

MA122 - Computer Programming and Applications

Indian Institute of Space Science and Technology

March 08, 2017

Lecture 19

MA122 -
Computer
Programming
and
Applications

Reference as
function
parameters

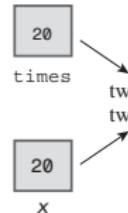
1 Reference as function parameters

Passing by value

```
void sneezy(int x);
int main()
{
    int times = 20;      → creates a variable
    sneezy(times);     called times, assigns
    ...                 it the value of 20
}

void sneezy(int x)
{
    ...                 → creates a variable
} ...                  called x, assigns it
                        the passed value of 20
```

The diagram illustrates the state of memory for the 'Passing by value' example. It shows two separate boxes, each containing the value '20'. The top box is labeled 'times' and the bottom box is labeled 'x'. An arrow from the text 'two variables, two names' points to both boxes.



Passing by reference

```
void grumpy(int &x);
int main()
{
    int times = 20;      → creates a variable
    grumpy(times);     called times, assigns
    ...                 it the value of 20
}

void grumpy(int &x)
{
    ...                 → makes x an
                        alias for times
}
```

The diagram illustrates the state of memory for the 'Passing by reference' example. It shows a single gray square containing the value '20'. This square is labeled 'times, x' above it. An arrow from the text 'one variable, two names' points to this single square.



example

```
1 #include <iostream>
2 void swapr(int & a, int & b); // a, b are aliases
3 void swapp(int * p, int * q); // p, q are addresses
4 void swapv(int a, int b); // a, b are new variables
5 int main()
6 {
7     using namespace std;
8     int wallet1 = 300;
9     int wallet2 = 350;
10
11    cout << "wallet1 = $" << wallet1;
12    cout << " wallet2 = $" << wallet2 << endl;
13
14    cout << "Using references to swap contents:\n";
15    swapr(wallet1, wallet2);
16    cout << "wallet1 = $" << wallet1;
17    cout << " wallet2 = $" << wallet2 << endl;
```

example

```
1 cout << "Using pointers to swap contents again:\n";
2 swapp(&wallet1, &wallet2);
3
4 cout << "wallet1 = $" << wallet1;
5 cout << " wallet2 = $" << wallet2 << endl;
6 cout << "Trying to use passing by value:\n";
7
8 swapv(wallet1, wallet2);
9 cout << "wallet1 = $" << wallet1;
10 cout << " wallet2 = $" << wallet2 << endl;
11 return 0;
12 }
13 void swapr(int & a, int & b) // use references
{
14     int temp;
15     temp = a;          // use a, b for values of variables
16     a = b;
17     b = temp; }
```

example

```
1 void swapp(int *p, int * q)
2 {
3     int temp;
4     temp = *p;
5     *p = *q;
6     *q = temp;
7 }
8 void swapv(int a, int b)
9 {
10    int temp;
11    temp = a;
12    a = b;
13    b = temp;
14 }
```

Regular and reference arguments

```
1 #include <iostream>
2 double cube(double a);
3 double refcube(double &ra);
4 int main ()
5 {
6     using namespace std;
7     double x = 3.0;
8
9     cout << cube(x);
10    cout << " = cube of " << x << endl;
11
12    cout << refcube(x);
13    cout << " = cube of " << x << endl;
14    return 0;
15 }
```

Regular and reference arguments

```
1 double cube(double a)
2 {
3     a *= a * a;
4     return a;
5 }
6 double refcube(double &ra)
7 {
8     ra *= ra * ra;
9     return ra;
10 }
```