Chapter 6
Arrays/Vectors

CISC 5004 - Programming C++
Sam Kamens
Fall, 2022

FORDHAM

FORDHAM

Array/vector concept (General)

- Variable stores one data item
  - `numPlayers`
- Array/vector stores a list of items
  - `pointsPerQuarter`
- Analogy: truck vs. van
- Index: Location number within the array
  - Start with 0
- Array vs. vector (more later)
  - Array: C-style
  - Vector: C++-style

FORDHAM

Vectors

- Declaration:

```
    #include <vector>
    vector<dataType> vectorName(numElements);
```

- E.g.

```
    vector<string> userNames(5)
```

- Angle brackets `<>` as opposed to parentheses `()` or braces `{}`
- numElements is strictly optional, but recommended
- Access elements using .at()
- A vector is an instance of a **standard C++ container** (more later)
  - Containers provide many of the same functions

FORDHAM

Vector initialization

- Elements are initialized to their default values when declared
  - 0 for int, char
  - Empty for strings
  - Undefined for other data types
- Can explicitly initialize the values
  - `vector<int> carSales = {5, 7, 11};`
  - Note size is not specified here
- Can also initialize after declaration using a loop
- Drills

FORDHAM

Iterating through vectors

- Common for loop structure

```
// Iterating through myVector
for (i = 0; i < myVector.size(); ++i) {
   // Loop body accessing myVector.at(i)
}
```

- Find the sum
- Find max value
- Watch for out of range accesses
- Note `[]` also works

FORDHAM

Multiple vectors

- Multiple same-size vectors for related data
  - Spoiler: there are much better ways to do this!
- TV watching time

FORDHAM

resize()

- `resize()`: Explicitly change the size of a vector
  - If increasing, new elements are added (and default-initialized)
  - If decreasing, elements are removed
    - This might not involve releasing memory, though - see `capacity()`
- Resize based on user input

FORDHAM

push_back()

- `push_back()` - Append an element to the end of a vector
  - Automatically does `resize()` as needed
  - Example
- Related functions
  - `back()`: Returns the last element of the vector
  - `pop_back()`: Removes the last element (Note: Does not return anything!)
- Grocery List example

FORDHAM

Modifying elements

- Modify during iteration
- Vector copy
  - `vectorB = vectorA;`
  - Element by element copy
  - Destination is resized as needed
  - Example
- Vector comparison
  - `vectorA == vectorB;`
  - Compares element-by-element
  - `true` if vectors are the same size and each element pair is equal

Swapping two variables (general)

- Analogy - switch hands
- Similar for variables
- Useful concept for many algorithms
  - Reversing a list using swaps
  - Sorting
- Debugging example: reversing a vector

FORDHAM

Arrays vs. Vectors

- Array
  - `int myList[10]`
  - `myList[i]`
- Vectors
  - `vector<int> myList(10)`
  - `myList.at(i)`
  - `myList[]` also works, but is not bounds-checked
- Which to use?
  - Generally better to use vectors, unless performance is extremely critical

Two-dimensional arrays

- Declare array with two dimensions (row/column, x/y)
  - `int myArray[numRows][numColumns];`
- Access using two indices
  - `myArray[1][2];`
  - `myArray[1] is a one-dimensional array`
  - Representation
- Note: can do this with vectors too
  - `vector< vector<int> > vectorOfVectors;`
- Example: Distance between cities

FORDHAM

Two-dimensional arrays

- Initializing during declaration

```
// Initializing a 2D array
int numVals[2][3] = { {22, 44, 66}, {97, 98, 99} };

// Use multiple lines to make rows more visible
int numVals[2][3] = {
   {22, 44, 66}, // Row 0
   {97, 98, 99}  // Row 1
};
```

Character arrays (C strings)

- Strings in C
  - Sequence of characters in an array
  - Referred to as C strings, as opposed to C++ string (std::string)
- `char bestSchool[20] = “Fordham”;`
- Ends with a null character (`‘\0’`) (ASCII 0)
  - aka null-terminated string
  - Added for you by the compiler
  - String length is one shorter than you might think
  - Output streams handle these automatically: example

FORDHAM

C Strings

- Can not assign string value after declaration

```
char movieTitle[] = “Avengers: Infinity War”;
movieTitle = “Avengers: End Game”; // Does not work
```

  - There are special functions to manipulate these strings
    - e.g. `strlen`, `strcpy`
- Traversal via a loop
  - Remember to stop at the null character!
  - Example
- Careful loading a string longer than the array
- Explictly populating the array

FORDHAM

C-string functions

- `#include <cstring>`
- Modification Functions
  - `strcpy(destStr, srcStr);`
  - `strncpy(destStr, srcStr, numChars);`
  - `strcat(destStr, srcStr);`
  - `strncat(destStr, srcStr, numChars);`
- Information Functions
  - `strchr(str, char);`
  - `strlen(str);`
  - `strcmp(str1, str2)`
    - Note: `==` does not work!

FORDHAM

C-string functions

- Examples
  - strcmp
    - Returns negative if s1 < s2, positive if s1 > s2, 0 if equal
  - Iterate with strlen
  - More functions
- For more info: http://www.cplusplus.com/reference/cstring/

char Library Functions: ctype

- Common character functions
- `#include <cctype>`
- Checking functions
- Conversion functions
- Example
- For more info: http://www.cplusplus.com/reference/cctype/

Coding examples/labs

- Tax Rate Calculation
- Word frequencies