1. What is an array - An array is a collection of elements, all the same type, stored in contiguous memory locations and accessible via an index.
2. Give the starting index and ending index of the array of the following statement: string name [20]

In C++ and many other programming languages that use zero-based indexing, the starting index of the array string name [20] is 0, and the ending index is 19.

3, when does an array go out of the range?

An array goes out of range when you attempt to access an element at an index that is outside the bounds of the array's declared size; for example, accessing an element before the first index (less than 0) or after the last index (greater or equal to the array's length).

1. Explain the difference between overflow and underflow of an array.

Array overflow happens when you access an element beyond the array's declared size, while underflow occurs when you attempt to access elements at negative indices.

1. Four elements of array are initialized for the following statement.

Double Number [8] = {2,4,7,8}

The declaration double Number [8] = {2, 4, 7, 8}; initializes the first four elements of a double-precision floating-point array of size 8 with the values 2.0, 4.0, 7.0, and 8.0, while the remaining four elements are automatically initialized to 0.0 by default in C++.

1. Give the size of array for the following statement: int Number [8];

The size of the array declared with the statement int Number [8]; is 8 elements.

1. Give the size of the array for the statement int number [] = {1,3,5,7,9,8,6}

Seven

1. Array’s name be treated as a pointer? Why or why not?

An array's name, in most contexts, is treated as a pointer to the first element of the array because it represents the address in memory where the array begins; however, an array is not a pointer itself, as it represents a block of memory, and its size is fixed, whereas a pointer is a variable that can be reassigned to point to different addresses.

1. Given the statement int size = 10, Number [size], write a function to read 10 integer numbers and store them in an array

#include <iostream>

#include <vector>

void readNumbers(std::vector<int>& numbers, int size) {

int input;

for (int i = 0; i < size; ++i) {

std::cout << "Enter number " << (i + 1) << ": ";

std::cin >> input;

numbers.push\_back(input);

}

}

int main() {

const int size = 10;

std::vector<int> Number;

readNumbers(Number, size);

// Optional: Output the numbers to verify they were stored correctly

std::cout << "You entered: ";

for (int num : Number) {

std::cout << num << " ";

}

std::cout << std::endl;

return 0;

10. For question 9, what are the inputs and informal parameters of the function?

In the readNumbers function of the program, the actual parameters are a std::vector<int> to hold the integers and an int specifying the count of numbers to read, while the formal parameters are a reference to a vector and an integer size that it uses to iterate the input process.

1. For question 9, did you use the keyword return?

Yes

1. What is a constant array?

A constant array is an array where the elements cannot be altered after initialization; they are declared with the const keyword, indicating their values are read-only.

1. When do you use a constant array?

You use a constant array when you want to store a list of values that should not be modified throughout the lifetime of the program, ensuring data integrity, and preventing accidental changes to the static or fixed data set.

1. What’s the number of iterations of the inner loop of a 4 \* 5 array?

The inner loop runs 5 times, resulting in 5 iterations for each of the 4 outer iterations.

1. what is the number of iterations of the inner loop of a 4 \* 5 \* 6 \* 10 array?

300

1. What is the number of iterations of the outer loop of a 5 \* 8 array?

5

1. Write a function to copy the content of array num\_1 into array num\_2

#include <iostream>

void copyArray(int\* num\_1, int\* num\_2, int size) {

for (int i = 0; i < size; ++i) {

num\_2[i] = num\_1[i];

}

}

int main() {

const int size = 5; // Example size

int num\_1[size] = {1, 2, 3, 4, 5}; // Example content

int num\_2[size]; // Destination array

// Copy content from num\_1 to num\_2

copyArray(num\_1, num\_2, size);

// Output num\_2 to verify contents

for (int i = 0; i < size; ++i) {

std::cout << num\_2[i] << " ";

}

std::cout << std::endl;

return 0;

}

1. For question number 17, write a function to compare the contents of 2 array.

#include <iostream>

bool compareArrays(const int\* arr1, const int\* arr2, int size) {

for (int i = 0; i < size; ++i) {

if (arr1[i] != arr2[i]) {

return false; // Arrays are not identical

}

}

return true; // Arrays are identical

}

int main() {

const int size = 5; // Example size

int array1[size] = {1, 2, 3, 4, 5}; // Example content for the first array

int array2[size] = {1, 2, 3, 4, 5}; // Example content for the second array

// Compare the two arrays

if (compareArrays(array1, array2, size)) {

std::cout << "The arrays are identical." << std::endl;

} else {

std::cout << "The arrays are not identical." << std::endl;

}

return 0;

}