

Ch 7. Arrays

Part 3

CS 1428
Fall 2011

Jill Seaman

Lecture 19

1

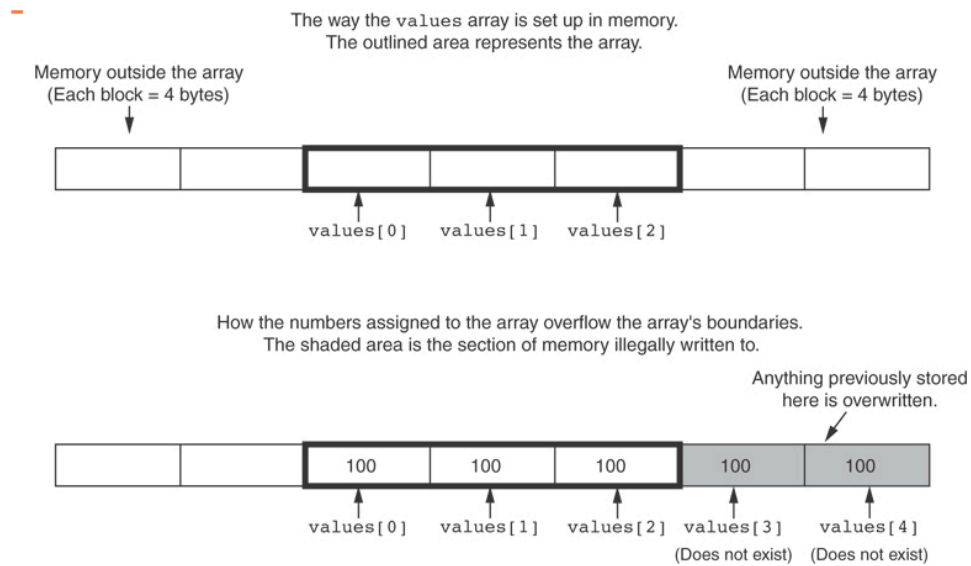
C++: No bounds checking

- When you use a value as an array subscript, C++ does not check it to make sure it is a valid subscript.
- In other words, you can use subscripts that are beyond the bounds of the array.

```
const int SIZE = 3;  
int values[SIZE];  
  
for (int i=0; i < 5; i++) {  
    values[i] = 100;  
}
```

2

What the code does



Watch out

- Be careful not to use invalid subscripts.
- Doing so can, without warning:
 - corrupt other memory locations
 - crash program
 - lock up computer
 - cause elusive bugs

Watch out: off by one

- It's easy to get the loop index off by one, especially if you
 - start at 1 instead of 0
 - use `<=` instead of `<`

```
// This code has an off-by-one error.  
const int SIZE = 100;  
int numbers[SIZE];  
for (int count = 1; count <= SIZE; count++)  
    numbers[count] = 0;
```

5

Parallel Arrays

- Parallel arrays: two or more arrays that contain related data
- A subscript is used to relate arrays: elements at same subscript are related, belong to the same entity
- Arrays may be of different types

6

Parallel Arrays

- Example: Employee hours worked and payrate

10	15	20	40	40
hours[0]	hours[1]	hours[2]	hours[3]	hours[4]
↑	↑	↑	↑	↑
Employee #1	Employee #2	Employee #3	Employee #4	Employee #5
↓	↓	↓	↓	↓
9.75	8.62	10.50	18.75	15.65
payRate[0]	payRate[1]	payRate[2]	payRate[3]	payRate[4]

7

Parallel Arrays

- Example: Employee hours worked and payrate

```
const int NUM_EMPS = 5;    // Number of Employees
int hours[NUM_EMPS];       // Holds hours worked
double payRate[NUM_EMPS]; // Holds pay rates

cout << "Enter the hours worked and pay rates:\n";
for(int i = 0; i < NUM_EMPS; i++) {
    cout << "Hours worked by employee " << i+1 << ": ";
    cin >> hours[i];
    cout << "Hourly pay rate for employee " << i+1 << ": ";
    cin >> payRate[i];
}
```

8

Parallel Arrays

- Example: Cont.

```
cout << "Here is the gross pay for each employee:\n";
cout << fixed << setprecision(2);
for(int i = 0; i < NUM_EMPS; i++) {
    double grossPay = hours[i] * payRate[i];
    cout << "Employee " << i+1 << ": $";
    cout << grossPay << endl;
}
```

9

Parallel Arrays

- output

```
Enter the hours worked and pay rates:
Hours worked by employee 1: 10
Hourly pay rate for employee 1: 9.75
Hours worked by employee 2: 15
Hourly pay rate for employee 2: 8.62
Hours worked by employee 3: 20
Hourly pay rate for employee 3: 10.50
Hours worked by employee 4: 40
Hourly pay rate for employee 4: 18.75
Hours worked by employee 5: 40
Hourly pay rate for employee 5: 15.65
Here is the gross pay for each employee:
Employee 1: $97.50
Employee 2: $129.30
Employee 3: $210.00
Employee 4: $750.00
Employee 5: $626.00
```

10