CSC 211: Object Oriented Programming

Arrays, Arrays and Functions

Michael Conti

Department of Computer Science and Statistics
University of Rhode Island

Spring 2020

Original design and development by Dr. Marco Alvarez

Question

- Write a program that reads in **3** values and outputs the same values in reverse order

- Write a program that reads in **n** values and outputs the same values in reverse order

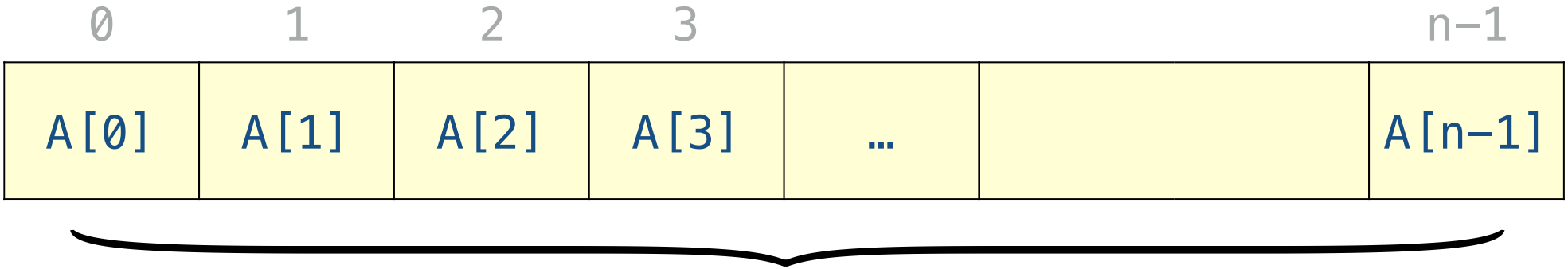
Arrays

Arrays

- An array is a **contiguous** sequence of elements of the **same type**

- Each element can be accessed using its **index**

array name: A
array length: n



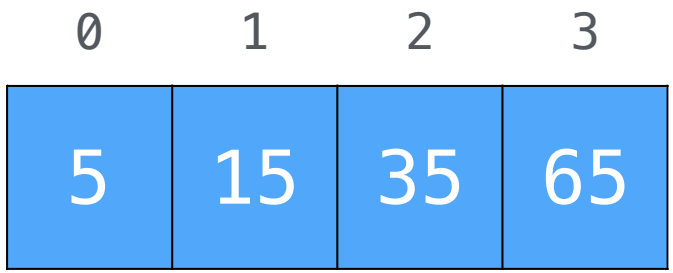
Declaration

```
// array declaration by specifying size
int myarray1[100];

// can also declare an array of
// user specified size
int n = 8;
int myarray2[n];

// can declare and initialize elements
double arr[] = { 10.0, 20.0, 30.0, 40.0 };
// compiler figures the right size

// a different way
int arr[5] = { 1, 2, 3 };
// compiler creates an array of length 5 and
// initializes first 3 elements
```

Initialization and indexing

- Elements in an array **must be initialized** before use
  - otherwise, their initial values are **undetermined**
  - can use a loop to initialize values or `std::fill()`
- Individual elements can be accessed by using the **subscription operator [ ]**

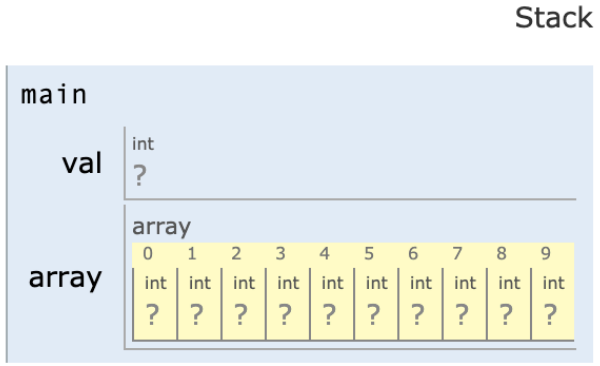
```
int array[4];
array[0] = 5;
array[1] = array[0] + 10;
array[2] = array[1] + 20;
array[3] = array[2] + 30;
```



Trace the code

C++ (gcc 4.8, C++11)
EXPERIMENTAL! known bugs/limitations

```
1  int main() {
2    int val = 0;
3    int array[10];
4
5    for (int i = 0 ; i < 10 ; i++) {
6      val += 50;
7      array[i] = val;
8    }
9
10   return 0;
11 }
```



http://pythontutor.com/cpp.html

Out of bounds?

- There is no **out of bounds** checking at compile time
  - ✓ **unexpected output**

A[9] ?

	0	1	2	3	4	5	6	7		
?	?	?	10	20	50	100	70	50	30	5	?	?

What is the output?

```cpp
#include <iostream>

int main() {
    int myarray[5];
    for (int i = 0 ; i < 5 ; i++) {
        myarray[i] = i;
    }
    for (int i = -10 ; i < 10 ; i++) {
        std::cout << myarray[i] << ' ';
    }
    std::cout << '\n';
    return 0;
}
```

Computer memory

- A **memory address** is a reference to a specific memory location
- Memory addresses are **fixed-length** sequences of digits (hexadecimal codes)
- **Word**-oriented memory organization (word size 32-bit in this illustration)

0x00000000	
0x00000004	
0x00000008	
0x0000000C	
0x00000010	
0x00000014	
0x00000018	
…	
…	
…	
0xFFFFFFEC	
0xFFFFFFF0	
0xFFFFFFF4	
0xFFFFFFF8	
0xFFFFFFFC	

address content

https://en.wikipedia.org/wiki/Random-access_memory

Computer memory (example)

```
int main() {
    int a = 4;
    int i = 0;
    double b = 10;
    int arr[5];

    for (; i < 5 ; i++) {
        arr[i] = i * 100;
    }

    return 0;
}
```

Assuming 32-bit words

…	
0x91340A04	
0x91340A08	4
0x91340A0C	5
0x91340A10	10
0x91340A14	
0x91340A18	0
0x91340A1C	100
0x91340A20	200
0x91340A24	300
0x91340A28	400
0x91340A2C	
0x91340A30	
0x91340A34	
…	

Passing arrays to functions

- When specifying the parameter, use **empty brackets**
- When providing the argument, use the **array name**
  - need to pass the **array length** separately

```
void zeros(int a[], int n) {
    for (int i = 0 ; i < n ; i ++) {
        a[i] = 0;
    }
}

int main() {
    int array[5];
    zeros(array, 5);
    // do stuff
}
```

Base address

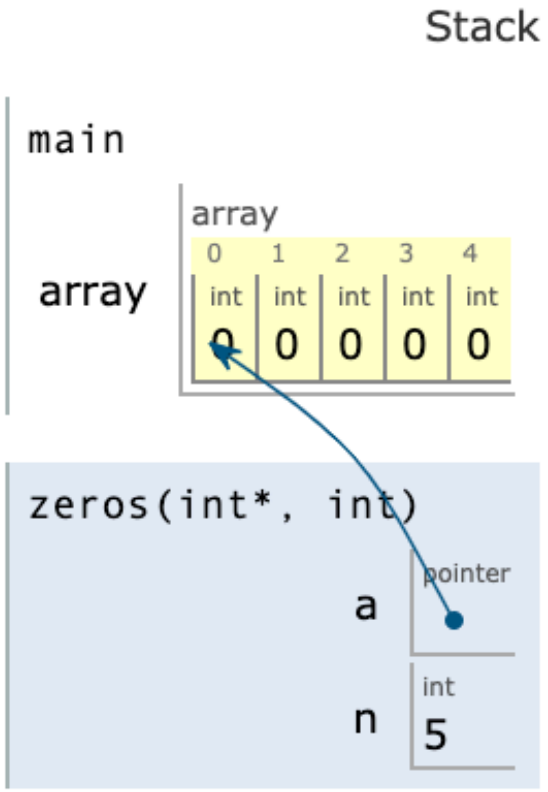
- **Base address** is the memory location of the first element in an array
  - base address of `arr` is `0x91340A18` (previous example)

- When passing arrays to functions, the base address of the array is passed to the formal parameter

Passing arrays to functions

C++ (gcc 4.8, C++11)
EXPERIMENTAL! known bugs/limitations

```
1  void zeros(int a[], int n) {
2      for (int i = 0 ; i < n ; i ++) {
3          a[i] = 0;
4      }
5  }
6
7  int main() {
8      int array[5];
9      zeros(array, 5);
10     // do stuff
11 }
```



http://pythontutor.com/cpp.html

Question

- Write a function that receives an array of integers and reverses the contents of the array