CALL BY VALUE

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

// function declaration
void swap(int x, int y);

int main () {
   // local variable declaration:
   int a = 100;
   int b = 200;

   cout << "Before swap, value of a :" << a << endl;
   cout << "Before swap, value of b :" << b << endl;

   // calling a function to swap the values.
   swap(a, b);

   cout << "After swap, value of a :" << a << endl;
   cout << "After swap, value of b :" << b << endl;

   return 0;
}
// function definition to swap the values.
void swap(int x, int y) {
   int temp;

   temp = x; /* save the value of x */
   x = y;    /* put y into x */
   y = temp; /* put x into y */

   return;
}
```

CALL BY REFRENCE

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

// function declaration
void swap(int &x, int &y);

int main () {
   // local variable declaration:
   int a = 100;
   int b = 200;

   cout << "Before swap, value of a :" << a << endl;
   cout << "Before swap, value of b :" << b << endl;

   /* calling a function to swap the values using variable reference.*/
   swap(a, b);
   return 0;
}
// function definition to swap the values.
   void swap(int &x, int &y) {
   int temp;
   temp = x; /* save the value at address x */
   x = y;    /* put y into x */
   y = temp; /* put x into y */
   cout << "After swap, value of a :" << x << endl;
   cout << "After swap, value of b :" << y << endl;
   return;
}
```

CALL BY POINTER

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

// function declaration
void swap(int *x, int *y);

int main () {
   // local variable declaration:
   int a = 100;
   int b = 200;

   cout << "Before swap, value of a :" << a << endl;
   cout << "Before swap, value of b :" << b << endl;

   /* calling a function to swap the values.
      * &a indicates pointer to a ie. address of variable a and
      * &b indicates pointer to b ie. address of variable b.
   */
   swap(&a, &b);
   cout << "After swap, value of a :" << a << endl;
   cout << "After swap, value of b :" << b << endl;

   return 0;
}
// function definition to swap the values.
   void swap(int *x, int *y) {
   int temp;
   temp = *x; /* save the value at address x */
   *x = *y; /* put y into x */
   *y = temp; /* put x into y */
   Return ;
}
```

DCLARATION CLASS FUNCTION INSIDE THE CLASS

```
#include <iostream>
using namespace std;
class date
{
private:
                int day;
                int month;
                int year;
public:
                void set(int dayin,int monthin,int yearin)
                {
                        day=dayin;
                        month=monthin;
                        year=yearin;
                }

                void show()
                {
                cout<<day<<"/"<<month<<"/"<<year<<endl;
                }
};

int main() {
        cout << "program for member functions inside the class\n"<<endl;
        date d1,d2,d3;
        d1.set(5,7,2017);
        d2.set(6,7,2017);
        d3.set(7,7,2017);
        cout<<"today day date is"<<endl;
        d1.show();
        cout<<"tomorrow day date is"<<endl;
```

```
    d2.show();
    cout<<"twodays from today day date is"<<endl;
    d3.show();
    return 0;
}
```

DECLARATION OF THR CLASS OUTSIDE THE CLASS

```
#include <iostream>
using namespace std;
class date
{
private:
        int day;
        int month;
        int year;
public:
        void set(int dayin,int monthin,int yearin);
        void show();
};
        void date::set(int dayin,int monthin,int yearin)
        {
            day=dayin;
            month=monthin;
            year=yearin;
        }
        void date::show()
        {
        cout<<day<<"/"<<month<<"/"<<year<<endl;
        }
```

```
int main() {
        cout << "program for member functions inside the class\n"<<endl;
        date d1,d2,d3;
        d1.set(5,7,2017);
        d2.set(6,7,2017);
        d3.set(7,7,2017);
        cout<<"today day date is"<<endl;
        d1.show();
        cout<<"tomorrow day date is"<<endl;
        d2.show();
        cout<<"twodays from today day date is"<<endl;
        d3.show();
        return 0;
}
```

INLINE FUNCTION

```
#include <iostream>
using namespace std;
class date
{
private:
                int day;
                int month;
                int year;
public:
                void set(int dayin,int monthin,int yearin);
                void show();
};
                inline void date::set(int dayin,int monthin,int yearin)
                {
                        day=dayin;
                        month=monthin;
```

```
            year=yearin;
        }
        inline void date::show()
        {
        cout<<day<<"/"<<month<<"/"<<year<<endl;
        }

int main() {
    cout << "program for member functions inside the class\n"<<endl;
    date d1,d2,d3;
    d1.set(5,7,2017);
    d2.set(6,7,2017);
    d3.set(7,7,2017);
    cout<<"today day date is"<<endl;
    d1.show();
    cout<<"tomorrow day date is"<<endl;
    d2.show();
    cout<<"twodays from today day date is"<<endl;
    d3.show();
    return 0;
}
```

NESTING FUNCTION(CLLING SAME FUNCTION OF THE CLASS)

```
// working.cpp by Bill Weinman <http://bw.org/>
#include <iostream>
using namespace std;
class NumberPairs
{
private:
    int num1;
    int num2;
public:
    void read()
    {
            cout<<"enter first number"<<endl;
            cin>>num1;
            cout<<"enter second number"<<endl;
            cin>>num2;
    }
    int max() //member function
    {
            if(num1>num2)
                    return num1;
            else
                    return num2;
    }
    void ShowMax()
    {
            cout<<"maximum is"<<max();
    }
};

int main() {
    NumberPairs n1;
    n1.read();
    n1.ShowMax();
    return 0;
}
```

DATA MEMBERS ACCESSING

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

class Box {
  public:
    double length;   // Length of a box
    double breadth;  // Breadth of a box
    double height;   // Height of a box
};

int main( ) {
  class Box Box1;                // Declare Box1 of type Box
  class Box Box2;               // Declare Box2 of type Box
  double volume = 0.0;    // Store the volume of a box here

  // box 1 specification
  Box1.height = 5.0;
  Box1.length = 6.0;
  Box1.breadth = 7.0;

  // box 2 specification
  Box2.height = 10.0;
  Box2.length = 12.0;
  Box2.breadth = 13.0;

  // volume of box 1
  volume = Box1.height * Box1.length * Box1.breadth;
  cout << "Volume of Box1 : " << volume <<endl;

  // volume of box 2
  volume = Box2.height * Box2.length * Box2.breadth;
  cout << "Volume of Box2 : " << volume <<endl;
```

```
   return 0;
}
```

SWAPING OF TWO NUMBERS(USING CLASS)

```
#include<iostream>

using namespace std;

class swap
{

public:
        int a,b;
        void getdata();
        void swapv();
        void display();
};

void swap::getdata()
{
cout<<"Enter two numbers:";
cin>>a>>b;
}

void swap::swapv()
{
a=a+b;
b=a-b;
a=a-b;
}
```

```
void swap::display()
{
cout<<"a="<<a<<"b="<<b;
}
int main()
{
class swap s1;
s1.getdata();
cout<<"\nBefore swap: \n";
s1.display();

s1.swapv();
cout<<"\nnAfter swap:\n";
s1.display();

return 0;
}
```