

Examples of Functions

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

void main()
{
    int i=4;    int j=10;
    i++;
    if (j > 0)
    cout<<"i is "<<i<<endl;

    if (j > 0)
    { int i=100;          /* 'i' is defined and so local to * this block */
    cout << "i is "<< i<<endl;
    } //end if

    cout<<"i is "<<i<<endl;
} //end main
```

OUTPUT:

```
i is 5
i is 100
i is 5
```

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

int main ( )
{
    int i = 5, j = 0;

    for (j = 0; j < 10; j++) {
        int i = j;    // OK, this is new i
        int k = 5;
        doSomething(i);
    }
    int sum = k;    // compile error, no k in scope
    j = i;    // sets j to 5
    for (j = 0; j < 100; j++ ) {
        int i = j;    // yet another new i
    }
    int i = 0;    // compile error -redefined variable
    return 0;
}

void doSomething(int i){ cout<<++i;}// OK, this is new i
```

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

int i = 10;
int main ( )
{
    cout<<i<<endl;//10
    for (int j = 0; j < 10; j++ ) {
        int i = 20;
        cout<<i<<endl;//20
    }
    cout<<i<<endl;//10
    int i = 30;
    cout<<i<<endl;//30
    cout<<i<<endl;//30
    return 0;
}
```

OUTPUT:

```
10
20
20
20
20
20
20
20
20
20
20
20
20
10
30
30
```

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

void foo(int y)
{
    cout << "y = " << y << endl;
}

int main()
{
    foo(5); // first call
    int x = 6;
    foo(x); // second call
    foo(x+1); // third call
    return 0;
}
```

OUTPUT:

```
y = 5
y = 6
y = 7
```

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

void foo(int y)
{
    cout << "y = " << y << endl;
    y = 6;
    cout << "y = " << y << endl;
} // y is destroyed here

int main()
{
    int x = 5;
    cout << "x = " << x << endl;
    foo(x);
    cout << "x = " << x << endl;

    return 0;
}
```

OUTPUT:

```
x = 5
y = 5
y = 6
x = 5
```

```

#include <iostream>
using namespace std;

int absolute (int); // function prototype for absolute()

int main()
{
    int num, answer;
    cout << "Enter an integer (0 to stop): ";
    cin >> num;
    while (num!=0)
    {
        answer = absolute(num);
        cout << "The absolute value of " << num
            << " is: " << answer << endl;
        cin >> num;
    }
    return 0;
}
// Define a function to take absolute value of an integer
int absolute(int x)
{
    if (x >= 0)
        return x;
    else
        return -x;
}

```

OUTPUT:

```

Enter an integer (0 to stop): -5
The absolute value of -5 is: 5

```

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

void foo(int &y)
// y is now a reference
{
    cout << "y = " << y << endl;
    y = 6;
    cout << "y = " << y << endl;
} // y is destroyed here

int main()
{
    int x = 5;
    cout << "x = " << x << endl;
    foo(x);
    cout << "x = " << x << endl;
    return 0;
}
```

OUTPUT:

```
x = 5
y = 5
y = 6
x = 6
```

```

#include <iostream>
using namespace std;

void AddOne(int &y)
{
    y++;
}

int main()
{
    int x = 1;
    cout << "x = " << x << endl;
    AddOne(x);
    cout << "x = " << x << endl;
    return 0;
}

```

OUTPUT:

```

x = 1
x = 2

```

```

#include <iostream>
using namespace std;

void duplicate (int& a, int& b, int & c);
int main ()
{
    int x=1,y=3,z=7;
    duplicate (x,y,z);
    cout << "x= "<<x<<"", y= "<<y<< ",z= "<<z<<endl;
    return 0;
}

void duplicate (int& a, int& b, int & c)
{
    a*=2;
    b*=2;
    c*=2;
}

```

OUTPUT:

```

x= 2, y= 6,z= 14

```

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

void swap(float &x, float &y);

int main()
{
    float a, b;
    cout << "Enter 2 numbers: " << endl;
    cin >> a >> b;
    if(a>b)
        swap(a,b);
    cout << "Sorted numbers: "; cout << a << " " << b << endl;
    return 0;
}

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

OUTPUT:

```
Enter 2 numbers:
4 3
Sorted numbers: 3 4
```

```
#include <iostream>
#include <math.h> // for sin() and cos()
using namespace std;

void GetSinCos(double dX, double &dSin, double &dCos)
{
    dSin = sin(dX);
    dCos = cos(dX);
}

int main()
{
    double dSin = 0.0;
    double dCos = 0.0;
    GetSinCos(30.0, dSin, dCos);
    cout << "The sin is " << dSin << endl;
    cout << "The cos is " << dCos << endl;
    return 0;
}
```

OUTPUT:

```
The sin is -0.988032
The cos is 0.154251
```

```

#include <iostream>
using namespace std;

void FtoC(int itemp);
void FtoC(float ftemp);
void FtoC(double dtemp);

int main( )
{
    int inttemp, level;
    float floattemp;
    double doubletemp;
    cout << "CONVERTING FAHRENHEIT TO CELSIUS\n";
    cout << "Select required level of precision\n";
    cout << "Integer (1) - Float (2) - Double (3)\n";
    cin >> level;
    cout << "Enter Fahrenheit temperature: ";

    switch (level)
    {

case 1 :
    cin >> inttemp;
    FtoC(inttemp);
    break;
case 2 :
    cin >> floattemp;
    FtoC(floattemp);
    break;
case 3 :
    cin >> doubletemp;
    FtoC(doubletemp);
    break;
default :
    cout << "Invalid selection\n";
    }
    return 0;
}

void FtoC( int itemp)
{
    int temp = (itemp - 32) * 5 / 9;
    cout << "Integer precision: ";
    cout << itemp << "F is " << temp << "C \n";
}

void FtoC( float ftemp)
{
    float temp = (ftemp - 32) * 5.0 / 9.0;
    cout << "Float precision: ";
    cout << ftemp << "F is " << temp << "C \n";;
}

void FtoC( double dtemp)
{
    double temp = (dtemp - 32) * 5.0 / 9.0;
    cout << "Double precision : ";
    cout << dtemp << "F is " << temp << "C \n";;
}

```

OUTPUT:

```
CONVERTING FAHRENHEIT TO CELSIUS
Select required level of precision
Integer (1) - Float (2) - Double (3)
1
Enter Fahrenheit temperature: 55
Integer precision: 55F is 12C
```

```
CONVERTING FAHRENHEIT TO CELSIUS
Select required level of precision
Integer (1) - Float (2) - Double (3)
2
Enter Fahrenheit temperature: 90.7
Float precision: 90.7F is 32.6111C
```

```
CONVERTING FAHRENHEIT TO CELSIUS
Select required level of precision
Integer (1) - Float (2) - Double (3)
3
Enter Fahrenheit temperature: 68.78
Double precision : 68.78F is 20.4333C
```

```

#include <iostream>
using namespace std;

int cubeByValue(int n); // prototype
int cubeByReference(int &nPtr); // function prototype
int main( ) {
    // insert code here...

    int number = 5; // initialize number
    cout<<"nThe original value of number is"<< number<<endl;
    // pass number by value to cubeByValue

    cout<<"The new value of number is "<< cubeByValue(number)<<endl;
    cout<<"The last version value of number is "<<number<<endl;

    cout<<"The original value of number is "<<number<<endl;
    // pass address of number to cubeByReference

    cout<<"The new value of number is "<<cubeByReference(number)<<endl;
    cout<<"The last version value of number is "<< number<<endl;

    return 0;
}
// calculate and return cube of integer argument
int cubeByValue(int n)
{
    return n * n * n; // cube local variable n and return result
}
// calculate cube of *nPtr; actually modifies number in main
int cubeByReference(int &nPtr)
{
    return nPtr = nPtr * nPtr * nPtr; // cube *nPtr
}

```

OUTPUT:

```

The original value of number is5
The new value of number is 125
The last version value of number is 5
The original value of number is 5
The new value of number is 125
The last version value of number is 125

```