# Common array patterns

There are many common patterns for using arrays, here are just a few that you will encounter.

1. Treat an array element like any variable:
   1. **++score[5];**
   2. **score[3]--;**
   3. **pay = hours[3] \* rate;**
2. The index is an **int**, watch the difference between the following:
   1. **score[counter++]**
   2. **score[++counter]**
   3. **score[counter]++**
3. You cannot copy one array into another with the assignment operator (**=**). You must copy element-by-element:
   1. **for (counter = 0; counter < ARRAY\_SIZE) counter++){**
   2. **array2[counter] = array1[counter];**
   3. **}**
4. You cannot print the contents of the array in one print statement (**cout**). You must print element-by-element:
   1. **for (counter = 0; counter < ARRAY\_SIZE) counter++){**
   2. **cout << array1[counter] << endl;**
   3. **}**
5. You use a loop and an accumulator to sum array elements:
   1. **double sum = 0.0;     // Initialize variable for sum**
   2. **for (counter = 0; counter < ARRAY\_SIZE) counter++){**
   3. **sum = sum + array1[counter]; // what is the shortcut operator?**
   4. **}**
6. You use a loop, an accumulator and a variable to average array elements:
   1. **double sum = 0.0;     // Initialize variable for sum**
   2. **double average = 0.0; // Initialize variable for average**
   3. **for (counter = 0; counter < ARRAY\_SIZE) counter++){**
   4. **sum += array1[counter]; // what is the shortcut operator?**
   5. **}**
   6. **average = sum / ARRAY\_SIZE;**
7. Find the highest and lowest values
   1. **double highest = array1[0];   // Initialize highest to first element**
   2. **double lowest = array1[0];    // Initialize lowest to first element**
   3. **for (counter = 1; counter < ARRAY\_SIZE) counter++){ //start at 1!**
   4. **if ( array1[counter] < lowest )**
   5. **lowest = array1[counter];**
   6. **if ( array1[counter] > highest )**
   7. **highest = array1[counter];**
   8. **}**
8. You can use a while loop to create partially filled arrays
   1. **int num;**
   2. **cout << "Enter a number or −1 to quit: ";**
   3. **cin >> num;**
   4. **while (num != −1 && count < SIZE)**
   5. **{**
   6. **count++;**
   7. **numbers[count − 1] = num;**
   8. **cout << "Enter a number or −1 to quit: ";**
   9. **cin >> num;**
   10. **}**
9. You cannot compare arrays using in as single line of code using a comparison operator (**==**), you must compare element-by-element.
   1. **const int SIZE = 5;**
   2. **int firstArray[SIZE] = { 5, 10, 15, 20, 25 };**
   3. **int secondArray[SIZE] = { 5, 10, 15, 20, 25 };**
   4. **bool arraysEqual = true; // Flag variable**
   5. **int count = 0;           // Loop counter variable**
   6. **// Determine whether the elements contain the same data.**
   7. **while (arraysEqual && count < SIZE)**
   8. **{**
   9. **if (firstArray[count] != secondArray[count])**
   10. **arraysEqual = false;**
   11. **count++;**
   12. **}**
   13. **if (arraysEqual)**
   14. **cout << "The arrays are equal.\n";**
   15. **else**
   16. **cout << "The arrays are not equal.\n"**