*Thomas Baratta*

***C++ Practice Sheet***

***Arrays and Pointers***

**Topic: Arrays**

**1.1) Debugging a program to find the square of 1 - 4**

**Goal: Understand the concept of zero-indexed arrays in C++**

**Instructions:** The following code tries to fill an array with the squares of the numbers from 1 to 4, but it contains an error. Find and fix the error.

**#include <iostream>**

**int main()**

**{**

**int arr[5];**

**for (int number = 0; number < 5; number++)**

**{**

**arr[number] = number \* number;**

**}**

**for (int counter = 1; counter <= 5; counter++)**

**{**

**std::cout << arr[counter] << " ";**

**}**

**return 0;**

**}**

Explain your solution to the error:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1.2) Evaluating an Array**

**Goal: Understand the concept of looping through arrays**

**Instructions:** Review the code below. Predict the output of the code if the user were to input 1, 2, 3, 4, 5, 6. Display the manual execution to explain your answer.

**#include <iostream>**

**int main()**

**{**

**int arr[6];**

**for(int index = 0; index < 6; index++)**

**{**

**std::cout << "Enter a value for index " << index << ": ";**

**std::cin >> arr[index]; // Storing inputted index values**

**}**

**for(int reverseIndex = 5; reverseIndex >= 0; reverseIndex--)**

**{**

**std::cout << arr[reverseIndex] << " ";**

**}**

**return 0;**

**}**

Manual Execution:

Final Output:

**1.3) Two Sum**

**Goal: Understand the concept of nested for loops and function declarations**

**Instructions:** Review the code below. What will the output of the program be? Display the manual execution to explain your answer.

**#include <iostream>**

**void twoSum(int arr[], int size, int target)**

**{**

**bool foundPair = false; // Setting a flag to check for a pair**

**for(int i = 0; i < size - 1; i++) // i = outer loop counter**

**{**

**for(int j = i + 1; j < size; j++) // j = inner loop counter**

**{**

**if(arr[i] + arr[j] == target)**

**{**

**std::cout << arr[i] << " + " << arr[j] << " = " <<**  **target << '\n';**

**foundPair = true;**

**}**

**}**

**}**

**if(!foundPair)**

**{**

**std::cout << "No pairs of indices add to the target value";**

**}**

**}**

**int main()**

**{**

**int arr[5] = {1,2,3,4,5};**

**int target = 9;**

**int size = sizeof(arr) / sizeof(arr[0]); // # of elements in arr**

**twoSum(arr, size, target);**

**return 0;**

**}**

Manual Execution:

Final Output:

**Topic: Pointers**

**2.1) Pointer Basics**

**Goal: Understand the basic concepts of pointers in C++**

**Instructions:** Review the code below. What will the output of this program be? Display the manual execution to explain your answer.

**#include <iostream>**

**int main()**

**{**

**int number = 10; // Declare and initialize an integer**

**int \*ptr = &number; // Declare a pointer to address of number**

**std::cout << number << std::endl;**

**std::cout << \*ptr << std::endl;**

**std::cout << ptr << std::endl;**

**return 0;**

**}**

Manual Execution:

Final Output:

**2.2) Evaluating a Program using Pointers**

**Goal: Understand the basic concepts of pointers and assigned variables**

**Instructions:** Review the following code. What will the output of the program be before and after when “mysteryFunction” is called in int main()? Display the manual execution to explain your answer.

**#include <iostream>**

**void mysteryFunction(int\* x, int\* y)**

**{**

**int temp = \*x;**

**\*x = \*y;**

**\*y = temp;**

**}**

**int main()**

**{**

**int x = 5;**

**int y = 10;**

**std::cout << "x = " << x << " and y = " << y << std::endl;**

**mysteryFunction(&x, &y);**

**std::cout << "x = " << x << " and y = " << y << std::endl;**

**return 0;**

**}**

Manual Execution:

Final Output:

x = \_\_\_\_ and y = \_\_\_\_\_

x = \_\_\_\_ and y = \_\_\_\_\_

**2.3) Pointer Arithmetic Basics**

**Goal: Understand basic concepts of pointer arithmetic**

**Instructions:** Review the code below. What will the output of the program be? Display the manual execution to explain your answer.

**#include <iostream>**

**int main()**

**{**

**int arr[] = {10, 20, 30, 40, 50};**

**int\* ptr = arr;**

**std::cout << "Array elements are: ";**

**for (int i = 0; i < 5; i++) // i = index counter arr**

**{**

**std::cout << \*(ptr + i) << " ";**

**}**

**std::cout << std::endl;**

**\*(ptr + 2) = 100;**

**std::cout << "Modified array elements are: ";**

**for (int j = 0; j < 5; j++) // j = index counter modified arr**

**{**

**std::cout << \*(ptr + j) << " ";**

**}**

**std::cout << std::endl;**

**return 0;**

**}**

Manual Execution:

Final Output: Original Array:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Modified Array:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2.4) Finding sum and average of an array using pointer arithmetic**

**Goal: Apply your understanding of pointer arithmetic to find the sum and average of an array**

**Instructions:** Review the code below. What will the output of the program be if the user inputs 1, 2, 3, 4, 5? Display the manual execution to explain your answer.

**#include <iostream>**

**int main()**

**{**

**int arr[5];**

**int\* ptr = arr; // ptr points to first index in arr[]**

**int sum = 0;**

**std::cout << "Enter 5 integers:\n";**

**for (int inputIndex = 0; inputIndex < 5; inputIndex++)**

**{**

**std::cin >> arr[inputIndex];**

**}**

**for (int reverseIndex = 4; reverseIndex >= 0; reverseIndex--)**

**{**

**std::cout << \*(ptr + reverseIndex) << " ";**

**}**

**std::cout << "\n";**

**for (int sumIndex = 0; sumIndex < 5; sumIndex++)**

**{**

**sum += \*(ptr + sumIndex);**

**}**

**std::cout << "Sum: " << sum << "\n";**

**return 0;**

**}**

Manual Execution:

Final Output: