MA122 - Computer Programming and Apllications

Indian Institute of Space Science and Technology

February 02, 2017

Lecture 8

string initialization


```
1  char dog[8] = { 'b', 'e', 'a', 'u', 'x', ' ', 'I',
       'I'}; // not a string!
2
3  char cat[8] = {'f', 'a', 't', 'e', 's', 's', 'a',
       '\0'}; // a string!
4
5  char bird[11] = "Mr. Cheeps"; // the \0 is
       understood
6
7  char fish[] = "Bubbles"; // let the compiler count
```

- When determining the minimum array size necessary to hold a string, remember to include the terminating null character in your count.

null character


```
char boss[8] = "Bozo";
```

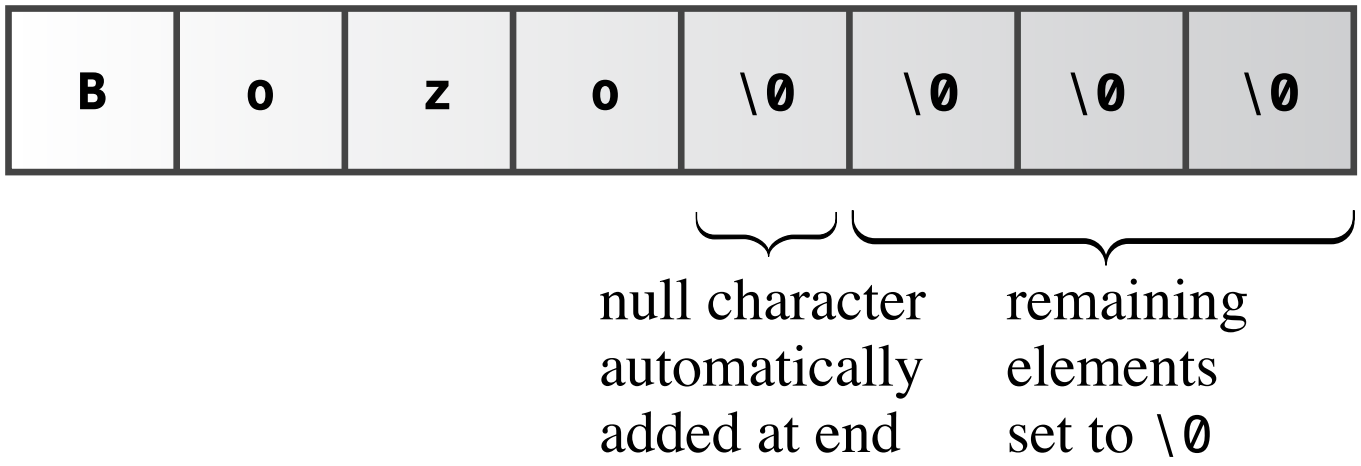


Figure 4.2 Initializing an array to a string.

string initialization


```
1  #include <iostream>
2  int main()
3  {
4    using namespace std;
5
6    char shirt_length = 'A';
7
8    //this is fine, A is a character constant
9
10   char shirt_size = "S"; // illegal type mismatch,
11
12   // "S" is not a character constant; It represents
13   //the string consisting of two characters, the S
14
15   //and the \0 characters.
16
17   return 0;
18 }
```

Using strings in a array

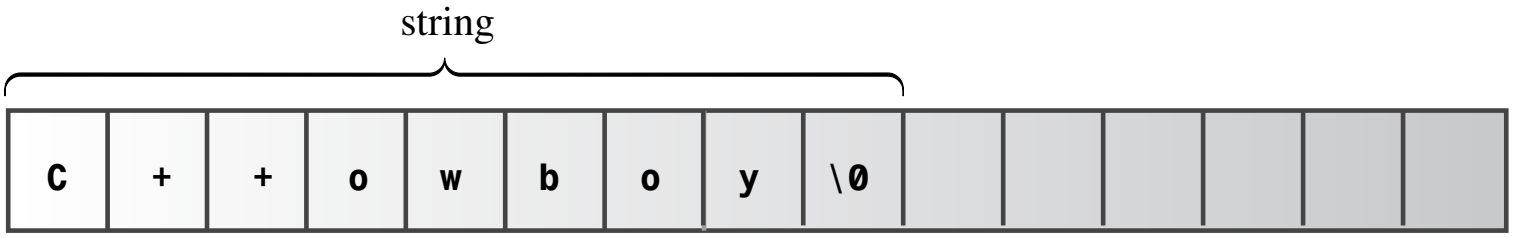

```
1  #include <iostream>
2  #include <cstring> // for the strlen() function
3  int main()
4  {
5    using namespace std;
6    const int Size = 15;
7
8    char name1[Size];              // empty array
9
10   char name2[Size] = "C++"; // initialized array
11
12   cout << "Hello! I'm " << name2;
13
14   cout << "! What's your name?\n";
15
16   cin >> name1;
```

Using strings in a array

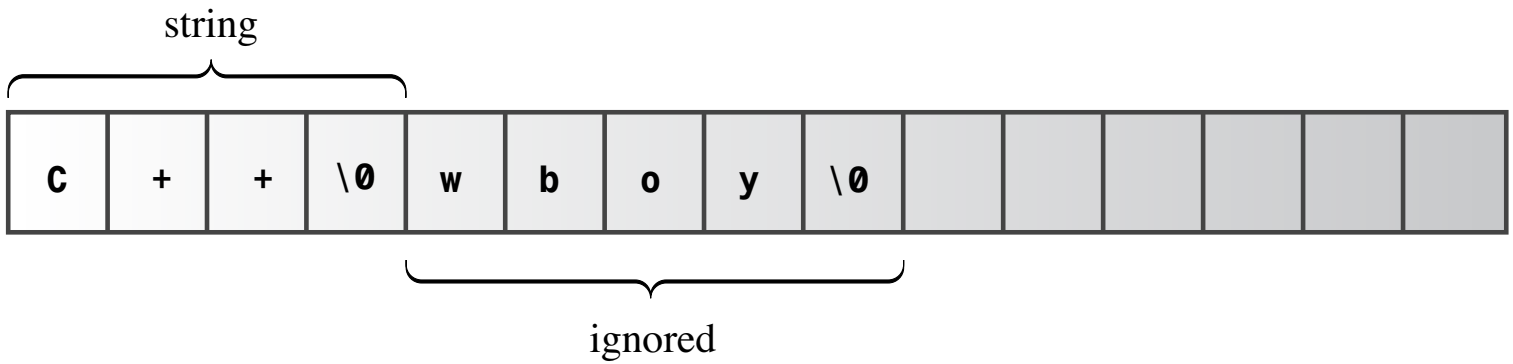

```
1  cout << "Well, " << name1 << ", your name has ";
2
3  cout << strlen(name1) << " letters and is stored\n";
4
5  cout << "in an array of " << sizeof(name1) << " bytes
       .\n";
6
7  cout << "Your initial is " << name1[0] << ".\n";
8
9  name2[3] = '\0';                    // set to null character
10
11 cout << "Here are the first 3 characters of my name: "
       ;
12
13 cout << name2 << endl;
14 return 0; }
```

Using strings in a array


```
const int ArSize = 15;
char name2[ArSize] = "C++owboy";
```



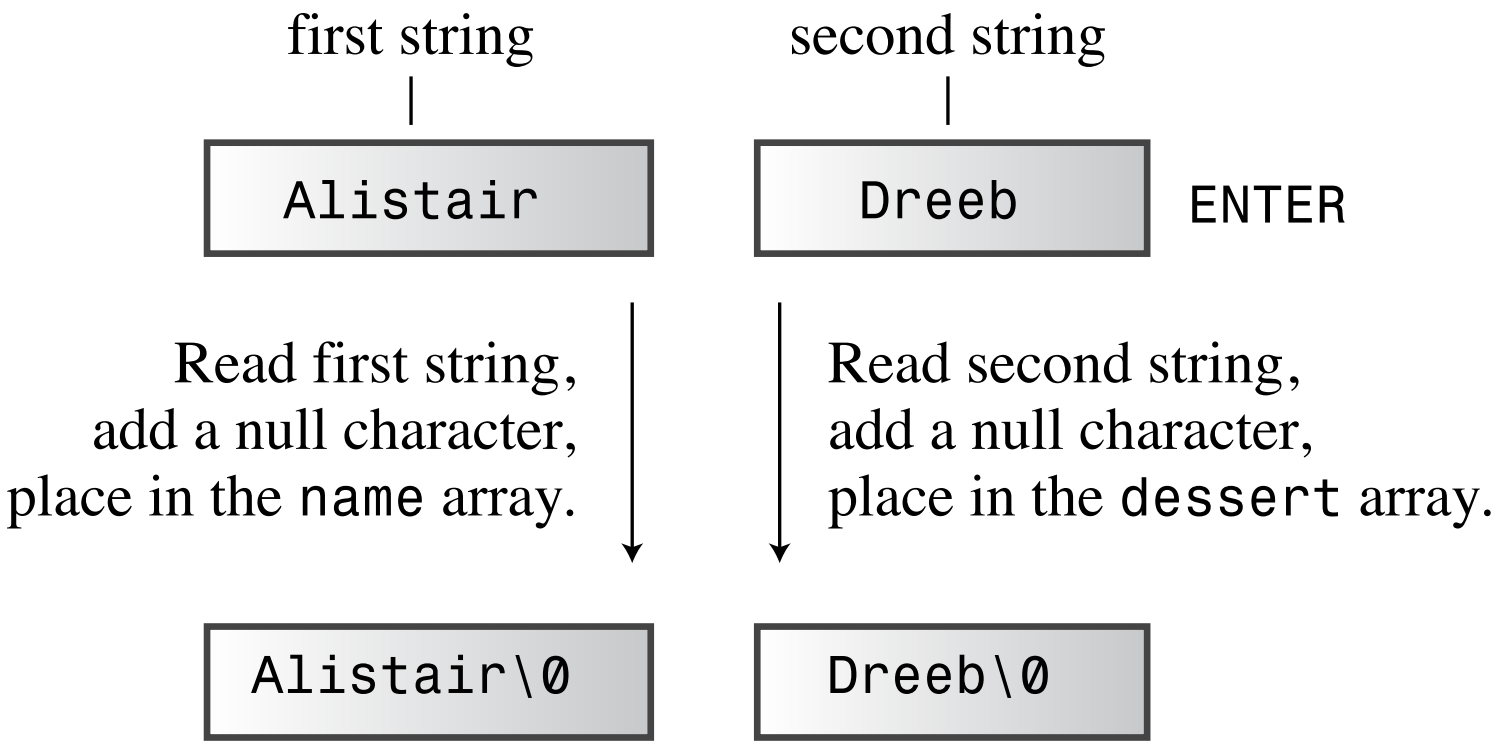

```
name2[3] = '\0';
```



Adventures in string input


```
1  // -- reading more than one string
2  #include <iostream>
3  int main()
4  {
5  using namespace std;
6  const int ArSize = 20;
7  char name[ArSize];
8  char dessert[ArSize];
9
10 cout << "Enter your name:\n";
11 cin >> name;
12 cout << "Enter your favorite dessert:\n";
13 cin >> dessert;
14
15 cout << "I have some delicious " << dessert;
16 cout << " for you, " << name << ".\n";
17 return 0; }
```

Adventures in string input



Reading string input a line at a time


```
1  //-- reading more than one word with getline
2  #include <iostream>
3  int main()
4  {
5  using namespace std;
6  const int ArSize = 20;
7  char name[ArSize];
8  char dessert[ArSize];
9
10 cout << "Enter your name:\n";
11 cin.getline(name, ArSize); // reads through newline
12
13 cout << "Enter your favorite dessert:\n";
14 cin.getline(dessert, ArSize);
15
16 cout << "I have some delicious " << dessert;
17 cout << " for you, " << name << ".\n";
18 return 0; }
```

Lecture 8

```
statement1
for (int_expr; test_expr; update_expr)
     statement2
statement3
```

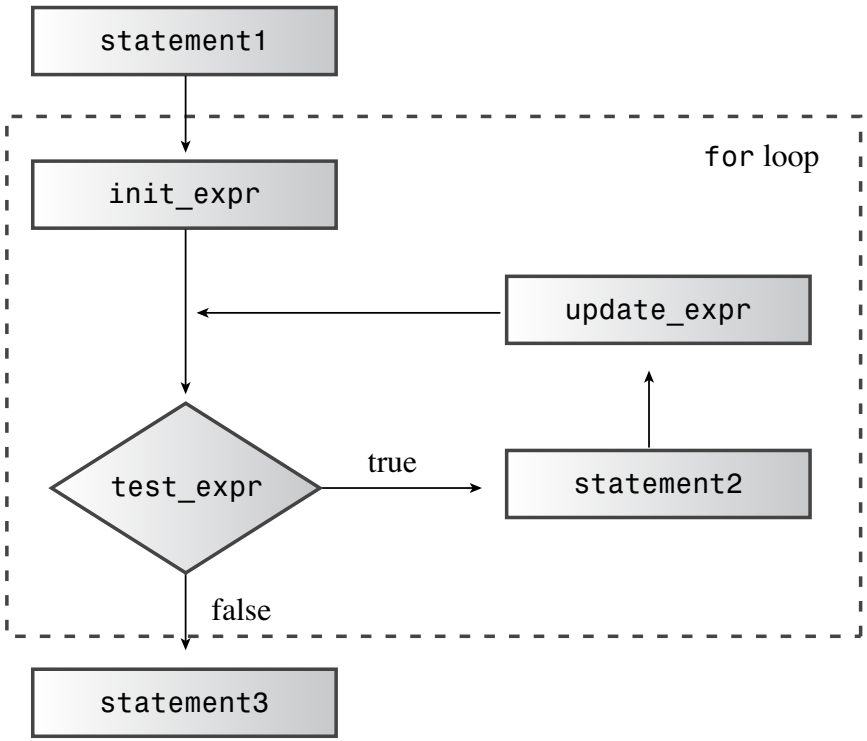


Figure 5.1 The design of `for` loops.

for loop


```
1  // forloop.cpp -- introducing the for loop
2  #include <iostream>
3  int main()
4  {
5    using namespace std;
6
7    int i; // create a counter
8    //   initialize; test ; update
9
10   for (i = 0; i < 5; i++)
11     cout << "C++ knows loops.\n";
12
13   cout << "C++ knows when to stop.\n";
14   return 0;
15 }
```

for loop


```
1  // num_test.cpp -- use numeric test in for loop
2  #include <iostream>
3  int main()
4  {
5    using namespace std;
6
7    cout << "Enter the starting countdown value: ";
8    int limit;
9    cin >> limit;
10
11   short i;
12   for (i = limit; i; i--)  // quits when i is 0
13     cout << "i = " << i << "\n";
14
15   cout << "Done now that i = " << i << "\n";
16   return 0;
17 }
```

factorial program


```
1  // -- more looping with for
2  #include <iostream>
3  const int ArSize = 16; // example of external
       declaration
4
5  int main()
6  {
7    short factorials[ArSize];
8    factorials[1] = factorials[0] = 1LL;
9
10   for (int i = 2; i < ArSize; i++)
11     factorials[i] = i * factorials[i-1];
12   for (int i = 0; i < ArSize; i++)
13     std::cout << i << "! = " << factorials[i] << std::
           endl;
14   return 0;
15 }
```