

MA122 - Computer Programming and Applications

Indian Institute of Space Science and Technology

February 22, 2017

Lecture 15

MA122 -
Computer
Programming
and
Applications

Address

pointer

Pointer
Arithmetic

Functions and
Arrays

1 Address

2 pointer

3 Pointer Arithmetic

4 Functions and Arrays

How to find the address of a variable

```
1 #include <iostream>
2 int main()
3 {
4     using namespace std;
5
6
7     int donuts = 6;
8     double cups = 4.5;
9
10    cout << "donuts value = " << donuts;
11    cout << " and donuts address = " << &donuts << endl;
12
13
14    cout << "cups value = " << cups;
15    cout << " and cups address = " << &cups << endl;
16
17    return 0;
18 }
```

Lecture 15

MA122 -
Computer
Programming
and
Applications

Address

pointer

Pointer
Arithmetic

Functions and
Arrays

1 Address

2 **pointer**

3 Pointer Arithmetic

4 Functions and Arrays

dereferencing

```
1 #include <iostream>
2 int main()
3 {
4     using namespace std;
5
6     int updates = 6;        // declare a variable
7
8     int * p_updates;        // declare pointer to an int
9
10    p_updates = &updates;    // assign address of int to
                             // pointer
11    // express values two ways
12
13    cout << "Values: updates = " << updates;
14
15    cout << ", *p_updates = " << *p_updates << endl;
```

program

MA122 -
Computer
Programming
and
Applications

Address

pointer

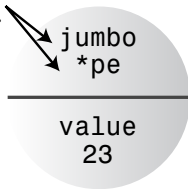
Pointer
Arithmetic

Functions and
Arrays

```
1 // express address two ways
2
3 cout << "Addresses: &updates = " << &updates;
4
5 cout << ", p_updates = " << p_updates << endl;
6
7 // use pointer to change value
8
9 *p_updates = *p_updates + 1;
10
11 cout << "Now updates = " << updates << endl;
12
13 return 0;
14 }
```

```
int jumbo = 23;  
int *pe = &jumbo;
```

These are
the same.



These are
the same.

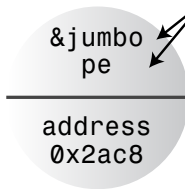
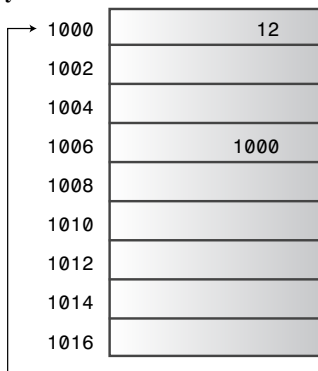


Figure 4.8 Two sides of a coin.

Memory address



Variable name

ducks

↑
birddog
points to
ducks

birddog

int ducks = 12;

creates ducks variable, stores
the value 12 in the variable

int *birddog = &ducks;

creates birddog variable, stores
the address of ducks in the variable

Figure 4.9 Pointers store addresses.

Initialize

```
1  #include <iostream>
2  int main()
3  {
4      using namespace std;
5
6      int higgins = 5;
7      int * pt = &higgins;
8
9      cout << "Value of higgins = " << higgins
10     << "; Address of higgins = " << &higgins << endl;
11
12     cout << "Value of *pt = " << *pt
13     << "; Value of pt = " << pt << endl;
14
15     return 0;
16 }
```

Lecture 15

MA122 -
Computer
Programming
and
Applications

Address

pointer

Pointer
Arithmetic

Functions and
Arrays

1 Address

2 pointer

3 Pointer Arithmetic

4 Functions and Arrays

example

```
1 #include <iostream>
2 int main()
3 {
4     using namespace std;
5     double wages[3] = {10000.0, 20000.0, 30000.0};
6     short stacks[3] = {3, 2, 1};
7
8     // Here are two ways to get the address of an array
9
10    double * pw = wages; // name of an array = address
11    short * ps = &stacks[0]; // or use address operator
12    // with array element
13
14    cout << "pw = " << pw << ", *pw = " << *pw << endl;
15
16    pw = pw + 1;
17    cout << "add 1 to the pw pointer:\n";
18    cout << "pw = " << pw << ", *pw = " << *pw << "\n\n"
```

example

```
1  cout << "ps = " << ps << ", *ps = " << *ps << endl;  
2  ps = ps + 1;  
3  cout << "add 1 to the ps pointer:\n";  
4  
5  cout << "ps = " << ps << ",*ps = " << *ps << "\n\n";  
6  
7  cout << "access two elements with array notation\n";  
8  cout << "stacks[0] = " << stacks[0]  
9  << ", stacks[1] = " << stacks[1] << endl;  
10  
11 cout << "access two elements with pointer notation\n  
    ";  
12  
13 cout << "*stacks = " << *stacks  
14 << ", *(stacks + 1) = " << *(stacks + 1) << endl;  
15  
16 cout << sizeof(wages) << " = size of wages array\n";  
17 cout << sizeof(pw) << " = size of pw pointer\n";
```

Pointer Addition

```
double wages[3] = {10000.0, 20000.0, 30000.0};  
short stacks[3] = {3, 2, 1};  
double * pw = wages;  
short * ps = &stacks[0];
```

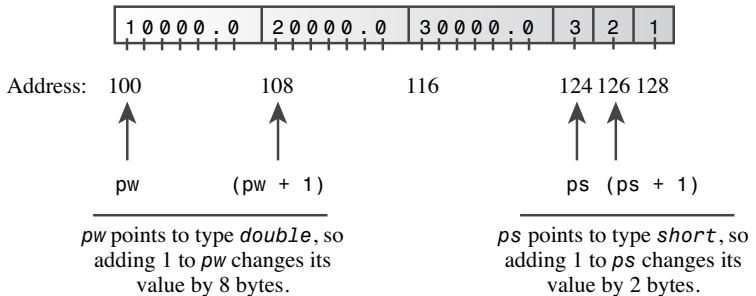


Figure 4.10 Pointer addition.

Lecture 15

MA122 -
Computer
Programming
and
Applications

Address

pointer

Pointer
Arithmetic

Functions and
Arrays

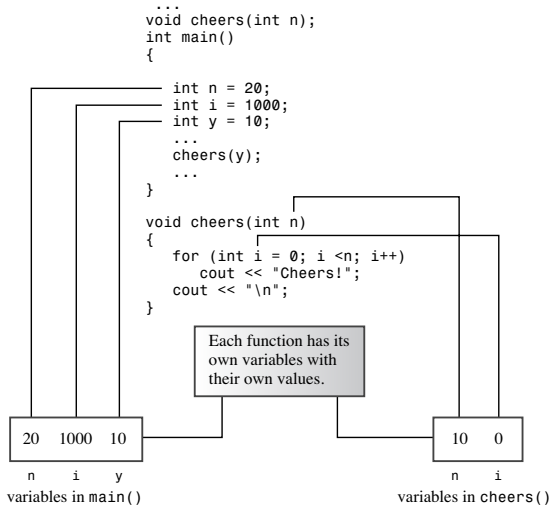
1 Address

2 pointer

3 Pointer Arithmetic

4 Functions and Arrays

Local Variables



local variable

MA122 -
Computer
Programming
and
Applications

Address

pointer

Pointer
Arithmetic

Functions and
Arrays

```
1  #include <iostream>
2  using namespace std;
3  void n_chars(char, int);
4  int main()
5  {
6      int times;
7      char ch;
8
9      cout << "Enter a character: ";
10     cin >> ch;
11
12     while (ch != 'q')        // q to quit
13     {
14         cout << "Enter an integer: ";
15
16         cin >> times;
17
18         n_chars(ch, times); // function with two arguments
```

local variable

```
1      cout << "\nEnter another character or press the"
2      " q-key to quit: ";
3      cin >> ch;
4  }
5
6
7      cout << "The value of times is " << times << ".\n";
8      cout << "Bye\n";
9      return 0;
10 }
11
12 void n_chars(char c, int n) // displays c n times
13 {
14     while (n-- > 0)          //continue until n reaches 0
15         cout << c;
16 }
```

Array in Function

```
1 #include <iostream>
2 const int ArSize = 8;
3
4 int sum_arr(int arr[], int n);    // prototype
5 int main()
6 {
7     using namespace std;
8
9     int cookies[ArSize] = {1,2,4,8,16,32,64,128};
10
11
12     int sum = sum_arr(cookies, ArSize);
13
14     cout << "Total cookies eaten: " << sum << "\n";
15
16     return 0;
17 }
```

Array in Function

MA122 -
Computer
Programming
and
Applications

Address

pointer

Pointer
Arithmetic

Functions and
Arrays

```
1
2 // return the sum of an integer array
3
4 int sum_arr(int arr[], int n)
5 {
6     int total = 0;
7
8     for (int i = 0; i < n; i++)
9         total = total + arr[i];
10
11     return total;
12 }
```

Array range in Function

```
1 #include <iostream>
2 const int ArSize = 8;
3 int sum_arr(int arr[], int n);
4 // use std:: instead of using directive
5 int main()
6 {
7     int cookies[ArSize] = {1,2,4,8,16,32,64,128};
8
9     std::cout << cookies << " = array address, ";
10
11     std::cout << sizeof cookies << " = sizeof cookies\n"
12         ;
13     int sum = sum_arr(cookies, ArSize);
14     std::cout << "Total cookies eaten: " << sum << std::
15         endl;
```

Array range in Function

MA122 -
Computer
Programming
and
Applications

Address

pointer

Pointer
Arithmetic

Functions and
Arrays

```
1  sum = sum_arr(cookies, 3);  
2  
3  std::cout << "First three eaters ate " << sum << "  
    cookies.\n";  
4  
5  sum= sum_arr(cookies + 4, 4);  
6  
7  std::cout << "Last four eaters ate " << sum << "  
    cookies.\n";  
8  
9  return 0;  
10 }
```

Array range in Function

```
1
2 int sum_arr(int arr[], int n)
3 {
4     int total = 0;
5
6     std::cout << arr << " = arr, ";
7
8     std::cout << sizeof arr << " = sizeof arr\n";
9
10    for (int i = 0; i < n; i++)
11        total = total + arr[i];
12
13    return total;
14 }
```

Array in Function Using Pointer

```
1 #include <iostream>
2 const int ArSize = 8;
3
4 int sum_arr(const int * begin, const int * end);
5 int main()
6 {
7     using namespace std;
8
9     int cookies[ArSize] = {1,2,4,8,16,32,64,128};
10
11     int sum = sum_arr(cookies, cookies + ArSize);
12     cout << "Total cookies eaten: " << sum << endl;
13
14     sum = sum_arr(cookies, cookies + 3);    // first 3
15     cout << "First three eaters ate " << sum << "
        cookies.\n";
```

Array in Function Using Pointer

MA122 -
Computer
Programming
and
Applications

Address

pointer

Pointer
Arithmetic

Functions and
Arrays

```
1  sum = sum_arr(cookies + 4, cookies + 8); // last 4
    elements
2  cout << "Last four eaters ate " << sum << " cookies.\n";
3  return 0;
4  }
5
6
7  int sum_arr(const int * begin, const int * end)
8  {
9      const int * pt;
10     int total = 0;
11
12     for (pt = begin; pt != end; pt++)
13         total = total + *pt;
14     return total;
15 }
```