Swinburne University of Technology

School of Science, Computing and Engineering Technologies

LABORATORY COVER SHEET

Subject Code:	COS30008
Subject Title:	Data Structures and Patterns
Lab number and title:	1, Print an Array of Integers in C++
Lecturer:	Dr. Markus Lumpe

[45,34,8,6,5,1,0,-2,-3,-100]
[34,45,8,6,5,1,0,-2,-3,-100]
[8,34,45,6,5,1,0,-2,-3,-100]
[6,8,34,45,5,1,0,-2,-3,-100]
[5,6,8,34,45,1,0,-2,-3,-100]
[1,5,6,8,34,45,0,-2,-3,-100]
[0,1,5,6,8,34,45,-2,-3,-100]
[-2,0,1,5,6,8,34,45,-3,-100]
[-3,-2,0,1,5,6,8,34,45,-100]
[-100,-3,-2,0,1,5,6,8,34,45]
[-100,-3,-2,0,1,5,6,8,34,45]

Figure 1: Output of Insertion Sort.

Print an Array of Integers in C++

How can we define a C++ function that prints an array of integers to the console?

An array is a contiguous storage location for elements of a specific data type. Unlike Java or C#, arrays in C++ do not carry runtime information. When working with arrays in C++, such as using an array as a function parameter, we also need to provide the size of an array. For instance, the function *printIntArray*, developed in this tutorial, takes two arguments: an unbounded array of integers and an unsigned integer representing the size of the array. We write

void printIntArray(**int** aArray[], size_t aSize);

to denote the signature of function *printIntArray*.

To test the function, use the following C++ program code:

```
#include <cstddef>

extern void printIntArray( int aArray[], size_t aSize );

int main()
{
  int lArray[] = { 23, 4, 67, 1, 0, -20, 29, -100, 32, 99, -56 };
  size_t lSize = sizeof(lArray) / sizeof(int);

  printIntArray( lArray, lSize );

  return 0;
}
```

The statement `#include<cstddef>` is a preprocessor directive to include the part of the utility library that defines type `size_t`, an unsigned integer type.

The statement

```
extern void printIntArray( int aArray[], size_t aSize );
```

specifies the signature of function *printIntArray* that is defined elsewhere. We use the keyword **`extern`** to denote the latter.

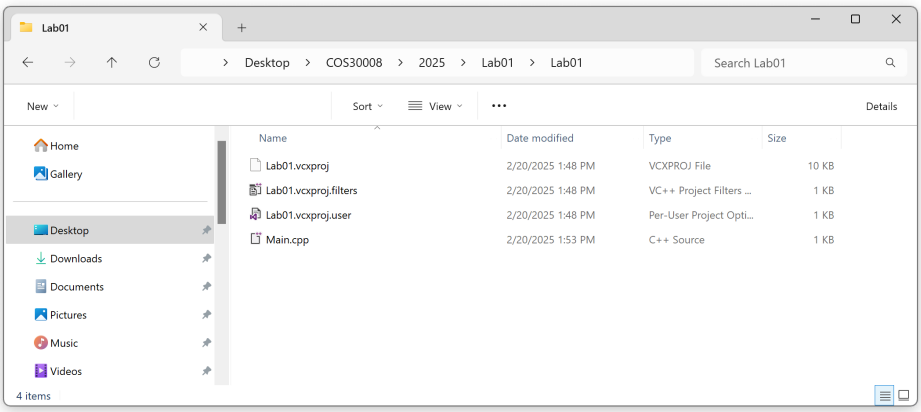
The `main` function declares an array of integers, `lArray`. Although we do not give it, C++ can implicitly determine the dimension using the specified array initializer. Array `lArray` contains 11 integer values, but the dimension is only used as a compile-time quantity. We need variable `lSize` to determine the size of `lArray` at runtime. For this reason, we define another variable, `lSize`, initialized by the expression **`sizeof`**`(lArray) / `**`sizeof`**`(int)`. This expression is calculated at compile-time. The **`sizeof`** operator yields the number of bytes occupied by a non-potentially-overlapping object of the type of its operand.

Once the call to `printIntArray` is completed, the `main` function returns value 0 to the runtime environment (or operating system) to indicate success. Non-zero values mean failure.

Problem 1

Create a new project (if you are uncertain about the necessary steps, refer to the document *LabNotesVS 30008*, available on Canvas).

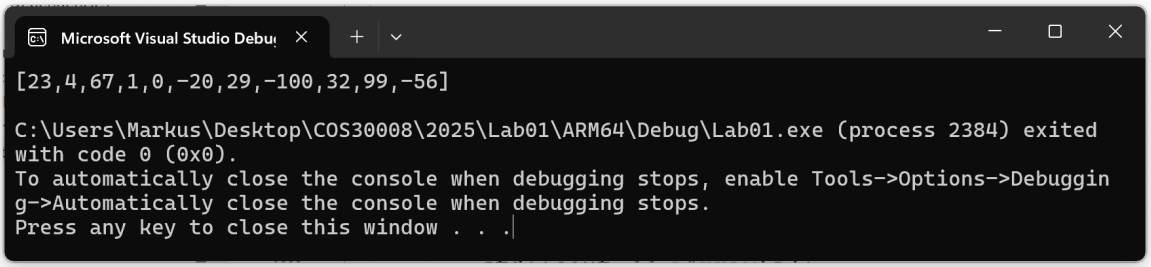
Add `Main.cpp`, available on Canvas, to your project. Copy `Main.cpp` into your project solution hierarchy, especially the leaf folder containing all project-specific artifacts.



Next, add a new item, `PrintIntArray.cpp`, to your project.

Implement the function `printIntArray` using the following requirements:

- The characters '[' and ']' enclose the output.
- The character ',' separates two adjacent elements.
- The output must end with a newline character.



To perform console output, you must include the library *iostream*, using the preprocessor directive `#include <iostream>` before defining function `printIntArray`.

Use `std::cout` and the operator `<<` to send data to the console. For instance, you can use the statement `std::cout << '[';` to send the character `'['` to the console. Similarly, you can write `std::cout << aArray[i];`, where variable `i` is the current index or loop variable, to send an integer from array `aArray` to output. Finally, you can use the statement `std::cout << std::endl;` to send the operating system-specific newline character to the console (see *LabNotesVS 30008* and *A Simple I/O Program* for examples or consult with the tutor).

How many tests does your implementation require? A test can be a loop condition and an if condition. Do not just count single occurrences. The number of tests is the sum of all checks of a loop condition and an if condition. For function `printIntArray`, the number must be **12** or n+1, where n is the number of elements in the array. If your number is greater, can you find a solution to revise `printIntArray` accordingly?

Complete this tutorial and check with your tutor.