

## Argument Passing Example

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

void Work ( int &, int, int & );

int main ( )
{
    int One = 5, Two = 10, Three = 15;
    cout << setw(5) << One << setw(5) << Two << setw(5) << Three << endl;
    Work ( One , Two, Three );
    cout << setw(5) << One << setw(5) << Two << setw(5) << Three << endl;
    return 0;
}

void Work ( int & First, int Second, int & Third )
{
    int Fourth;
    Fourth = First + Second + Third;
    cout << setw(5) << First << setw(5) << Second
        << setw(5) << Third << setw(5) << Fourth << endl;
    First = First + 2;
    Second = Second + 4;
    Third = Third + 6;
    Fourth = Fourth + 8;
    cout << setw(5) << First << setw(5) << Second
        << setw(5) << Third << setw(5) << Fourth << endl;
}
```

OUTPUT

### 1) What are actual arguments?

Actual arguments are the values or identifiers that are passed into a called function through the argument list during a function call.

```
Work ( One, Two, Three ); // function call in main
```

The variables One , Two, and Three are actual arguments.

### 2) What are formal/dummy arguments?

Formal arguments are the arguments that are used to represent the values that are passed into a called function through the argument list.

```
void Work ( int & First, int Second, int & Third ) // Work function header
```

The arguments First, Second, and Third are formal arguments. First represents One. Second represents Two. Third represents Three.

### 3) What is Pass-by-Value?

A value is passed into a called function through the argument list of the called function. The value may be modified inside the called function. Upon exit of the called function the modified value is discarded and the original value is maintained. (A copy of the original value is used in the called function – causing a temporary change)

### 4) What is Pass-by-Reference?

A value is passed into a called function through the argument list of the called function. The value may be modified inside the called function. Upon exit of the called function the modified value is maintained and the original value is discarded. (The address of the original value is used in the called function – causing a permanent change)