Computer Programming

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: Arrays in C++

Quick Recap

- Need to handle a large number of values
- Instead of many variables with different names, we need
  - A single name to represent a set of variables
  - Individual elements should be accessed using an index

Overview of This Lecture

- Array data structure in C++, and its properties
- Accessing elements of an array
  - Index expressions
- Example of use of arrays in a C++ program

Array in C++

- Array is a collection of elements of the same type
  - It has a name, chosen by us, and a fixed size (number of elements)
- Declaring an array

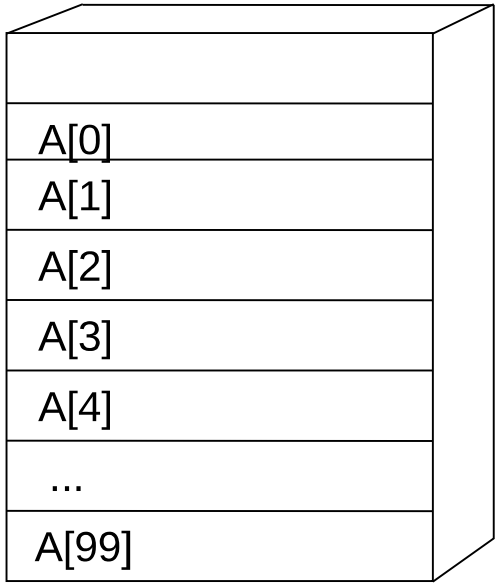
  int marks[500], roll_numbers[500]; float distances[25];
- **Only one element** participates in an operation
  - Input or output
  - As an operand in an expression
  - As a location on LHS of an assignment statement

Index expression for an array element

- **Index expression** is used to refer to an element
  - Written in square brackets [ ], immediately after the array name
  - the **value** of index expression is the actual reference.
- Thus when we use a[i]
  - if i is equal to 5 at that point, refers to 5th element
  - if i is equal to 123, 123rd element
- **C++ array index starts with 0**

A pictorial view of C++ array

IIT Bombay



- Array A has 100 int elements

  int A[100];

- Suppose it stores 5 values

  53, 79, 41, 94, 38

  A[0] = 53
  A[1] = 79
  A[2] = 41
  A[3] = 94
  A[4] = 38

Index expression

- Index can be any expression, which will be evaluated first, the resulting integer value is used to identify a particular element.
- An index expression must result in a value in the range 0 to size-1, where size is as declared in the definition
  - If not, the results are **unpredictable**

Index expression …

IIT Bombay

- Index for the first element is 0, for next element it is 1, etc.

  A[j] , for j = 86 means 86th element

  A[k-m+52], for k = 1200, m=1240

  the index value evaluates to 12

  A[x/5.0] for x = 7.0, expression evaluates to 1

  for x = 22.5, it evaluates to 4

A program to find sum of marks of N students


```
int main(){
   // program to find the sum of N marks
   int marks[600] , sum = 0, count, N;
   cin >> N;
   for (count =1; count <= N; count = count +1){
         cin >> marks[count]; sum = sum + marks[count];
   }
   cout << sum;
return 0;
}
```

corrected program to find sum of marks of N students


```
int main(){
   // program to find the sum of N marks
   int marks[600] , sum = 0, count, N;
   cin >> N;
   for (count =0; count < N; count = count +1){
         cin >> marks[count]; sum = sum + marks[count];
   }
   cout << sum;
return 0;
}
```

Program to find the value of largest element


```
int main(){
      int a[1000], max, N, i;
      cin >> N;
      for (i=0; i < N; i++) {cin >> a[i];}
      max = a[0];
      for (i=1; i < N; i++) {
           if (a[i] > max) {  max = a[i];}
       }
       cout << “maximum Value is  ”<< max << endl;
       return 0;
}
```

Summary

- We have learnt how to declare arrays in C++
- Must declare its name and size. Size must be an integer value
- We have learnt how to use an array
- Using an index expression
  - must evaluate to an integer value between 0 and size-1
- We can iterate over an index variable, to successively scan/process all elements of the array