :

1) The initialization of an object by another object of the same class.
2) Return of objects as a function value.

The general format of copy constructor :

classname :: classname(classname & obj)

```
  class X
         {  .......
           public :
                X();
                X(X & obj); //copy constructor
         };
```

Algorithm :

```
1. Start
2. Create Fibonacci class in which declare and define its data member(f0,f1,fib) and
   member functions:  fibonacci( ),fibonacci(Fibonacci &ptr), increment( ), display( );
3. define constructors outside from the class.
   Inside default constructor :  Fibonacci( )
           fo=0;
           f1=1;
           fib=f0+f1;
4. Inside copy constructor : fibonacci(Fibonacci &ptr)
           f0=ptr.f0;
           f1=ptr.f1;
           fib=ptr.fib;
5. Inside Increment( ) function
           f0=f1;
           f1=fib;
           fib=f0+f1;
6. Inside display( ) function  :  we will display the value of  "fib".
7. Inside main() function we create an object of class Fibonacci as "number"
   Call  display and increment function for given terms.
8. Stop.
```

Program:

(a)
```
#include<iostream.h>
class Fibonacci
  {
```

```
                private:
                unsigned long int f0,f1,fib;
                public:
        fibonacci( )
          {
                fo=0;
                f1=1;
                fib=f0+f1;
          }

        fibonacci(Fibonacci &ptr)
          {
                f0=ptr.f0;
                f1=ptr.f1;
                fib=ptr.fib;
         }

        void increment( )
          {
                f0=f1;
                f1=fib;
                fib=f0+f1;
          }
        void display( )
          {
                cout<<fib<<"\t";
          }
        };

        void main(void)
          {     fibonacci number;
                for(int i=0;i<=15;i++)
                  {
                        number.display( );
                        number.increment( );
                  }
               getche( );
          }
```

(b) #include<iostream.h>

```
class Fibonacci
          {
                private:
                unsigned long int f0,f1,fib;
                public:
                fibonacci( );
                fibonacci(Fibonacci &ptr);
                void increment( );
                void display( );
          };
        fibonacci ::Fibonacci( )
          {
                fo=0;
                f1=1;
                fib=f0+f1;
          }
    fibonacci::fibonacci(Fibonacci &ptr)
          {
                f0=ptr.f0;
                f1=ptr.f1;
                fib=ptr.fib;
          }
    void fibonacci::increment( )
          {
                f0=f1;
                f1=fib;
                fib=f0+f1;
          }

     void fibonacci::display( )
                {
                        cout<<fib<<"\t";
                }
     void main(void)
          {
                fibonacci number;
                for(int i=0;i<=15;i++)
                  {
                        number.display( );
                        number.increment( );
                  }
                getche( );
          }
```

Output:

```
        1       2       3       5       8       13      21      34      55      89      144 233
                377     610     987     1597
```

Assignment :

1. Write a program to add complex numbers using overloaded constructor.

Viva-Voce Questions

1. What is a class ? How is it different from structure?

2. What are objects ? How are they created ?
3. What do you understand by data member and member function ?
4. What is static member function.Why we use it.
5. What is constructor ? Also explain types of constructor?
6. Why we use copy constructor ?
7. What is parameterized constructor ?
8. What is constructor overloading ?
9. What is destructor ? How is it invoked?

Experiment No. 9

Aim: Write a program to demonstrate how ambiguity is avoided using scope resolution operator in the following inheritance

(a) single inheritance

(b) multiple inheritance

Description:

Inheritance (Extending Classes)

The mechanism of deriving a new class from an old one is called inheritance. The old class is referred to as the base class and the new one is called the derived class.

The derived class inherits some or all properties of the base class.

Types of inheritance :

(1) Single inheritance :- A derived class with only one base class, is called single inheritance.

$$\begin{tikzcd} A \arrow[d, no head] \\ B \end{tikzcd}$$

2) Multiple inheritance : A derived class with several base class is called multiple inheritance.

$$\begin{tikzcd} A \arrow[dr, no head] & & B \arrow[dl, no head] \\ & C & \end{tikzcd}$$

3) Hierarchical inheritance :

The properties of one class may be inherited by more than one class. This process is known as hierarchical inheritance.

$$\begin{tikzcd} & A \arrow[dl, no head] \arrow[d, no head] \arrow[dr, no head] & \\ B & C & D \end{tikzcd}$$

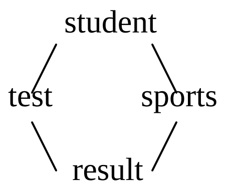
4) Multilevel inheritance :-

The mechanism of deriving a class from another derived class is known as multilevel inheritance.

$$\begin{tikzcd} A \arrow[d, no head] \\ B \arrow[d, no head] \\ C \end{tikzcd}$$

5) Hybrid inheritance :

There could be situations where we need to apply two or more types of inheritance to design a program. This is called hybrid inheritance.



Algorithm :

Algorithm for multiple inheritance :

1. Create a class baseA in which declare its data member(i) and member functions: getdata(int x), display( ).
2. Create a class baseB in which declare its data member(j) and member functions: getdata(int y), display( ).
3. Create a derived class derivedC, it will inherits both baseA and baseB class.
4. Define member functions of both base class baseA and baseB outside from the class.
5. Inside main function declare an object of derived class derivedC as “objc”.
6. Input a value x and pass it like this: objc.baseA::getdata(x);
7. Input a value y and pass it like this: objc.baseB::getdata(y)
8. call diplay function of baseA and then baseB.
9. Stop.

Program:

(a)
```
#include<iostream.h>
Class baseA
        {
                private:
                        int i;
                public:
                        void getdata(int x);
                        void display( );
        };
class baseB
        {
                private:
                        int j;
                public:
                        void getdata(int y);
                        void display( );
        };


Class derivedC:public baseA,public baseB
        {

        };

void baseA::getdata(int x)
        {
                i=x;
        }

void baseA::display( )
        {
                cout<<"value of i="<<i<<endl;
        }

 void baseB::getdata(int y)
         {
                j=y;
        }

void baseB::display( )
        {
                cout<<"value of j="<<j<<endl;
        }

void main( )
        {
                derivedC objc;
                int x,y;
                cout<<"enter the value for i\n";
                cin>>x;
                objc.baseA::getdata(x);
                cout<<"enter the value for j\n";
                cin>>y;
                objc.baseB::getdata(y);
                objc.baseA::display( );
                objc.baseB::display( );
        }
```

Output:

```
enter the value for i
10
```

```
enter the value for j
20
value of i = 10
value of j = 20
```

(b)

```
#include<iostream.h>
class baseA
    {
        public:
        int a;
    };
 class baseB
    {
        public:
          int a ;
    };
class baseC
    {   public:
          int a;
    };
class derivedD : public baseA , public baseB , public baseC
    {
        public:
        int a;
    };
 void main( )
    {
        derivedD objd;
        objd.a=10;
        objd.baseA::a=20;
        objd.baseB::a=30;
        objd.baseC::a=40;
        cout<<"value of a in the derived class="<<objd.a<<endl;
        cout<<"value of a in baseA="<<objd.baseA::a<<endl;
        cout<<"value of a in baseB="<<objd.baseB::a<<endl;
        cout<<"value of a in baseC="<<objd.baseC::a<<endl;
    }
```

Output:

```
Value of a in the derived class = 10
Value of a in baseA = 20
Value of a in baseB = 30
Value of a in baseC = 40
```

Assignment :

8. Write a program to display name , roll number , sex using single inheritance.
9. Write a program to display the multiplication of two numbers using single inheritance.
10. Write a program to display the details of students using multilevel inheritance.

Viva-Voce Questions

1. What is inheritance and its type ?
2. What is multilevel inheritance ?
3. What is multiple inheritance ?
4. Which type of problem we can face in hybrid inheritance ?

Experiment No. 10

Aim: Write a program to perform swapping of two data items of integer , floating point number and character type with the help of function overloading.

Description:

Function overloading refers to the use of the same thing for different purpose. C++ also permits overloading of functions. This means that we can use the same function name to create functions that perform a variety of different task. This is known as function overloading.

The function would perform different operations depending on the argument list in the function call. The correct function to be invoked is determined by checking the number and type of the arguments but not on the function type.

Declaration :

```
int add(int a,int b);
int add(int a,int b,int c);
```

Algorithm :

1. Declare prototypes with same function name “swap” three times with the arguments integer, float and char type respectively.
2. Inside first swap function with integer type arguments we will swap two int values with the help of third argument.
3. Inside second swap function with float type arguments we will swap two float values with the help of third argument.
4. Inside third swap function with char type arguments we will swap two character with the help of third argument.
5. Inside main( ) function
    (a) enter two integer, two float and two characters from the keyboard.
    (b) Call swap function three times for three type of data members.
    (c) Print the values before swap and after swap.
6. Stop.

Program:

```
#include<iostream.h>
void swap(int &ix, int &iy);
void swap(float &fx,float &fy);
void swap(char &cx, char &cy);

void swap(int &a,int &b)
  {
        int temp;
        temp=a;
        a=b;
        b=temp;
  }
void swap(float &a,float &b)
  {
        float temp;
        temp=a;
        a=b;
        b=temp;
  }
void swap(char &a,char &b)
```

```
        {
            char temp;
            temp=a;
            a=b;
            b=temp;
        }

void main(void)
        {
            int  ix,iy;
            float fx , fy;
            char cx , cy;
            cout<<"enter any two integers"<<endl;
            cin>>ix>>iy;
            cout<<"enter any two floating point numbers"<<endl;
            cin>>fx>>fy;
            cout<<"enter any two characters"<<endl;
            cin>>cx>>cy;
            cout<<"swapping of integers\n";
            cout<<"ix="<<ix<<"iy="<<iy<<endl;
            swap(ix,iy);
            cout<<"after swapping\n";
            cout<<"ix="<<ix<<"iy="<<iy<<endl;
            cout<<"swapping of floating point numbers\n";
            cout<<"fx="<<fx<<"fy="<<fy<<endl;
            swap(fx,fy);
            cout<<"after swapping\n";
            cout<<"fx="<<fx<<"fy="<<fy<<endl;
            cout<<"swapping of characters\n";
            cout<<cx="<<cx<<"cy="<<cy<<endl;
            swap(cx,cy);
            cout<<"after swapping\n";
            cout<<cx="<<cx<<"cy="<<cy<<endl;
        }
```

Output:

```
enter any two integers
100     200
enter any two floating point numbers
-11.11  22.22
enter any two characters
s       t
swapping of integers
ix = 100        iy=200
after swapping
ix=200 iy=100
swapping of floating point numbers
fx=-11.11       fy=22.22
```

```
after swapping
fx=22.22        fy=-11.11
swapping of characters
cx=s    cy=t
after swapping
cx=t    cy=s
```

Assignment:

11. Write a program to compute volume of various objects using function overloading.

Viva-Voce Questions

1. What is function overloading ?
2. What is Inline function ? What is its characteristics .
3. What is ambiguity problem and how it is avoided ?
4. What do you mean by swapping ?
5. How can we swap two values without using any third argument ?

Experiment No. 11

Aim: Write a program to generate a Fibonacci series by overloading
(a) Prefix operator (b) postfix operator

Description:

Operator overloading *:*

C++ has the ability to provide the operators with a special meaning for a data type. The mechanism of giving such special meaning to an operator is known as operator overloading. Operator overloading provides a flexible option for the creation of new definitions for most of the c++ operators.

We can overload all the c++ operators except the following :

=> class member access operator (.,.*)
=> scope resolution operator (::)
=> size operator (sizeof)
=> conditional operator (?:)

The ***general form*** of an operator function is :

```
returnType className :: operator op(a-l)
        {
          //function body
        }
```

Rules for overloading operator :

1) Only existing operators can be overloaded.
2) The overloaded operator must have at least one operand that is of user-defined type.
3) we can not change the basic meaning of an operator.
4) follwing operators can not be overloaded

sizeof	- size of variable
.	- membership operator
.*	- pointer to member operator
::	- scope resolution operator
?:	- conditional operator

5) we can not use friend function to overload certain operators. However member function can be used to overload them.

=	assignment operator
()	function call operator
[]	subscripting operator
->	class member access operator

6) a friend function will have only one argument for unary operators and two for binary operators.

7) a member function has no arguments for unary operators and only one for binary operators.

Algorithm :

```
1.  Start
2.  Create a fibonacci class in which declare its data member(f0,f1,f2) and member
        functions:  fibonacci( ), operator++( ), display( ).
3.  Define constructor outside from the class.
        Inside default constructor :  Fibonacci( )
                fo=0;
                f1=1;
                fib=f0+f1;
4. Define display( ) function :Inside display( ) function we will display the value of "fib".
5. Define operator++( ) function : Inside operator++( ) function we compute
                f0=f1;
                f1=fib;
                fib=f0+f1;
6.  Inside main() function
                a.  we create an object of class Fibonacci as "obj"
                b.  Input no. of terms in variable n
```

c. Call display function and operator function with the help of increment operator ++obj upto given no. of terms.

7. Stop.

Program:

(a)
```
#include<iostream.h>
class fibonacci
  {
        private:
        unsigned long int f0,f1,f2;
        public:
        fibonacci( );
        void operator++( );
        void display( );
  };

fibonacci::fibonacci( )
  {     f0=0;
        f1=1;
        fib=f0+f1;
  }

void fibonacci::display( )
  {     cout<<fib<<'\t';      }

void fibonacci::operator++( )
  {
        f0=f1;
        f1=fib;
        fib=f0+f1;
  }
void main( )
  {
        fibonacci obj;
        int n;
        cout<<"How many Fibonacci numbers are to be displayed?\n";
        cin>>n;
        for(int i=0;i<=n-1;++i)
        {
                obj.display( );
                ++obj;
        }
 }
```

(b)
```
#include<iostream.h>
```

```
        class fibonacci
          {
                private:
                unsigned long int f0,f1,f2;
                public:
                fibonacci( );
                void operator++(int);
                void display( );
          };

        fibonacci::fibonacci( )
          {     f0=0;
                f1=1;
                fib=f0+f1;
          }


void fibonacci::display( )
        {    cout<<fib<<'\t';   }

Void fibonacci::operator++(int x)
        {
                f0=f1;
                f1=fib;
                fib=f0+f1;
                return *this;
        }
void main( )
        {       fibonacci obj;
                int n;
                cout<<"How many Fibonacci numbers are to be displayed?\n";
                cin>>n;
                for(int i=0;i<=n-1;++i)
                        {       obj.display( );
                                Obj++;
                        }
        }
```

Output:

```
        How many Fibonacci numbers are to be displayed?
        5
        0       1       1       2       3
```

Experiment No. 12

Aim: Write a program to define a class time that will represent a time period in minutes and seconds. Overload the following operators for the following :
(a) ++operator that will increment the seconds by 1.
(b) + operator that will add two objects of time class.

Description:

Overloaded operator functions can be invoked by expression such as
for unary operator
op X; ex -> -X;

OR

X op; ex-> X-;

for binary operator

X op Y; ex -> Z = X + Y;

X + Y would be interpreted as

X.operator +(Y);

Program:

```
#include<iostream.h>
#include<conio.h>
class time
  {
          int min,sec;
       public:
       void getdata()
          {
               cout<<"Enter minute and second: ";
               cin>>min>>sec;
          }
       void display()
          {
                cout<<"\n minute :"<<min<< " sec :"<<sec;
          }
       time operator +(time);
       void operator ++( );
  };
void time :: operator ++()
  {
```

```
                sec = sec + 1;
       }
     time time :: operator +(time t)
       {
         time temp;
         temp.min = min + t.min;
         temp.sec = sec + t.sec;
         if(temp.sec>60)
           {
                 temp.min = temp.min + 1;
                 temp.sec = temp.sec - 60;
           }
        return(temp);
       }
     void main()
       {
               cout<<"Demonstrate ++ operator overloading"<<endl;
               time t;
               t.getdata();
               cout<<"Before Overloading : ";
               t.display();
               ++t;
               cout<<"\nAfter Overloading : ";
               t.display();
               cout<<"Demonstrate + operator overloading"<<endl;
               time t1,t2,tsum;
               t1.getdata();
               t2.getdata();
               tsum = t1 + t2;
          tsum.display();
          getch();
     }
```

Output:

```
Demonstrate ++ operator overloading
Enter minute and second 12 35
Before overloading
Minute : 12       Sec: 35
After overloading
Minute : 12       Sec: 36
Demonstrate + operator overloading
Enter minute and second : 4 40
Enter minute and second : 5 50
Minute : 10       Sec : 30
```

Experiment No.13

Aim: Write a program to overload the comma operator for a class such that for the instruction a = (b,c) the larger object of b and c is assigned to a.

Algorithm :

```
1. Create a class comma in which declare a data member( a ) and define member
   functions: comma(),comma(int x),display(),comma operator ,(comma).
2. Define operator function as member function outside from the comma class as
     a. Inside operator function create an object of comma class as "temp"
     b. check a >= c.a then set temp.a = a otherwise set temp.a = c.a.
     c. return temp.

3. Inside main( ) function
          i.   create two object c1 and c2 and pass initial value of their data member.
          ii.  create an extra object max, it will set with return object from comma
                 operator function.
          iii. Compute max = (c1,c2)
          iv.  Call display( ) function for each object.
4. Stop.
```

Program:

```
#include<iostream.h>
#include<conio.h>
class comma
  {
        int a;
        public:
                comma()
                     {}

            comma(int x)
                        {
                           a=x;
                         }

                void display()
                        {
                         cout<<a<<endl;
                        }
                comma operator ,(comma);
  };

comma comma::operator ,(comma c)
```

```
            {
              comma temp;
                   if(a>=c.a)
                       temp.a=a;
                 else
                         temp.a=c.a;

                 return(temp);
            }



          void main()
            {
                   comma c1(50),c2(24),max;
                   clrscr();
                   max=(c1,c2);
                   c1.display();
                   c2.display();
                   cout<<"maximum :";
                   max.display();
                   getch();
            }
```

Output:

```
50
24

Maximum : 50
```

Assignment:

12. Write a program to display the use of unary overloading operators.
13. Write a program to add two complex numbers using binary overloading operators.

Viva-Voce Questions

1. Explain Operator overloading ?
2. What are the rules of operator overloading ?
3. How many arguments are required in the definition of an overloaded unary operator.
4. How many arguments are required in the definition of an overloaded binary operator.

Experiment No. 14

Aim: Write a program to access the private data of a class by non-member function through friend function where the friend function is declared:

(a) in the location of public category (b) in the location of private category (c) within the scope of a class definition itself (d) Defined with inline code substitution.

Description:

Friend Function & Friend Class :

A friend function posses the following special characteristics :

1)The scope of a friend function is not limited to the class in which it has been declared as a friend.
2) A friend function can not be called using the object of that class.It can be invoked like a normal function without the use of any object.
3)Unlike class member functions, it cannot access the class member directly. However, it can use the object and the dot operator with each member name to access both the private and public members.
4) It can be declared in the private part or the public part of a class without affecting its meaning.

```
 class TestClass
           {
                int num1,num2;
            public :
               -------
               friend float sum(TestClass obj);
              --------
                --------
       };


   float sum(TestClass obj)
           {
              float result;
              result = obj.num1 + obj.num2;
            return result;
       }
```

Friend functions are useful in the following situations :-

1) Function operating on objects of two different classes.This is the ideal situation where the friend function can be used to bridge two class.
2) Friend function can be used to increase the versatility of overloaded operator.

Algorithm :

Algorithm where the friend function is declared in the location of public category :

1. Create a class sample in which declare a data member(x) and member functions: getdata( ), friend display(class sample).
2. Define member function getdata( ) in which take input a value in variable x.
3. Define friend function display( ) as ordinary c function in which display the entered no. x.
4. Inside main function
   a. create an object "obj" of sample class.
   b. Call getdata( ) function to enter a value from keyboard.
   c. Call friend function display( ) to access private data member of obj object.
5. Stop.

Program:

(a)
```
#include<iostream.h>
class sample
  {
   private:
         int x;
   public:
         void getdata( );
         friend void display(class sample);
  };

void sample::getdata( )
    {
         cout<<"enter the value of x\n";
         cin>>x;
    }

void display(class sample abc)
    {
         cout<<"Entered number is:"<<abc.x;
         cout<<endl;
    }

void main( )
    {
          sample obj;
          obj.getdata( );
          cout<<"accessing the private data by non-member function\n";
          display(obj);
    }
```

Output:

```
Enter the value of x
10
accessing the private data by non-member function
Entered number is : 10
```

(b)
```
#include<iostream.h>
class sample
  {
    private:
        int x;
        friend void display(class sample);
    public:
        void getdata( );
  };

void sample::getdata( )
  {
        cout<<"enter the value of x\n";
        cin>>x;
  }
void display(class sample abc)
  {
        cout<<"Entered number is:"<<abc.x;
        cout<<endl;
  }
void main( )
  {
        sample obj;
        obj.getdata( );
        cout<<"accessing the private data by non-member function\n";
        display(obj);
  }
```

Output:

```
Enter the value of x
10
accessing the private data by non-member function
Entered number is : 10
```

(c)
```
#include<iostream.h>
class sample
  {
        private:
                int x;
        public:
        inline void getdata( );
                friend void display(sample abc)
                  {
                    cout<<"Entered number is:"<<abc.x;
                    cout<<endl;
                  }
   };
```

```
inline void sample::getdata( )
          {
                    cout<<"enter the value of x\n";
                    cin>>x;
          }
void main( )
     {
          sample obj;
          obj.getdata( );
          cout<<"accessing the private data by non-member function\n";
          display(obj);
     }
```

Output:

```
Enter the value of x
10
accessing the private data by non-member function
Entered number is : 10
```

(d)

```
#include<iostream.h>
class sample
  {
          private:
                    int x;
          public:
                    inline void getdata( );
                    friend void display(class sample);\
  };

inline void sample::getdata( )
          {
                    cout<<"enter the value of x\n";
                    cin>>x;
          }
inline void display(class sample abc)
 {        cout<<"Entered number is:"<<abc.x;
          cout<<endl;
  }
void main( )
{
          sample obj;
          obj.getdata( );
          cout<<"accessing the private data by non-member function\n";
          display(obj);
}
```

Output:

```
Enter the value of x
10
accessing the private data by non-member function
Entered number is : 10
```

Experiment No.15

Aim: Write a program to define two classes alpha and beta containing an integer each as data members . Define a function sum( ) that will be a friend to both alpha and beta , that will take one object from each class as argument and return the sum of the data members of the argument objects.

Program:

```
#include<iostream.h>
class beta;
class alpha
    {       int a;
        public:
                 void getdata(int a1)
                     {   a=a1; }
    friend int sum(alpha,beta);
  };
class beta
{             int b;
    public :
    void getdata(int b1)
            {
               b=b1;
           }
    friend int sum(alpha,beta);
};
int sum(alpha x,beta y)
  {   return(x.a+y.b);    }

void main()
  {
            alpha alp;
            beta bet;
            clrscr();
            alp.getdata(900);
            bet.getdata(200);
            cout<<"\nSum : "<<sum(alp,bet);
            getch();
   }
```

Output

```
Sum : 1100
```

Experiment No.16

Aim: Write a program to demonstrate how a pure virtual function defined declared and invoked from the object of derived class through the pointer of the base class.

Description:

A Pure virtual function is a function declared in a base class that has no definition relative to the base class.
A “do-nothing” function may be defined as follows:
virtual void display( ) = 0;

Program:

```
#include<iostream.h>
class base
   {
            private:
                        int x;
                        float y;
            public:
                        virtual void getdata( );
                        virtual void display( );
 };

class derivedB:public base
   {
            private:
                        long int rollno;
                        char name[20];
            public:
                        void getdata( );
                        void display( );
  };
void base::getdata( )
   {
   }
 void base::display( )
    {
    }
void derivedB::getdata( )
    {
             cout<<”enter roll number of a student ?\n”;
             cin>>rollno;
             cout<<”enter name of the student?\n;
             cin>>name;
    }
 void derivedB::display( )
    {
```

```
                cout<<"Roll number  student's name\n";
                cout<<rollno<<'\t'<<name<<endl;
           }

     void main( )
           {
                 base *ptr;
                derivedB obj;
                 ptr->getdata( );
                 ptr->display( );
           }
```

Output:

```
enter roll number of a student?
98501
enter name of the student?
Ravi
Roll number              student's name
98501                    Ravi
```

Experiment No. 17

Aim: Write a program to define a class base that will contain a protected integer data member and inherit this class in class called derived , so that it returns the factorial of the base class member.

Algorithm :

1. Create a class base in which declare data members(n,i,mul) and member functions: display( ), getdata.
2. Create a derived class "derived" in which declare member functions: display( ), fact( ) Inherits base class in public mode.
3. Define member function getdata( ) of base class in which take input a value in variable n.
4. Define member function fact( )of derived class in which calculate the factorial value of variable n.
5. define member function display( ) of base class in which show the message "You have entered n :"
6. Define member function display( ) of derived class in which we call display function display() of base class and show the factorial value calculated in mul variable.
7. Inside main function
    a. Create an object "d" of derived class.
    b. Create an object "b" of base class.
    c. Call getdata( ) function from object d.
    d. Call fact( ) function from object d.
    e. Call display( ) function from object d.
8. Stop.

Program:

```
#include<iostream.h>
#include<conio.h>
class base
  { protected :
             int n,i,mul;
    public:
            void display();
            void getdata();
  };
class derived : public base
  { public:
    void display();
    void fact();
  };
void base:: getdata()
  {
    cout<<"Enter the value for N : ";
    cin>>n;
  }
void derived:: fact()
 {
    mul=1;
    for(int i=1;i<=n;i++)
    mul=mul*i;
 }

void base :: display()
    {
            cout<<"You have entered n : "<<n;
    }

void derived :: display()
   {
            base::display();
            cout<<"\nFactorial : "<<mul;
   }

void main()
{
    clrscr();
    derived d;
    base b;
    d.getdata();
    d.fact();
    d.display();
    getch();
}
```

Output :

```
        Enter the value for N: 4
        You have entered N : 4
        Factorial : 24
```

Assignment :

14. . Write a program to demonstrate virtual function and virtual destructor function.

Viva-Voce Questions

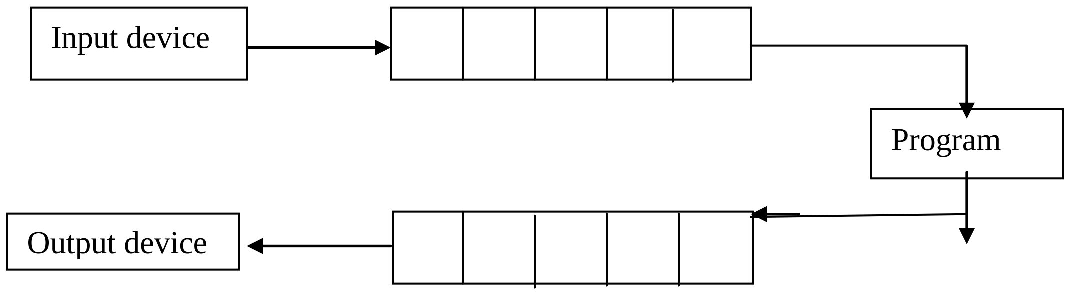
1. What is friend function ?
2. What are the characteristics of a friend function ?
3. What is virtual function ?
4. What is pure virtual function ?
5. What is abstract base class ?

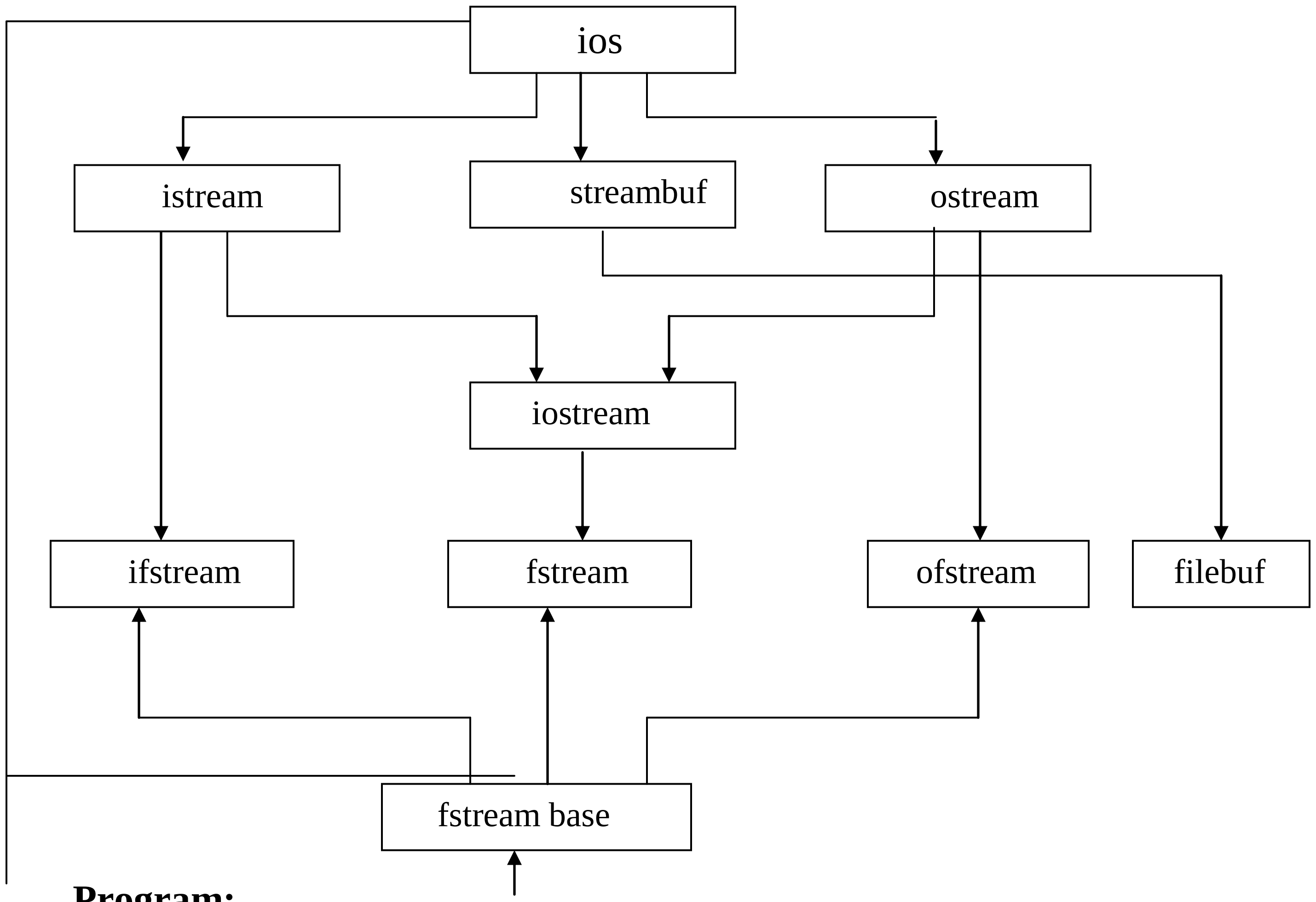
Experiment No. 18

Aim: Write a program that will ask the user to enter a file name and a line of text and transfer that line of text into a text file as named by the user.

Description:

C++ streams : A stream is a sequence of bytes. It acts either as a source from which the input data can be obtained or as a destination to which the output data can be sent. The source stream that provides data to the program is called the input stream and the destination stream that receives output from the program is called output stream.





Program:

```
#include<fstream.h>
```

```
class file
{
        char str[900];
    public :
        void take()
              {
                        cout<<"Enter Some Text : ";
                        cin.getline(str,900);
             }
    void show()
      {
            cout<<str;
      }
};
void main()
  {
    clrscr();
    file nw;
    fstream f1;
    nw.take();
    f1.open("test",ios::app);
    f1.write((char *)&nw,sizeof(nw));
    f1.close();
    fstream f2;
    f2.open("test",ios::in);
    while(!f2.eof())
    {
            f2.read((char *)&nw,sizeof(nw));
            nw.show();
    }
    f2.close();
    getch();
  }
```

Output:

```
Enter some text   Bhilai
Bhilai
```

Experiment No. 19

Aim: Write a program to read and write class objects using file handling.

Program :

```
#include<iostream.h>
#include<fstream.h>
#include<iomanip.h>
class INVENTORY
        {       char name[10];
                int code;
                float cost;
           public :
                 void readdata(void);
                 void writedata(void);
        };
void INVENTORY :: readdata(void)
        { cout<<"Enter Name:";cin>>name;
          cout<<"Enter Code:";cin>>code;
          cout<<"Enter cost:";cin>>cost;
        }
void INVENTORY :: writedata(void)
     {   cout<<setiosflags(ios::left);
         cout<<setw(10)<<name;
         cout<<setiosflags(ios::right);
         cout<<setw(10)<<code;
         cout<<setprecision(2);
         cout<<setw(10)<<cost;
         cout<<endl;
     }
void main()
        { INVENTORY item[3];
          fstream file;
          file.open("STOCK.DAT",ios::in|ios::out);
          cout<<"Enter details for three items\n";
          for(int i=0;i<3;i++)
              {       item[i].readdata();
                      file.write((char *) & item[i],sizeof(item[i]));
              }
          file.seekg(0);     //reset to start
          cout<<"\n OutPut \n\n";
          for(int i=0;i<3;i++)
             {     file.read((char *) & item[i],sizeof(item[i]));
                   item[i].writedata();
             }
         file.close( );       }
```

Output :

```
Enter Details for three items :
Enter Name : C++
Enter Code :  101
Enter Cost  :  175

Enter Name : Fortran
Enter Code :  102
Enter Cost  :  150

Enter Name : Java
Enter Code :  105
Enter Cost  :  225

    C++             101             175
    Fortran102              150
    Java            105             225
```

Assignment :

15. Write a program to work with multiple files.
16. Write a program to find the size of file using seekg and tellg.
17. Write a program to display “Divide Operation validation(Divide-by-zero)”.

Viva-Voce Questions

1. What is stream ?
2. What is the role of file() function.
3. What are the input and output streams ?
4. What are the steps involved in using a file in a C++ program?
5. Explain how while(fin) statement detects the end of a file that is connected to fin stream ?
6. Explain seekg( ) and seekp( ) function ?
7. Distinguish between tellg( ) and tellp( )?
8. What is a file mode ? Describe the various file mode options available.
9. Describe the various classes available for file operations.

Experiment No. 20

Aim: Write a program to Bubble sort using template function.

Description :

Bubble sort, sometimes shortened to **bubblesort**, also known as **exchange sort**, is a simple sorting algorithm. It works by repeatedly stepping through the list to be sorted, comparing two items at a time and swapping them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which means the list is sorted. The algorithm gets its name from the way smaller elements "bubble" to the top (i.e. the beginning) of the list via the swaps.

Algorithm :

1. Compare the first two elements in array say **a[1]**, and **a[2]**. If **a[2]** is smaller than **a[1]**, then interchange their values.
2. Compare **a[2]** and **a[3]**; interchange if **a[3]** is smaller than **a[2]**.
3. Continue this process till the last two elements are compared and interchanged.
4. Repeat the above steps **n-1** times.

Program:

```
#include<iostream.h>
template<class T>
void bubble(T a[ ]      , int n)
{
  for(int i=0;i<n-1;i++)
    {
          for(int j=n-1; j<i ; j++)
            {
                  if(a[j]<a[j-1])
                    {
                          swap(a[j],a[j-1]);
                    }
            }
    }
}
template<class X>
void swap(X &a, X &b)
        {
                X temp=a;
                a=b;
                b=temp;
        }
int main( )
{
        int x[5]={10,50,30,40,20};
        float y[5]={1.1,5.5,3.3,4.4,2.2};
        bubble(x,5);    //calls template function for int values
        bubble(y,5);    //calls template function for float values
        cout<<"Sorted x-array";
        for(int i=0;i<5;i++)
        cout<<x[i]<<"  ";       cout<<endl;
        cout<<"Sorted y-array";
        for(j=0;j<5;j++)
                cout<<y[j]<<"  "<<endl;
        return(0);
}
```

Output:

```
Sorted x-array:         10      20      30      40      50
Sorted y-array:         1.1     2.2     3.3     4.4     5.5
```

Experiment No. 21

Aim: Write a program for invoking function that generates and handle exception.

Description:

Exception handling is a programming language construct or computer hardware mechanism designed to handle the occurrence of some condition that changes the normal flow of execution. The condition is called an **exception**. Exceptions are used only for signaling error (exceptional) conditions.

The following keywords are used for handling error functions in C++ :

try
catch
throw

Program:

```
#include<iostream.h>
void divide(int x, int y , int z)
  {
        cout<<"\nWe are inside the function\n";
        if((x-y)!=0)
          {
                int R=z/(x-y);
                cout<<"Result="<<R<<"\n";
          }
        else
         {
                throw(x-y);
        }
  }
int main( )
  {
    try
        {
                cout<<"we are inside the try block\n";
                divide(10,20,30);       //Invoke divide( )
                divide(10,10,20);       //Invoke divide( )
        }
   catch(int i)
        {
                cout<<"caught the exception\n";
        }
   return(0);
  }
```

Output:

```
we are inside the try block
we are inside the function
Result = -3
we are inside the function
caught the exception
```

Assignment:

18. Write a program to demonstrate template function.
19. Write a program to demonstrate class template.
20. Write a program to demonstrate a program in which multiple catch Statements are used to handle various types of exceptions.

Viva-Voce Questions

1. What is containership ? How does it differ from inheritance?
2. What is this pointer? why we use it.
3. What do you understand by templates?
4. Distinguish between overloaded function and function templates.
5. Distinguish between the terms class template and template class.
6. What is an exception ?

Experiment No. 22

Aim: Create a class FLOAT that has one data member float. Perform all arithmetic & relational operation for FLOAT class.

Program

```
#include<iostream.h>
#include<conio.h>
class FLOAT
{
    float a,b;
    public:
    FLOAT(int a1,int b1)
    {
    a=a1;
    b=b1;
    }
    void Arithmetic()
    {
       float add,sub,mul,div;
       add=a+b;
       sub=a-b;
       mul=a*b;
       div=a/b;
       cout<<"Addition is"<<add;
       cout<<"Substraction is"<<sub;
      cout<<"Multiplication is"<<mul;
       cout<<"Division is"<<div;

    }
       void Relational()
       {
              if(a<b)
              {
                 cout<<b<<"is Greater";
              }
              else if(a>b)
                 cout<<a<<"is Greater";
              else
              cout<<a<<"is Equal to "<<b;
       }
};



void main()
{
  FLOAT f(1,2);
  clrscr();
  f.Arithmetic();
  f.Relational();
  getch();
}
```

Output:

```
Addition is          3
Substraction is      1
Multiplication is    2
Division is          2

2 is greater than 1
```

Experiment No. 23

Aim: Write a program to define three functions with the same name area( ) , taking one , two & three arguments respectively. The function taking one argument will consider it as the side of a square and calculate area, the function with two argument will consider it the sides of a rectangle and the function with three arguments will consider will calculate the area of a cuboid.

Program

```
#include<iostream.h>
#include<conio.h>
class area1
{    public:
    void area(int a)
    {       int ans;
            ans=a*a;
            cout<<"Area of a Square is"<<ans;
    }
    void area(int l,int b)
    {       int ans;
            ans=l*b;
            cout<<"Area of a rectangle is "<<ans;
    }
    void area(int l , int b, int h)
    {       int ans;
            ans=l*b*h;
            cout<<"Area of a Cuboid"<<ans;
    }
};
void main()
{   area1 a;
  clrscr();
  a.area(2);
  a.area(5,6);
  a.area(1,2,3);
  getch();
}
```

Output:

```
Area of a Square is          4
Area of a rectangle is       30
Area of a Cuboid             6
```

Experiment No. 24

Aim: Write a program to define a function a function power that will take two arguments base & exponent and return the value of base raised to the power exponent. The function should have a default value of base as 10.

Program

```
#include<iostream.h>
#include<conio.h>
int power(int b,int e)
{
    int i,s=1,b1;
    b1=b;
    for(i=1;i<e;i++)
    {

        s=b*b1;
        b=s;
    }
    return s;
}

void main()
{
   int b,e,ans;
   clrscr();
   cout<<"Enter The base and Exponent";
   cin>>b>>e;
   ans=power(b,e);
   cout<<ans;
   getch();
  }
```

Output:

```
Enter The base and Exponent 2  3
8
```

Experiment No. 25

Aim: Write a program to define a class Complex that will simulate the Complex numbers of mathematics, containing real and imaginary as the data members. Define suitable constructors and a function that will display the data members.

Program:

```
#include<iostream.h>
#include<math.h>
Class complex
{
private:
        float real;
        float imag;
public:
        complex()
                { real= imag = 0.0;
                }
complex(float real_in, float imag_in =0.0)
        {
                real = real_in;
                imag = imag_in;
        }
void show(char *msg)
        {       cout<<msg<<endl;
                if(imag < 0)
                        cout<<”-i:
                                else
                        cout<<”+i”;
                cout<<fabs(imag)<<endl;
        }
complex add(complex c2);
};
complex complex :: add(complex c2)
        {
                complex temp;
                temp.real= real + c2.real;
                temp.imag = imag + c2.imag;
                return(temp);
        }

void main( )
        { complex c1(1.5 ,2.0);
          complex c2(2.2);
          complex c3;
        c1.show(“c1=”);
        c2.show(“c2=”);
        c3=c1.add(c2);
        c3.show(“c3=c1.add(c2):”);
}
```

Output:

```
c1= 1.5 + i2
c2= 2.2 + i0;
```

```
c3= c1.add(c2): 3.7 + i2
```

Experiment No. 26

Aim: Write a program that will ask the user to input a file name and copy the contents of that file into another file.

Program

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

```
const int BUFFSIZE = 512;
int copyfile(char *sourcefile, char *destinationfile)
{
        fstream infile;
        fstream outfile;
        char buff[ BUFFSIZE + 1];
infile.open(sourcefile, ios::in| ios::binary);
if(infile.fail( ))
{
        cout<<"error :"<<sourcefile<<"non-existent";
        return(1);
}
outfile.open(destinationfile, ios::out| ios::binary);
if(outfile.fail( ))
{       cout<<"error:"<<destinationfile<<"unable to open";
        return(2);
}
while(!infile.eof( ))
{
        infile.read((char *) buff, BUFFSIZE);
        outfile.write((char *) buff, infile.gcount( ));
        if(infile.gcount( )<<BUFFSIZE;
        break;
}
infile.close( );
outfile.close( );
return(0);
}

void main( int argc, char argv[ ])
{
        cout<<"cp-copy file , 1996, raj\n"
        if(argc<3)


        {
        cout<<"Usage: cp<source file> <destination file>";
        exit(1);
        }
        if(copyfile(argv[1], argv[2]!=0)
        cout<<"\nfile copy operation failed.";
        }
```

Output:

```
cp- copyfile, 1996 , raj
```

Experiment No. 27

Aim: Write a program that will ask the user to enter the details of 5 employees & transfer the details into a binary file named as Emp.dat. Write another file that will read the details and print it.

Program

```
#include<iostream.h>
#include<fstream.h>
#include<iomanip.h>
const char *filename=”BINARY”;
int main( )
{
        char name[5][20]={ “Deepak” , “Raj”, “Tomy”, “Ramu”, “Rahul”};
        float salary[5]= { 10000, 5000, 8000 , 2000, 3300};
ofstream outfile;
outfile.open(“Emp.dat”);
outfile.write(char *) & name, sizeof(name));
outfile.write(char *) & salary, sizeof(salary));
outfile.close( );
for(int i=0 ; i<5 ; i++)
salary[i]=0;

ifstream infile;
infile.open(“Emp.dat”);
infile.read((char *)& name, sizeof(name));
infile.read((char *)& salary, sizeof(salary));

for(i=0; i<5;i++)
{
cout.setf(ios::showpoint);
cout<<setw(10)<<setprecision(2)<<name[i]<<salary[i];
}
infile.close( );
return(0);
}
```

Output:

```
Deepak   Raj    Tomy      Ramu      Rahul
10000    5000    8000      2000       3300
```

Experiment No. 28

Aim: Write a program to define an abstract class Person that will contain the Essential Information like name, age and sex of a person. Now derive two classes Student & Employee both from the class Person.The class Student will contain the academic information such as roll number; school etc. and the class Employee will contain information such as department and salary. In the main function declare & array of Person pointers that can hold the address of either Student or Employee object. The program will ask the user to enter the details of Students/Employees,create dynamic objects of these classes using new operator & store them in the array. The program will then display the contents of these objects.

Program

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

```
#include <iomanip.h>
class person
 {
private:
char name[20];
long int rollno;
char sex;
public:
void getdata();
void display();
};
class student {
private:
char course[20];
char semester[10];
int rank;
public:
void getdata();
void display();
};//end of class definition
class Employee : private person,private student
{
private:
float amount;
public:
void getdata();
void display();
}; //end of class definition
void person :: getdata()
{
cout<<"enter a name?\n";
cin>>name;
cout<<"enter roll no \n";
cin>>rollno;
cout<<"sex?\n";
cin >>sex;
}
void person::display()
{
cout<<name<<"     ";
cout<<rollno<<"    ";
cout<<sex<<"    ";
}
void student ::getdata()
{
cout<<"course name(B.E/MCA/DCA etc)?\n";
cin>>course;
cout<<"semester (Third/Fourth etc)?\n";
cin>>semester;
cout<<"rank of the student\n";
cin>>rank;
}
void student :: display()
{
cout <<course<<"     ";
cout <<semester<<"    ";
cout <<rank<< "        ";
}
void Employee :: getdata()
{
 person:: getdata();
 student :: getdata();
 cout<<"amount in rupees?\n";
 cin>>amount;
}
void student :: display()
{
 person :: display();
 student::display();
// cout<<setprecision(2);
 cout<<amount<<"    ";
}
void main()
{
person   f;
cout<<"enter the following information for";
cout<<"financial assistance\n";
f.getdata();
cout<<endl;
cout<<"Academic Performance for Financial Assistance\n";
cout<<"_______________________________________________\n";
cout<<"Name Rollno Sex Course Sememster Rank Amount\n";
cout<<"______________________________________________\n";
f.display();
cout<<endl;
cout<<"_______________________________________________\n";
}
```

Output:

```
Name of Employee: Rajesh
Sex :Male
Amount : 9000

Name of Student : Romesh
```

```
            Roll No : 121
            Sex : Male
            Course : B.E.
             Rank : 5th
```

Experiment No. 29

Aim: Write a program to define a class Complex that will simulate the Complex numbers of mathematics, containing real & imaginary as the data members. Define suitable constructors and a function that will display the data members.

Program

```
#include<iostream.h>
#include<conio.h>
class complex
{
  private:
  int x,y;
  public:
  complex(int a,int b)
  {
    x=a;
    y=b;
  }
  void display()
  {
```

```
           cout<<"\n complex no="<<x<<"+i"<<y;
          }
        };
        void main()
        {
         clrscr();
         int a,b;
         cout<<"\n enter real part";
         cin>>a;
         cout<<"\n enter imaginary part";
         cin>>b;
         complex c(a,b);
         c.display();
         getch();
        }
```

Output:

Complex no= 2.0 + I5.0

Experiment No. 30

Aim: Create a class for making a student's mark-sheet & include a static data member total marks to calculate average marks of a particular class.

Program

```
#include<iostream.h>
#include<conio.h>
class student
{  float n[6],ag;
  static double tm;
   public:
  void getdata();
  void average();
};
  double student::tm;
void student::getdata()
  {
   cout<<"\n enter your six subject marks";
   for(int  i=0;i<6;i++)
   cin>>n[i];
  }
  void student::average()
```

```
            {
             for(int i=0;i<6;i++)
             {
             tm=tm+n[i];
             }
             ag=tm/6;
             cout<<ag;
               }
          void main()
          {
            clrscr();
            student s1;
            s1.getdata();
            s1.average();
            getch();
          }
```

Output:

```
20  30  40 60 80 20
41.6
```

List of Assignment Questions

1. Write a program to check whether a given no. is palindrome or not .
2. Write a program to sort the given data in ascending order.
3. Write a program to display the number in reverse order.
4. Write a program to print the sum of the series :
   sum = $x^3/3! + x^5/5! + x^7/7! + \ldots\ldots\ldots\ldots x^{(2n+1)} / (2n+1)!$ terms .
5. Write a program to find the addition and subtraction of matrices.
6. Write a program to find the distance between two points using pointer.
7. Write a program to find the addition of two complex numbers by using constructor overloading.
8. Write a program to display name , roll number , sex using single inheritance.
9. Write a program to display the multiplication of two numbers using single inheritance.
10. Write a program to display the details of students using multilevel
11. Write a program to compute volume of various objects using function overloading.
12. Write a program to display the use of unary overloading operators.
13. Write a program to add two complex numbers using binary overloading operators.
14. Write a program to demonstrate virtual function and virtual destructor function.
15. Write a program to work with multiple files.
16. Write a program to find the size of file using seekg and tellg.
17. Write a program to display “Divide Operation validation(Divide-by-zero)”.
18. Write a program to demonstrate template function.
19. Write a program to demonstrate class template.
20. Write a program to demonstrate a program in which multiple catch

Statements are used to handle various types of exceptions.

CHHATTISGARH SWAMI VIVEKANAND TECHNICAL UNIVERSITY, BHILAI (C.G.)

Semester: IV Branch: Computer Science & Engg.

Subject: Object Oriented Concepts & Practical Code: 322423 (22)

Programming using C++ Lab Total Practical Period: 36

Total Marks in End Semester Exam: 40

1. Write a Program to check whether number is prime or not.
2. Write a Program to read number and to display the largest value between:
A. Two number B. Three Numbers C. Four number by using switch-case statements.
3. Write a Program to find sum of first natural numbers : sum= 1+2+3+4+……. 100 by using a. for loop b. while loop c. do-while loop
4. Write a Program to find sum of the following series using function declaration.
Sum= x-(x)3/3!+(x)5/5!-………..(x)n/n!
5. Write a Program to read the element of the given two matrix & to perform the matrix multiplication.
6. Write a Program to exchange the contents of two variable by using
(a) call by value (b) Call by reference.
7. Write a Program to perform the following arithmetic operations of a complex number using a structure (a). Addition of the two complex number (b). Subtraction of the two complex number (c). Multiplication of the two complex number (d). Division of the two complex number.
8. Write a Program to generate a series of Fibonacci Nos. using the constructor where the constructor member function had been defines (a). is the scope of class definition itself (b). out of the class definitions using the scope resolutions operator. Also make this program with the help of the copy constructor.
9. Write a Program to demonstrate how ambiguity is avoided using scope resolution operator in the following inheritance (a). Single inheritance (b). Multiple inheritance
10. Write a Program to perform the swapping of two data items of integer, floating point number and character type with the help of function overloading.
11. Write a Program to generate a Fibonacci series by overloading a. Prefix Operator b. Postfix Operator.
12. Write a Program to access the private data of a class by non-member function through friend function where the friend function is declared : (1). is the location of public category (2). is the location of private category (3). With in the scope of a class definition itself (4). Defined with inline code subtraction.
13. Write a Program to demonstrate how a pure virtual function defined declared and invoked from the object of derived class through the pointed of the base class.
14. Write a Program to Bubble Sort Using template function.
15. Write a Program for invoking for that Generate & Handle exception.

List of Equipment/Machine Required

Pentium IV machine, Turbo C++ compiler

Name of Text Books :

1. Programming with C++ : D Ravichandran
2. OOP's with C++ : E. Balaguruswamy .

Name of Reference Books:

1. Programming with C++ : Venugopal .

C H H A T T I S G A R H S W A M I
V I V E K A N A N D T E C H N I C A L
U N I V E R S I T Y
B H I L A I (C . G .)
Semester : **B.E. IV Sem.** Branch: **Electronics & Telecommunication**
Subject: **Programming Lab (C++ Language)** Code: **328423 (28)**
Total Practical Periods: **36** Total Marks in End Semester Examination: 4**0**

List of Programmes to be performed (but not less than 10)

1. Write a program to define three functions with the same name area( ) ,taking one , two and three arguments respectively. The function taking one argument will consider it as the side of a square and calculate area, the function with two argument will consider it the sides of a rectangle and the function with three arguments will consider will calculate the area of a cuboid.
2. Write a program to define a function a function power that will take two arguments base and exponent and return the value of base raised to the power exponent. The function should have a default value of base as 10.
3. Create a class FLOAT that has one data member float. Perform all arithmetic & relational operation for FLOAT class.
4. Write a program to define a class Complex that will simulate the Complex numbers of mathematics, containing real and imaginary as the data members. Define suitable constructors and a function that will display the data members.
5. Create a class for making a student's mark-sheet & include a static data member total marks to calculate average marks of a particular class.
6. Write a program to define two classes Alpha and Beta containing an integer each as data members. Define a function Sum( ) that will be a friend to both Alpha and Beta ,that will take one object from each class as argument and return the sum of the data members of the argument objects.
7. Write a Program to define a class Complex that will contain real and imaginary as the data members. Define appropriate constructors and a display function. Overload the binary + and the *operator to add and multiply two complex numbers respectively.
8. Write a program to define a class time that will represent a time period in minutes and seconds.
Overload the following operators for the following:
i. ++ Operator that will increment the seconds by 1
ii. + Operator that will add two objects of time class
9. Write a program to overload the comma operator for a class such that for the instruction a = (b,c) the larger object of and b is assigned to a
10. Write a program to define a class Base that will contain a protected integer data member and inherit this class in class called Derived. Override the display function of Base class and add a new member function in the Derived class so that it returns the factorial of the Base class member.
11. Write a program to define an abstract class Person that will contain the essential information like name, age and sex of a person. Now derive two classes Student and Employee both from the class Person. The class Student will contain the academic information such as roll number; school etc. and the class Employee will contain information such as department and salary. In the main function declare and array of Person pointers that can hold the address of either Student or Employee object. The

program will ask the user to enter the details of Students/Employees, create dynamic objects of these classes using new operator and store them in the array. The program will then display the contents of these objects.
12. Write a program to that will ask the user to enter a file name and a line of text and transfer that line of text into a text file as named by the user.
13. Write a program that will ask the user to input a file name and copy the contents of that file into another file.
14. Write a program that will ask the user to enter the details of 5 employees and transfer the details into a binary file named as Emp.dat. Write another file that will read the details and print it.
15. Write a program that will take the details of 10 students as input and transfer it into a binary file.Write another program that will be provide a menu to the user for the following purposes:
i. To display details of all the students
ii. To display details of all the students having total marks greater than a given value
iii. To sort the file on the basis of Roll number of students
iv. To sort the file on the basis of Total marks of students
v. To update the record for a particular student
vi. To delete the record for a particular student
vii. To search the details of a particular student on the basis of Roll number or Name
List of Equipments/Machine Required:
PCs, C++ Compiler

BHILAI (C.G.)

Semester: IV **Branch**: Information Technology

Subject: Object Oriented Concepts & Programming using C++ Lab **Practical Code**: 333424(33)

Total Practical Period: 36

Total Marks in End Semester Exam: 40

(minimum 10 experiments)

1. Write a Program to check whether number is prime or not.
2. Write a Program to read number and to display the largest value between:
A. Two numbers B. Three Numbers C. Four numbers by using switch-case statements.
3. Write a Program to find sum of first natural numbers : sum= 1+2+3+4+……. 100 by using a. for loop b.while loop c. do-while loop
4. Write a Program to find sum of the following series using function declaration.
Sum= x-(x)3/3!+(x)5/5!-………..(x)n/n!
5. Write a Program to read the element of the given two matrix & to perform the matrix multiplication.
6. Write a Program to exchange the contents of two variable by using
(a) call by value (b) Call by reference.
7. Write a Program to perform the following arithmetic operations of a complex number using a structure (a).Addition of the two complex numbers (b). Subtraction of the two complex numbers (c). Multiplication of the two complex numbers (d). Division of the two complex numbers.
8. Write a Program to generate a series of Fibonacci Nos. using the constructor where the constructor member function defines (a). is the scope of class definition itself (b). out of the class definitions using the scope resolutions operator. Also make this program with the help of the copy constructor.
9. Write a Program to demonstrate how ambiguity is avoided using scope resolution operator in the following inheritance (a). Single inheritance (b). Multiple inheritance
10. Write a Program to perform the swapping of two data items of integer, floating point number and character type with the help of function overloading.
11. Write a Program to generate a Fibonacci series by overloading a. Prefix Operator b. Postfix Operator.
12. Write a Program to access the private data of a class by non-member function through friend function where the friend function is declared : (1). is the location of public category (2). is the location of private category (3). With in the scope of a class definition itself
(4). Defined with inline code subtraction.
13. Write a Program to demonstrate how a pure virtual function defined declared and invoked from the object of derived class through the pointer of the base class.
14. Write a Program to Bubble Sort Using template function.
15. Write a Program for invoking Generate & Handle exception.

Name of Text Books :

1. OOP’s with C++ : E. Balaguruswamy .
2. OOP with C++ : Robert Lafore

Name of Reference Books:

1. Programming with C++ : Venugopal .
2. Object Oriented Programming in C++ : StroutStrups.

CHATTISGARH SWAMI VIVEKANAND UNIVERSITY , BHILAI

Semester: **MCA 3rd Sem.** Branch : Computer Applications

Subject: **Programming Lab in C++** Practical Code : 521321(21)

Total Practical Periods : 36 Total Marks in End Semester Exam : 50

Experiments to be performed: (minimum 10 experiments)

(i) Write a program to define a class Book that will contain Title, Author and Price of the book as data members. Define Null constructor, Parameterized constructor and copy constructor for the class and a function to display the details of an object. Use the new operator to initialize object of this class through a pointer and display the data member through a member function.

(ii) Write a program to define two classes Alpha and Beta containing an integer each as data members. Define a function Sum( ) that will be a friend to both Alpha and Beta ,that will take one object from each class as argument and return the sum of the data members of the argument objects.

(iii) Write a program to define a class Sample containing a static data member count that will maintain the total number of objects of this class initialized so far.

(iv) Write a Program to define a class Complex that will contain real and imaginary as the data members. Define appropriate constructors and a display function. Overload the binary + and the * operator to add and multiply two complex numbers respectively.

(v) Write a program to define a class time that will represent a time period in minutes and seconds. Define member functions and overload the following operators for the following:

_ ++ Operator that will increment the seconds by 1

_ + Operator that will add two objects of time class

(vi) Write a program to define a class Array that will contain an array of integers as a private data member of the class. Overload the subcript operator [ ] so that it will take an integer index as an argument and return the reference of element at that index in the array.

(vii) Write a program to overload the comma operator for a class such that for the instruction a = (b,c) the larger object of 'c' and 'b' is assigned to 'a'.

(viii) Write a program to define a class Base that will contain a protected integer data member and inherit this class in class called Derived. Override the display function of Base class and add a new member function in the Derived class so that it returns the factorial of the Base class member.

(ix) Write a program to define a class Two dimensional that will represent a point in the plane by its x and y coordinates. The class will contain constructors and member function that can calculate the distance between any two points in the plane. Derive a new class Three dimensional from the class Two dimensional that will add a new member, the z coordinate. Override the function that calculates distance so that kit can calculate the distance between two points in the space.

(x) Write a program to define an abstract class Person that will contain the essential information like name, age and sex of a person. Now derive two classes Student and Employee both from the class Person. The class Student will contain the academic information such as roll number; school etc. and the class Employee will contain information such as department and salary. In the main function declare and array of Person pointers that can hold the address of either Student or Employee object. The program will ask the user to enter the details of Students/Employees, create dynamic objects of these classes using new operator and store them in the array. The program will then display the contents of these objects.

(xi) Write a program to read the contents of a text file and count the number of characters read from the file.

(xii) Write a program that will ask the user to input a file name and copy the contents of that file into another file.

(xiii) Write a program that will ask the user to enter the details of 5 students and transfer those details into a binary file Stud.dat. Write another file that will read the details of the students and print the names of all those students who have total marks greater that a particular given value.
(xiv) Write a program that will take the details of 10 students as input and transfer it into a binary file. Write another program that will be provide a menu to the user for the following purposes:
_ To display details of all the students
_ To display details of all the students having total marks greater than a given value
_ To sort the file on the basis of Roll number of students
_ To sort the file on the basis of Total marks of students
_ To update the record for a particular student
_ To delete the record for a particular student
_ To search the details of a particular student on the basis of Roll number or Name
(xv) Write a program that will take any number of integers from the command line as argument and print the sum of all those integers.
List of Equipments/Machine required :
(i) PC with Wndows xp
(ii) Turbo C++ compiler
Recommended Books :
(i) K.R.Venugopal, Raj Kumar & T.Ravi Shankar : Mastering C++ ,TMH pub.
(ii) H. Schildt :C++ complete reference, TMH