### A). C++ Basic Example Programs

**1). Hello World C++ Example Program**

/\*  Example Program For Hello World In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

// Header Files

#include<iostream>

#include<conio.h>

//Standard namespace declaration

using namespace std;

//Main Function

int main()

{

     //Standard Ouput Statement

     cout<<"My First C++ Program";

     // Wait For Output Screen

     getch();

     //Main Function return Statement

     return 0;

}

### 2). if Statement Example Program in C++

/\*  Example Program For If Statement In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

int main()

{

     // Variable Declaration

     int a;

     //Get Input Value

     cout<<"Enter the Number :";

     cin>>a;

     //If Condition Check

     if(a > 10)

     {

        // Block For Condition Success

        cout<<a<<" Is Greater than 10";

     }

     // Wait For Output Screen

     getch();

     return 0;

 }

### 3). if..else Statement Example Program In C++

/\*  Example Program For if...else Statement In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

int main()

{

     // Variable Declaration

     int a;

     // Get Input Value

     cout<<"Enter the Number :";

     cin>>a;

     //If Condition Check

     if(a > 10)

     {

        // Block For Condition Success

        cout<<a<<" Is Greater than 10";

     }

     else

     {

       // Block For Condition Fail

       cout<<a<<" Is Less than/Equal 10";

     }

     // Wait For Output Screen

     getch();

     return 0;

 }

### 4). For Loop Example Program In C++

/\*  Example Program For for Loop In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

int main()

{

     // Variable Declaration

     int a;

     // Get Input Value

     cout<<"Enter the Number :";

     cin>>a;

     //for Loop Block

     for (int counter = 1; counter <= a; counter++)

     {

         cout<<"Execute "<<counter<<" time"<<endl;

     }

     // Wait For Output Screen

     getch();

     return 0;

 }

### 5). While Loop Example Program In C++

/\*  Example Program For While Loop In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

int main()

{

     // Variable Declaration

     int a;

     // Get Input Value

     cout<<"Enter the Number :";

     cin>>a;

     int counter = 1;

     //while Loop Block

     while (counter <= a)

     {

         cout<<"Execute While "<<counter<<" time"<<endl;

         counter++;

     }

     // Wait For Output Screen

     getch();

     return 0;

 }

### 6). Do While Loop Example Program In C++

/\*  Example Program For Do..While In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

int main()

{

     // Variable Declaration

     int a;

     // Get Input Value

     cout<<"Enter the Number :";

     cin>>a;

     int counter = 1;

     //Do while Loop Block

     do

     {

         cout<<"Execute Do While "<<counter<<" time"<<endl;

         counter++;

     }while (counter <= a);

     // Wait For Output Screen

     getch();

     return 0;

 }

### B). C++ Common Example Programs

### 1). Factorial Using Loop Example Program In C++

/\*  Example Program For Factorial Value Using For Loop In C++

    little drops @ thiyagaraaj.com

6! = 6 x 5 x 4 x 3 x 2 x 1 = 720

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

int main()

{

     // Variable Declaration

     int counter, n, fact = 1;

     // Get Input Value

     cout<<"Enter the Number :";

     cin>>n;

     //for Loop Block

     for (int counter = 1; counter <= n; counter++)

     {

         fact = fact \* counter;

     }

     cout<<n<<" Factorial Value Is "<<fact;

     // Wait For Output Screen

     getch();

     return 0;

 }

### 2). Factorial Using Function Example Program In C++

/\*  Example Program For Factorial Value Using Function In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

//Function

long factorial(int);

int main()

{

     // Variable Declaration

     int counter, n;

     // Get Input Value

     cout<<"Enter the Number :";

     cin>>n;

     // Factorial Function Call

     cout<<n<<" Factorial Value Is "<<factorial(n);

     // Wait For Output Screen

     getch();

     return 0;

 }

// Factorial Function

long factorial(int n)

{

    int counter;

    long fact = 1;

     //for Loop Block

     for (int counter = 1; counter <= n; counter++)

     {

         fact = fact \* counter;

     }

  return fact;

}

### 3). Factorial Using Recursion Example Program In C++

/\*  Example Program For Factorial Value Using Recursion In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

//Function

long factorial(int);

int main()

{

     // Variable Declaration

     int counter, n;

     // Get Input Value

     cout<<"Enter the Number :";

     cin>>n;

     // Factorial Function Call

     cout<<n<<" Factorial Value Is "<<factorial(n);

     // Wait For Output Screen

     getch();

     return 0;

 }

// Factorial recursion Function

long factorial(int n)

{

  if (n == 0)

    return 1;

  else

    return(n \* factorial(n-1));

}

### 4). Find Prime Number ( Method1 ) Example Program In C++

/\*  Example Program For Find Prime Number Using For Loop In C++

    little drops @ thiyagaraaj.com

5 is prime, as only 1 and 5 divide it.

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

#include<math.h>    // Math.h For sqrt function

using namespace std;

int main()

{

     // Variable Declaration

     int n;

     // Get Input Value

     cout<<"Enter the Number :";

     cin>>n;

      cout<<"List Of Prime Numbers Below "<<n<<endl;

     //for Loop Block For Find Prime Number

     for (int i=2; i<n; i++)

        for (int j=2; j\*j<=i; j++)

        {

            if (i % j == 0)

                break;

            else if (j+1 > sqrt(i)) {

                cout << i << endl;

            }

        }

     // Wait For Output Screen

     getch();

     return 0;

 }

### 5). Find Prime Number ( Method2 ) Example Program In C++

/\*  Example Program For Find Prime Number Using For Loop In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

#include<math.h>    // Math.h For sqrt function

using namespace std;

int main()

{

     // Variable Declaration

     int n;

     // Get Input Value

     cout<<"Enter the Number :";

     cin>>n;

      cout<<"List Of Prime Numbers Below "<<n<<endl;

     //for Loop Block For Find Prime Number

     for (int i=2; i<n; i++)

     {

        bool prime=true;

        for (int j=2; j\*j<=i; j++)

        {

            if (i % j == 0)

            {

                prime=false;

                break;

            }

        }

        if(prime) cout << i << endl;

      }

     // Wait For Output Screen

     getch();

     return 0;

 }

### 6). Fibonacci series Example Program In C++

/\*  Example Program For Fibonacci Series Using Loop In C++

    little drops @ thiyagaraaj.com

1  1  2  3  5  8  13  21  34  55  89  144 ......

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

int main()

{

     // Variable Declaration

     int counter, n;

     long last=1,next=0,sum;

     // Get Input Value

     cout<<"Enter the Number :";

     cin>>n;

     //Fibonacci Series Calculation

     while(next<n/2)

     {

      cout<<last <<"  ";

      sum=next+last;

      next=last;

      last=sum;

     }

     // Wait For Output Screen

     getch();

     return 0;

 }

**C). C++ Class Example Programs**

**1). Simple Class Example Program In C++**

/\*  Example Program Simple Class Example Program In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP                             \*/

// Header Files

#include <iostream>

#include<conio.h>

using namespace std;

// Class Declaration

class person

{

//Access - Specifier

public:

//Varibale Declaration

  string name;

  int number;

};

//Main Function

int main()

{

    // Object Creation For Class

    person obj;

    //Get Input Values For Object Varibales

    cout<<"Enter the Name :";

    cin>>obj.name;

    cout<<"Enter the Number :";

    cin>>obj.number;

    //Show the Output

    cout << obj.name << ": " << obj.number << endl;

    getch();

    return 0;

}

**2). Simple Class Example Program For Find Prime Number In C++**

/\*  Example Program For Simple Class Example Program For Prime Number In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

// Class Declaration

class prime

{

    //Member Varibale Declaration

int a,k,i;

public:

prime(int x)

{

a=x;

}

// Object Creation For Class

void calculate()

{

k=1;

{

for(i=2;i<=a/2;i++)

if(a%i==0)

{

k=0;

break;

}

else

{

k=1;

}

}

}

void show()

{

if(k==1)

cout<<"\n"<<a<<" is Prime Number.";

else

cout<<"\n"<<a<<" is Not Prime Numbers.";

}

};

//Main Function

int main()

{

int a;

cout<<"Enter the Number:";

cin>>a;

// Object Creation For Class

prime obj(a);

// Call Member Functions

obj.calculate();

obj.show();

getch();

return 0;

}

**D). Namespace Example Programs In C++**

**1). Simple Example Program For Namespace In C++**

## Definition

*A namespace (sometimes also called a name scope) is an abstract container or environment created to hold a logical grouping of unique identifiers or symbols (i.e., names). An identifier defined in a namespace is associated only with that namespace. The same identifier can be independently defined in multiple namespaces. That is, the meaning associated with an identifier defined in one namespace may or may not have the same meaning as the same identifier defined in another namespace. Languages that support namespaces specify the rules that determine to which namespace an identifier (not its definition) belongs.*

*The functionality of namespaces is especially useful in the case that there is a possibility that a global object or function uses the same identifier as another one, causing redefinition errors.*

## Syntax

### Declaration:

namespace abc {

 int variable;

}

### Adding Namespace:

using namespace abc;

### Usage Namespace Member:

abc::variable

## Example Program

/\*  Example Program For namespace Example In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include <iostream>

using namespace std;

//Namespace namespacefirst

namespace namespacefirst

{

  int value = 5;

}

//Namespace namespacesecond

namespace namespacesecond

{

  double value = 3.1416;

}

int main () {

  //Namespace namespacefirst Varibale Usage

  cout << "namespacefirst value : " <<namespacefirst::value << endl;

  //Namespace namespacesecond Varibale Usage

  cout << "namespacesecond value : "<<namespacesecond::value << endl;

  return 0;

}

## Sample Output:

namespacefirst value : 5

namespacesecond value : 3.1416

**E). C++ Constructor Example Programs**

**1). Simple Example Program For Constructor In C++**

/\*  Example Program For Simple Example Program Of Constructor In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

class Example        {

    // Variable Declaration

    int a,b;

    public:

    //Constructor

    Example()            {

    // Assign Values In Constructor

    a=10;

    b=20;

    cout<<"Im Constructor\n";

    }

    void Display()    {

    cout<<"Values :"<<a<<"\t"<<b;

    }

};

int main()                {

        Example Object;

        // Constructor invoked.

        Object.Display();

        // Wait For Output Screen

        getch();

        return 0;

}

## Sample Output

Im Constructor

Values :10      20

**2). Simple Example Program For Parameterized Constructor In C++**

/\*  Example Program For Simple Example Program Of Parameterized Constructor In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

class Example        {

    // Variable Declaration

    int a,b;

    public:

    //Constructor

    Example(int x,int y)            {

    // Assign Values In Constructor

    a=x;

    b=y;

    cout<<"Im Constructor\n";

    }

    void Display()    {

    cout<<"Values :"<<a<<"\t"<<b;

    }

};

int main()                {

        Example Object(10,20);

        // Constructor invoked.

        Object.Display();

        // Wait For Output Screen

        getch();

        return 0;

}

## Sample Output

Im Constructor

Values :10      20

**3). Simple Example Program For Constructor Overloading In C++**

/\*  Example Program For Simple Example Program Of Constructor Overloading In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

class Example        {

    // Variable Declaration

    int a,b;

    public:

    //Constructor wuithout Argument

    Example()            {

    // Assign Values In Constructor

    a=50;

    b=100;

    cout<<"\nIm Constructor";

    }

    //Constructor with Argument

    Example(int x,int y)            {

    // Assign Values In Constructor

    a=x;

    b=y;

    cout<<"\nIm Constructor";

    }

    void Display()    {

    cout<<"\nValues :"<<a<<"\t"<<b;

    }

};

int main()                {

        Example Object(10,20);

        Example Object2;

        // Constructor invoked.

        Object.Display();

        Object2.Display();

        // Wait For Output Screen

        getch();

        return 0;

}

## Sample Output

Im Constructor

Im Constructor

Values :10      20

Values :50      100

**4). Simple Constructor Example Program For Find Prime Number In C++**

## Algorithm:

STEP 1:  Start the program.

STEP 2:  Declare the class as Prime with data members,

                  Member functions.

STEP 3:  Consider the argument constructor Prime() with integer

                  Argument.

STEP 4:  To cal the function calculate() and do the following steps.

STEP 5:  For i=2 to a/2 do

STEP 6:  Check if a%i==0 then set k=0 and break.

STEP 7:  Else set k value as 1.

STEP 8:  Increment the value i as 1.

STEP 9:  Check whether the k value is 1 or 0.

STEP 10:If it is 1 then display the value is a prime number.

STEP 11:Else display the value is not prime.

STEP 12:Stop the program.

## Example Program

/\*  Example Program For Simple Constructor Example Program For Prime Number Overloading In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

// Class Declaration

class prime

{

    //Member Varibale Declaration

int a,k,i;

public:

prime(int x)

{

a=x;

k=1;

{

for(i=2;i<=a/2;i++)

if(a%i==0)

{

k=0;

break;

}

else

{

k=1;

}

}

}

void show()

{

if(k==1)

cout<<"\n"<<a<<" is Prime Number.";

else

cout<<"\n"<<a<<" is Not Prime Numbers.";

}

};

//Main Function

int main()

{

int a;

cout<<"Enter the Number:";

cin>>a;

// Object Creation For Class

prime obj(a);

// Call Member Functions

obj.show();

getch();

return 0;

}

## Sample Output

Enter the Number:7

7 is Prime Number.

**5). Simple Example Program For Copy Constructor In C++**

/\*  Example Program For Simple Example Program Of Copy Constructor Overloading In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

class Example        {

    // Variable Declaration

    int a,b;

    public:

    //Constructor with Argument

    Example(int x,int y)            {

    // Assign Values In Constructor

    a=x;

    b=y;

    cout<<"\nIm Constructor";

    }

    void Display()    {

    cout<<"\nValues :"<<a<<"\t"<<b;

    }

};

int main()                {

        Example Object(10,20);

        //Copy Constructor

        Example Object2=Object;

        // Constructor invoked.

        Object.Display();

        Object2.Display();

        // Wait For Output Screen

        getch();

        return 0;

}

## Sample Output

Im Constructor

Values :10      20

Values :10      20

**F). C++ Function Example Programs**

**1). Simple Example Program for Function In C++**

/\*  Example Program For Simple Example Program Of Function In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

// Simple Function

void printmessage ()

{

  cout << "Im Function In C++";

}

int main()                {

        printmessage ();

        getch();

        return 0;

}

## Sample Output

Im Function In C++

**2). Simple Example Program for Function Find Smallest Number In C++**

/\*  Example Program For Simple Example Program Of Function Find Smallest Number In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

// Simple Function

int compare( int a, int b )

{

    return (a+4 < b)? a : b;

}

int main()                {

    cout<<"\nSmallest Number :"<<compare(1,10);

    cout<<"\nSmallest Number :"<<compare(31,10);

    cout<<"\nSmallest Number :"<<compare(11,8);

    getch();

    return 0;

}

## Sample Output

Smallest Number :1

Smallest Number :10

Smallest Number :8

**3). Simple Example Program for Function to Find Factorial In C++**

/\*  Example Program For Simple Example Program Of Function Find Factorial Number In C++

    little drops @ thiyagaraaj.com

    Coded By:THIYAGARAAJ MP             \*/

#include<iostream>

#include<conio.h>

using namespace std;

// Simple factorial Function

int factorial(int var)

{

    int fact=1;

    for(int i=1;i<=var;i++)

    fact = fact \* i;

    return fact;

}

int main()                {

    cout<<"5 Factorial Number :"<<factorial(5);

    getch();

    return 0;

}

## Sample Output

5 Factorial Number :120

**4). Simple Example Program for Inline Function Using C++ Programming**

## Program Aim

To write a program to find the multiplication values and the cubic values using inline function.

## ALGORITHM:

Step 1: Start the pogram.

Step 2: Declare the class.

Step 3: Declare and define the inline function for multiplication and cube.

Step 4: Declare the class object and variables.

Step 5: Read two values.

Step 6: Call the multiplication and cubic functions using class objects.

Step 7: Return the values.

Step 8: Display.

Step 9: Stop the program.

## Example Program Of Inline Function

#include<iostream.h>

#include<conio.h>

class line

{

   public:

              inline float mul(float x,float y)

              {

                            return(x\*y);

              }

              inline float cube(float x)

              {

                            return(x\*x\*x);

              }

};

void main()

{

              line obj;

              float val1,val2;

              clrscr();

              cout<<"Enter two values:";

              cin>>val1>>val2;

              cout<<"\nMultiplication value is:"<<obj.mul(val1,val2);

              cout<<"\n\nCube value is          :"<<obj.cube(val1)<<"\t"<<obj.cube(val2);

              getch();

}

## Sample Output:

              Enter two values: 5  7

              Multiplication Value is: 35

              Cube Value is: 25 and 343

**G). C++ Operator Overloading Example Programs**

**Simple Addition in C++ Binary Operator Overloading Using C++ Programming**

## Example Program

// Header Files

#include<iostream>

#include<conio.h>

//Standard namespace declaration

using namespace std;

class overloading

{

 int value;

 public:

 void setValue(int temp)

 {

      value = temp;

 }

 overloading operator+(overloading ob)

 {

  overloading t;

  t.value=value+ob.value;

  return(t);

  }

void display()

{

 cout<<value<<endl;

}

};

//Main Functions

int main()

{

   overloading obj1,obj2,result;

   int a,b;

   cout<<"Enter the value of Complex Numbers a,b:";

   cin>>a>>b;

   obj1.setValue(a);

   obj2.setValue(b);

   result = obj1+obj2;

   cout<<"Input Values:\n";

   obj1.display();

   obj2.display();

   cout<<"Result:";

   result.display();

   getch();

   return 0;

}

## Sample Output:

Enter the value of Complex Numbers a,b:10

5

Input Values:

10

5

Result:15

**Simple Program Book Entry Using structure Variable in C++ Programming**

## Example Program

#include<iostream.h>

#include<stdio.h>

struct books

{

    char name[20],author[20];

}a[50];

int main()

{

    int i,n;

    cout<<"No Of Books[less than 50]:";

    cin>>n;

cout<<"Enter the book details\n";

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

    for(i=0;i<n;i++)

    {

 cout<<"Details of Book No "<<i+1<<"\n";

 cout<<"Book Name :";

 cin>>a[i].name;

 cout<<"Book Author :";

 cin>>a[i].author;

 cout<<"----------------------\n";

    }

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

    cout<<" S.No\t| Book Name\t|author\n";

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

    for(i=0;i<n;i++)

    {

 cout<<"\n  "<<i+1<<"\t|"<<a[i].name<<"\t| "<<a[i].author;

    }

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

    return 0;

}

## Sample Output

No Of Books[less than 50]:2

Enter the book details

----------------------

Details of Book No 1

Book Name :Programming

Book Author :Dromy

Details of Book No 2

Book Name :C

Book Author :Byron

=======================================================

 S.No | Book Name |author

=======================================================

  1 |Programming | Dromy

  2 |C | Byron

=======================================================

**Simple Program for Binary Operator Overloading Using C++ Programming**

To write a program to add two complex numbers using binary operator overloading.

## ALGORITHM:

Step 1: Start the program.

Step 2: Declare the class.

Step 3: Declare the variables and its member function.

Step 4: Using the function getvalue() to get the two numbers.

Step 5: Define the function operator +() to add two complex numbers.

Step 6: Define the function operator –()to subtract two complex numbers.

Step 7: Define the display function.

Step 8: Declare the class objects obj1,obj2 and result.

Step 9: Call the function getvalue using obj1 and obj2

Step 10: Calculate the value for the object result by calling the function operator + and     operator -.

Step 11: Call the display function using obj1 and obj2 and result.

Step 12: Return the values.

Step 13: Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

class complex

{

              int a,b;

    public:

              void getvalue()

              {

                 cout<<"Enter the value of Complex Numbers a,b:";

                 cin>>a>>b;

              }

              complex operator+(complex ob)

              {

                            complex t;

                            t.a=a+ob.a;

                            t.b=b+ob.b;

                            return(t);

              }

              complex operator-(complex ob)

              {

                            complex t;

                            t.a=a-ob.a;

                            t.b=b-ob.b;

                            return(t);

              }

              void display()

              {

                            cout<<a<<"+"<<b<<"i"<<"\n";

              }

};

void main()

{

   clrscr();

   complex obj1,obj2,result,result1;

   obj1.getvalue();

   obj2.getvalue();

   result = obj1+obj2;

   result1=obj1-obj2;

   cout<<"Input Values:\n";

   obj1.display();

   obj2.display();

   cout<<"Result:";

   result.display();

  result1.display();

   getch();

}

## Output:

Enter the value of Complex Numbers a, b

4                  5

Enter the value of Complex Numbers a, b

2                  2

Input Values

4 + 5i

2 + 2i

Result

6 +   7i

2 +   3i

**Simple Program for Constructor Using C++ Programming**

This Example Programs Calculate Prime Number Using Constructor

|  |
| --- |
| ALGORITHM: STEP 1:  Start the program.  STEP 2:  Declare the class as Prime with data members,                    Member functions.  STEP 3:  Consider the argument constructor Prime() with integer                    Argument.  STEP 4:  To cal the function calculate() and do the following steps.  STEP 5:  For i=2 to a/2 do  STEP 6:  Check if a%i==0 then set k=0 and break.  STEP 7:  Else set k value as 1.  STEP 8:  Increment the value i as 1.  STEP 9:  Check whether the k value is 1 or 0.  STEP 10:If it is 1 then display the value is a prime number.  STEP 11:Else display the value is not prime.  STEP 12:Stop the program. |

#include<iostream.h>

#include<conio.h>

class prime

{

                int a,k,i;

              public:

              prime(int x)

              {

                            a=x;

              }

              void calculate()

              {

                 k=1;

                {

                     for(i=2;i<=a/2;i++)

       if(a%i==0)

                     {

                              k=0;

                              break;

                     }

                     else

                    {

                            k=1;

                  }

                }

              }

void show()

              {

                if(k==1)

                  cout<< “\n\tA is prime Number. ";

                else

                  cout<<"\n\tA is Not prime.";

              }

};

void main()

{

    clrscr();

    int a;

    cout<<"\n\tEnter the Number:";

    cin>>a;

    prime obj(a);

    obj.calculate();

    obj.show();

    getch();

}

## Sample Output:

Enter the number: 7

Given number is Prime Number

**Simple Program for Copy Constructor Using C++ Programming**

To calculate factorial of a given number using copy constructor.

## ALGORITHM:

STEP 1:  Start the program.

STEP 2:  Declare the class name as Copy with data members and member functions.

STEP 3:  The constructor Copy() with argument to assign the value.

STEP 4:  To cal the function calculate() do the following steps.

STEP 5:   For i=1 to var do

STEP 6:   Calculate fact\*i to assign to fact.

STEP 7:   Increment the value as 1.

STEP 8:   Return the value fact.

STEP 9:   Print the result.

STEP 10: Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

class copy

{

              int var,fact;

              public:

                copy(int temp)

                {

                 var = temp;

                }

                double calculate()

                {

                            fact=1;

                            for(int i=1;i<=var;i++)

                            {

                            fact = fact \* i;

                            }

                            return fact;

                }

};

void main()

{

    clrscr();

    int n;

    cout<<"\n\tEnter the Number : ";

    cin>>n;

    copy obj(n);

    copy cpy=obj;

    cout<<"\n\t"<<n<<" Factorial is:"<<obj.calculate();

    cout<<"\n\t"<<n<<" Factorial is:"<<cpy.calculate();

    getch();

}

## Output:

Enter the Number: 5

Factorial is: 120

Factorial is: 120

**Simple Program for Exception Handling Divide by zero Using C++ Programming**

## ALGORITHM:

Step 1: Start the program.

Step 2: Declare the variables a,b,c.

Step 3: Read the values a,b,c,.

Step 4: Inside the try block check the condition.

            a. if(a-b!=0) then calculate the value of d and display.

            b. otherwise throw the exception.

Step 5: Catch the exception and display the appropriate message.

Step 6: Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

void main()

{

   int a,b,c;

   float  d;

   clrscr();

   cout<<"Enter the value of a:";

   cin>>a;

   cout<<"Enter the value of b:";

   cin>>b;

   cout<<"Enter the value of c:";

   cin>>c;

   try

   {

              if((a-b)!=0)

              {

                 d=c/(a-b);

                 cout<<"Result is:"<<d;

              }

              else

              {

                 throw(a-b);

              }

   }

   catch(int i)

   {

              cout<<"Answer is infinite because a-b is:"<<i;

   }

   getch();

}

## Output:

              Enter the value for a: 20

              Enter the value for b: 20

              Enter the value for c: 40

              Answer is infinite because a-b is: 0

**Simple Program for Exception Handling with Multiple Catch Using C++ Programming**

To perform exception handling with multiple catch.

## ALGORITHM:

Step 1: Start the program.

Step 2: Declare and define the function test().

Step 3: Within the try block check whether the value is greater than zero or not.

a.    if  the value greater than zero throw the value and catch the corresponding exception.

b.    Otherwise throw the character and catch the corresponding exception.

Step 4: Read the integer and character values for the function test().

Step 5: Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

void test(int x)

{

   try

   {

              if(x>0)

                 throw x;

        else

                 throw 'x';

   }

   catch(int x)

   {

              cout<<"Catch a integer and that integer is:"<<x;

   }

   catch(char x)

   {

              cout<<"Catch a character and that character is:"<<x;

   }

}

void main()

{

   clrscr();

   cout<<"Testing multiple catches\n:";

   test(10);

   test(0);

   getch();

}

## Output:

Testing multiple catches

Catch a integer and that integer is: 10

Catch a character and that character is: x

**Simple Program for Friend Function Using C++ Programming**

To find the mean value of a given number using friend function.

## ALGORITHM:

STEP 1:  Start the program.

STEP 2:  Declare the class name as Base with data members and member functions.

STEP 3:  The function get() is used to read the 2 inputs from the user.

STEP 4:  Declare the friend function mean(base ob) inside the class.

STEP 5:  Outside the class to define the friend function and do the following.

STEP 6:  Return the mean value (ob.val1+ob.val2)/2 as a float.

STEP 7:  Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

class  base

{

    int val1,val2;

   public:

    void get()

    {

       cout<<"Enter two values:";

       cin>>val1>>val2;

    }

    friend float mean(base ob);

};

float mean(base ob)

{

   return float(ob.val1+ob.val2)/2;

}

void main()

{

    clrscr();

    base obj;

    obj.get();

    cout<<"\n Mean value is : "<<mean(obj);

    getch();

}

## Output:

Enter two values: 10, 20

Mean Value is: 15

**Simple Program for Function Overloading Using C++ Programming**

To calculate the area of  circle, rectangle and  triangle using function overloading.

## ALGORITHM:

STEP 1:  Start the program.

STEP 2:  Declare the class name as fn with data members and member functions.

STEP 3:  Read the choice from the user.

STEP 4:  Choice=1 then go to the step 5.

STEP 5:  The function area() to find area of circle with one integer argument.

STEP 6:  Choice=2 then go to the step 7.

STEP 7:  The function area() to find area of rectangle with two integer argument.

STEP 8:  Choice=3 then go to the step 9.

STEP 9:  The function area() to find area of triangle with three arguments, two as Integer and one as float.

STEP 10: Choice=4 then stop the program.

## PROGRAM:

#include<iostream.h>

#include<stdlib.h>

#include<conio.h>

#define pi 3.14

class fn

{

      public:

        void area(int);  //circle

        void area(int,int);  //rectangle

        void area(float ,int,int);  //triangle

};

void fn::area(int a)

{

cout<<"Area of Circle:"<<pi\*a\*a;

}

void fn::area(int a,int b)

{

      cout<<"Area of rectangle:"<<a\*b;

}

void fn::area(float t,int a,int b)

{

      cout<<"Area of triangle:"<<t\*a\*b;

}

void main()

{

     int ch;

     int a,b,r;

     clrscr();

     fn obj;

     cout<<"\n\t\tFunction Overloading";

     cout<<"\n1.Area of Circle\n2.Area of Rectangle\n3.Area of Triangle\n4.Exit\n:”;

     cout<<”Enter your Choice:";

     cin>>ch;

     switch(ch)

     {

              case 1:

                cout<<"Enter Radious of the Circle:";

                cin>>r;

                obj.area(r);

                break;

              case 2:

                cout<<"Enter Sides of the Rectangle:";

                cin>>a>>b;

                obj.area(a,b);

                break;

              case 3:

                cout<<"Enter Sides of the Triangle:";

                cin>>a>>b;

                obj.area(0.5,a,b);

                break;

              case 4:

                exit(0);

     }

getch();

}

## Output:

              Function Overloading

              1. Area of Circle

              2. Area of Rectangle

              3. Area of Triangle

              4. Exit

              Enter Your Choice: 2

              Enter the Sides of the Rectangle: 5 5

              Area of Rectangle is: 25

              1. Area of Circle

              2. Area of Rectangle

              3. Area of Triangle

              4. Exit

              Enter Your Choice: 4

**Simple Program for Function Template Using C++ Programming**

To swap the numbers using the concept of function template.

## ALGORITHM:

STEP 1:  Start the program.

STEP 2:  Declare the template class.

STEP 3:  Declare and define the functions to swap the values.

STEP 4:  Declare and define the functions to get the values.

STEP 5:  Read the values and call the corresponding functions.

STEP6:   Display the results.

STEP 7:  Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

template<class t>

void swap(t &x,t &y)

{

   t temp=x;

   x=y;

   y=temp;

}

void fun(int a,int b,float c,float d)

{

   cout<<"\na and b before swaping :"<<a<<"\t"<<b;

   swap(a,b);

   cout<<"\na and b after swaping  :"<<a<<"\t"<<b;

   cout<<"\n\nc and d before swaping :"<<c<<"\t"<<d;

   swap(c,d);

   cout<<"\nc and d after swaping  :"<<c<<"\t"<<d;

}

void main()

{

    int a,b;

    float c,d;

    clrscr();

    cout<<"Enter A,B values(integer):";

    cin>>a>>b;

    cout<<"Enter C,D values(float):";

    cin>>c>>d;

    fun(a,b,c,d);

    getch();

}

## Output:

Enter A, B values (integer): 10  20

Enter C, D values (float):    2.50  10.80

A and B before swapping: 10 20

A and B after    swapping:  20 10

C and D before swapping: 2.50  10.80

C and D after               swapping: 10.80  2.50

**Simple Program for Inline Function Using C++ Programming**

To write a program to find the multiplication values and the cubic values using inline function.

## ALGORITHM:

Step 1: Start the pogram.

Step 2: Declare the class.

Step 3: Declare and define the inline function for multiplication and cube.

Step 4: Declare the class object and variables.

Step 5: Read two values.

Step 6: Call the multiplication and cubic functions using class objects.

Step 7: Return the values.

Step 8: Display.

Step 9: Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

class line

{

   public:

              inline float mul(float x,float y)

              {

                            return(x\*y);

              }

              inline float cube(float x)

              {

                            return(x\*x\*x);

              }

};

void main()

{

              line obj;

              float val1,val2;

              clrscr();

              cout<<"Enter two values:";

              cin>>val1>>val2;

              cout<<"\nMultiplication value is:"<<obj.mul(val1,val2);

              cout<<"\n\nCube value is          :"<<obj.cube(val1)<<"\t"<<obj.cube(val2);

              getch();

}

## Output:

              Enter two values: 5  7

              Multiplication Value is: 35

              Cube Value is: 25 and 343

**Simple Program for Multiple Inheritance Using C++ Programming**

To find out the student details using multiple inheritance.

## ALGORITHM:

Step 1: Start the program.

Step 2: Declare the base class student.

Step 3: Declare and define the function get() to get the student details.

Step 4: Declare the other class sports.

Step 5: Declare and define the function getsm() to read the sports mark.

Step 6: Create the class statement derived from student and sports.

Step 7: Declare and define the function display() to find out the total and average.

Step 8: Declare the derived class object,call the functions get(),getsm() and display().

Step 9: Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

class student

{

    protected:

       int rno,m1,m2;

    public:

                void get()

              {

                            cout<<"Enter the Roll no :";

                            cin>>rno;

                            cout<<"Enter the two marks   :";

                            cin>>m1>>m2;

              }

};

class sports

{

    protected:

       int sm;                   // sm = Sports mark

    public:

                void getsm()

              {

                 cout<<"\nEnter the sports mark :";

                 cin>>sm;

              }

};

class statement:public student,public sports

{

    int tot,avg;

    public:

    void display()

              {

                 tot=(m1+m2+sm);

                 avg=tot/3;

                 cout<<"\n\n\tRoll No    : "<<rno<<"\n\tTotal      : "<<tot;

               cout<<"\n\tAverage    : "<<avg;

              }

};

void main()

{

   clrscr();

   statement obj;

   obj.get();

   obj.getsm();

   obj.display();

   getch();

}

## Output:

              Enter the Roll no: 100

              Enter two marks

              90

              80

              Enter the Sports Mark: 90

              Roll No: 100

              Total    : 260

              Average: 86.66

**Simple Program for Read & Write File Operation (Convert lowercase to uppercase) Using C++ Programming**

program to convert lowercase to uppercase .

## ALGORITHM:

STEP 1:  Start the program.

STEP 2:  Declare the variables.

STEP 3:  Read  the file name.

STEP 4:  open the file to write the contents.

STEP 5:  writing the file contents up to reach a particular condition.

STEP6:   write the file contents as uppercase.

STEP7:   open the file to read the contents.

STEP 8:  Stop the program.

## PROGRAM:

#include<fstream.h>

#include<stdio.h>

#include<ctype.h>

#include<string.h>

#include<iostream.h>

#include<conio.h>

void main()

{

              char c,u;

              char fname[10];

              clrscr();

              ofstream out;

              cout<<"Enter File Name:";

              cin>>fname;

              out.open(fname);

              cout<<"Enter the text(Enter # at end)\n";   //write contents to file

              while((c=getchar())!='#')

              {

                            u=c-32;

                            out<<u;

              }

              out.close();

              ifstream in(fname);        //read the contents of file

              cout<<"\n\n\t\tThe File contains\n\n";

              while(in.eof()==0)

              {

                            in.get(c);

                            cout<<c;

              }

              getch();

}

## Output:

Enter File Name: two.txt

Enter contents to store in file (enter # at end)

oops programming

The File Contains

OOPS PROGRAMMING

**Simple Program for Read File Operation Using C++ Programming**

Program for read the content of  a file.

## ALGORITHM:

STEP 1:  Start the program.

STEP 2:  Declare the variables.

STEP 3:  Get the file name to read.

STEP 4:  Using ifstreamin(filename) check whether the file exist.

STEP 5:  If the file exist then check for the end of file condition.

STEP 6:  Read the contents of the file.

STEP 7:  Print the contents of the file.

STEP 8:  Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

#include<fstream.h>

void main()

{

              char c,fname[10];

              clrscr();

              cout<<"Enter file name:";

              cin>>fname;

              ifstream in(fname);

              if(!in)

              {

                            cout<<"File Does not Exist";

                            getch();

                            return;

              }

              cout<<"\n\n";

              while(in.eof()==0)

              {

                            in.get(c);

                            cout<<c;

              }

              getch();

}

## Output:

              Enter File name: one.txt

              Master of Computer Applications

**Simple Program for Single Inheritance Using C++ Programming**

To write a program to find out the payroll system using single inheritance.

## ALGORITHM:

Step 1: Start the program.

Step 2: Declare the base class emp.

Step 3: Define and declare the function get() to get the employee details.

Step 4: Declare the derived class salary.

Step 5: Declare and define the function get1() to get the salary details.

Step 6: Define the function calculate() to find the net pay.

Step 7: Define the function display().

Step 8: Create the derived class object.

Step 9: Read the number of employees.

Step 10: Call the function get(),get1() and calculate() to each employees.

Step 11: Call the display().

Step 12: Stop the program.

## PROGRAM:PAYROLL SYSTEM USING SINGLE INHERITANCE

#include<iostream.h>

#include<conio.h>

class emp

{

   public:

     int eno;

     char name[20],des[20];

     void get()

     {

              cout<<"Enter the employee number:";

              cin>>eno;

              cout<<"Enter the employee name:";

              cin>>name;

              cout<<"Enter the designation:";

              cin>>des;

     }

};

class salary:public emp

{

     float bp,hra,da,pf,np;

   public:

     void get1()

     {

              cout<<"Enter the basic pay:";

              cin>>bp;

              cout<<"Enter the Humen Resource Allowance:";

              cin>>hra;

              cout<<"Enter the Dearness Allowance :";

              cin>>da;

              cout<<"Enter the Profitablity Fund:";

              cin>>pf;

     }

     void calculate()

     {

              np=bp+hra+da-pf;

     }

     void display()

     {

              cout<<eno<<"\t"<<name<<"\t"<<des<<"\t"<<bp<<"\t"<<hra<<"\t"<<da<<"\t"<<pf<<"\t"<<np<<"\n";

     }

};

void main()

{

    int i,n;

    char ch;

    salary s[10];

    clrscr();

    cout<<"Enter the number of employee:";

    cin>>n;

    for(i=0;i<n;i++)

    {

              s[i].get();

              s[i].get1();

              s[i].calculate();

    }

    cout<<"\ne\_no \t e\_name\t des \t bp \t hra \t da \t pf \t np \n";

    for(i=0;i<n;i++)

    {

              s[i].display();

    }

    getch();

}

## Output:

Enter the Number of employee:1

Enter the employee No: 150

Enter the employee Name: ram

Enter the designation: Manager

Enter the basic pay: 5000

Enter the HR allowance: 1000

Enter the Dearness allowance: 500

Enter the profitability Fund: 300

E.No   E.name   des      BP    HRA   DA   PF     NP

150    ram      Manager  5000  1000  500  300    6200

**Simple Program for Static Data and Member Function Using C++ Programming**

To count the object value using the storage keyword static.

## ALGORITHM:

STEP 1:  Start the program.

STEP 2:  Declare the class name as Stat with data member s and member functions.

STEP 3:  The constructor Stat() which is used to increment the value of count as 1 to to assign the variable code.

STEP 4:  The function showcode() to display the code value.

STEP 5:  The function showcount() to display the count value.

STEP 6:   Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

class stat

{

    int code;

    static int count;

   public:

    stat()

    {

      code=++count;

    }

    void showcode()

    {

      cout<<"\n\tObject number is :"<<code;

    }

    static void showcount()

    {

              cout<<"\n\tCount Objects :"<<count;

    }

};

int stat::count;

void main()

{

   clrscr();

   stat obj1,obj2;

   obj1.showcount();

   obj1.showcode();

   obj2.showcount();

   obj2.showcode();

   getch();

}

## Output:

Count Objects: 2

Object Number is: 1

Count Objects: 2

Object Number is: 2

**Simple Program for Unary Operator Overloading Using C++ Programming**

To write a program to find the complex numbers using unary operator overloading.

## ALGORITHM:

Step 1: Start the program.

Step 2: Declare the class.

Step 3: Declare the variables and its member function.

Step 4: Using the function getvalue() to get the two numbers.

Step 5: Define the function operator ++ to increment the values

Step 6: Define the function operator - -to decrement the values.

Step 7: Define the display function.

Step 8: Declare the class object.

Step 9: Call the function getvalue

Step 10: Call the function operator ++() by incrementing the class object and call the   function display.

Step 11: Call the function operator - -() by decrementing the class object and call the function display.

Step 12: Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

class complex

{

     int a,b,c;

    public:

        complex(){}

        void getvalue()

       {

                 cout<<"Enter the Two Numbers:";

                 cin>>a>>b;

       }

  void operator++()

      {

                 a=++a;

                 b=++b;

       }

       void operator--()

       {

                 a=--a;

                 b=--b;

        }

        void display()

        {

                 cout<<a<<"+\t"<<b<<"i"<<endl;

         }

};

void main()

{

     clrscr();

     complex obj;

     obj.getvalue();

     obj++;

     cout<<"Increment Complex Number\n";

     obj.display();

     obj--;

     cout<<"Decrement Complex Number\n";

     obj.display();

     getch();

}

## Output:

Enter the two numbers: 3 6

Increment Complex Number

4 +               7i

Decrement Complex Number

3 +               6i

**Simple Program for Virtual Base Class Using C++ Programming**

To calculate the total mark of a student using the concept of virtual base class.

## ALGORITHM:

Step 1: Start the program.

Step 2: Declare the base class student.

Step 3: Declare and define the functions getnumber() and putnumber().

Step 4: Create the derived class test virtually derived from the base class student.

Step 5: Declare and define the function getmarks() and putmarks().

Step 6: Create the derived class sports virtually derived from the base class student.

Step 7: Declare and define the function getscore() and putscore().

Step 8: Create the derived class result derived from the class test and sports.

Step 9: Declare and define the function display() to calculate the total.

Step 10: Create the derived class object obj.

Step 11: Call the function get number(),getmarks(),getscore() and display().

Step 12: Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

class student

{

   int rno;

  public:

   void getnumber()

   {

              cout<<"Enter Roll No:";

              cin>>rno;

   }

   void putnumber()

   {

              cout<<"\n\n\tRoll No:"<<rno<<"\n";

   }

};

class test:virtual public student

{

  public:

   int part1,part2;

   void getmarks()

   {

              cout<<"Enter Marks\n";

              cout<<"Part1:";

              cin>>part1;

              cout<<"Part2:";

              cin>>part2;

   }

   void putmarks()

   {

              cout<<"\tMarks Obtained\n";

              cout<<"\n\tPart1:"<<part1;

              cout<<"\n\tPart2:"<<part2;

   }

};

class sports:public virtual student

{

  public:

    int score;

    void getscore()

    {

              cout<<"Enter Sports Score:";

              cin>>score;

    }

    void putscore()

    {

              cout<<"\n\tSports Score is:"<<score;

    }

};

class result:public test,public sports

{

    int total;

  public:

   void display()

   {

      total=part1+part2+score;

      putnumber();

      putmarks();

      putscore();

      cout<<"\n\tTotal Score:"<<total;

   }

};

void main()

{

   result obj;

   clrscr();

   obj.getnumber();

   obj.getmarks();

   obj.getscore();

   obj.display();

   getch();

}

## Output:

              Enter Roll No: 200

              Enter Marks

              Part1: 90

              Part2: 80

              Enter Sports Score: 80

              Roll No: 200

              Marks Obtained

              Part1: 90

              Part2: 80

              Sports Score is: 80

              Total Score is: 250

**Simple Program for Virtual Functions Using C++ Programming**

Simple Example Program for virtual functions.

## ALGORITHM:

Step 1: Start the program.

Step 2: Declare the base class base.

Step 3: Declare and define the virtual function show().

Step 4: Declare and define the function display().

Step 5: Create the derived class from the base class.

Step 6: Declare and define the functions display() and show().

Step 7: Create the base class object and pointer variable.

Step 8: Call the functions display() and show() using the base class object and pointer.

Step 9: Create the derived class object and call the functions display() and show() using the derived class object and pointer.

Step 10: Stop the program.

## PROGRAM:

#include<iostream.h>

#include<conio.h>

class base

{

    public:

      virtual void show()

      {

                cout<<"\n  Base class show:";

      }

      void display()

      {

              cout<<"\n  Base class display:" ;

      }

};

class drive:public base

{

   public:

      void display()

      {

              cout<<"\n  Drive class display:";

      }

      void show()

      {

              cout<<"\n  Drive class show:";

      }

};

void main()

{

   clrscr();

   base obj1;

   base \*p;

   cout<<"\n\t P points to base:\n"  ;

   p=&obj1;

   p->display();

   p->show();

   cout<<"\n\n\t P points to drive:\n";

   drive obj2;

   p=&obj2;

   p->display();

   p->show();

   getch();

}

## Output:

              P points to Base

              Base class display

              Base class show

              P points to Drive

              Base class Display

              Drive class Show

**Simple Program for Write File Operation Using C++ Programming**

To perform the write operation with in a file.

## ALGORITHM:

STEP 1:  Start the program.

STEP 2:  Declare the variables.

STEP 3:  Read  the file name.

STEP 4:  open the file to write the contents.

STEP 5:  writing the file contents up to reach a particular condition.

STEP 6:  Stop the program.

## PROGRAM:

#include<iostream.h>

#include<stdio.h>

#include<conio.h>

#include<fstream.h>

void main()

{

              char c,fname[10];

              ofstream out;

              cout<<"Enter File name:";

              cin>>fname;

              out.open(fname);

              cout<<"Enter contents to store in file (Enter # at end):\n";

              while((c=getchar())!='#')

              {

                            out<<c;

              }

              out.close();

              getch();

}

## Output:

              Enter File name: one.txt

              Enter contents to store in file (enter # at end)

              Master of Computer Applications#