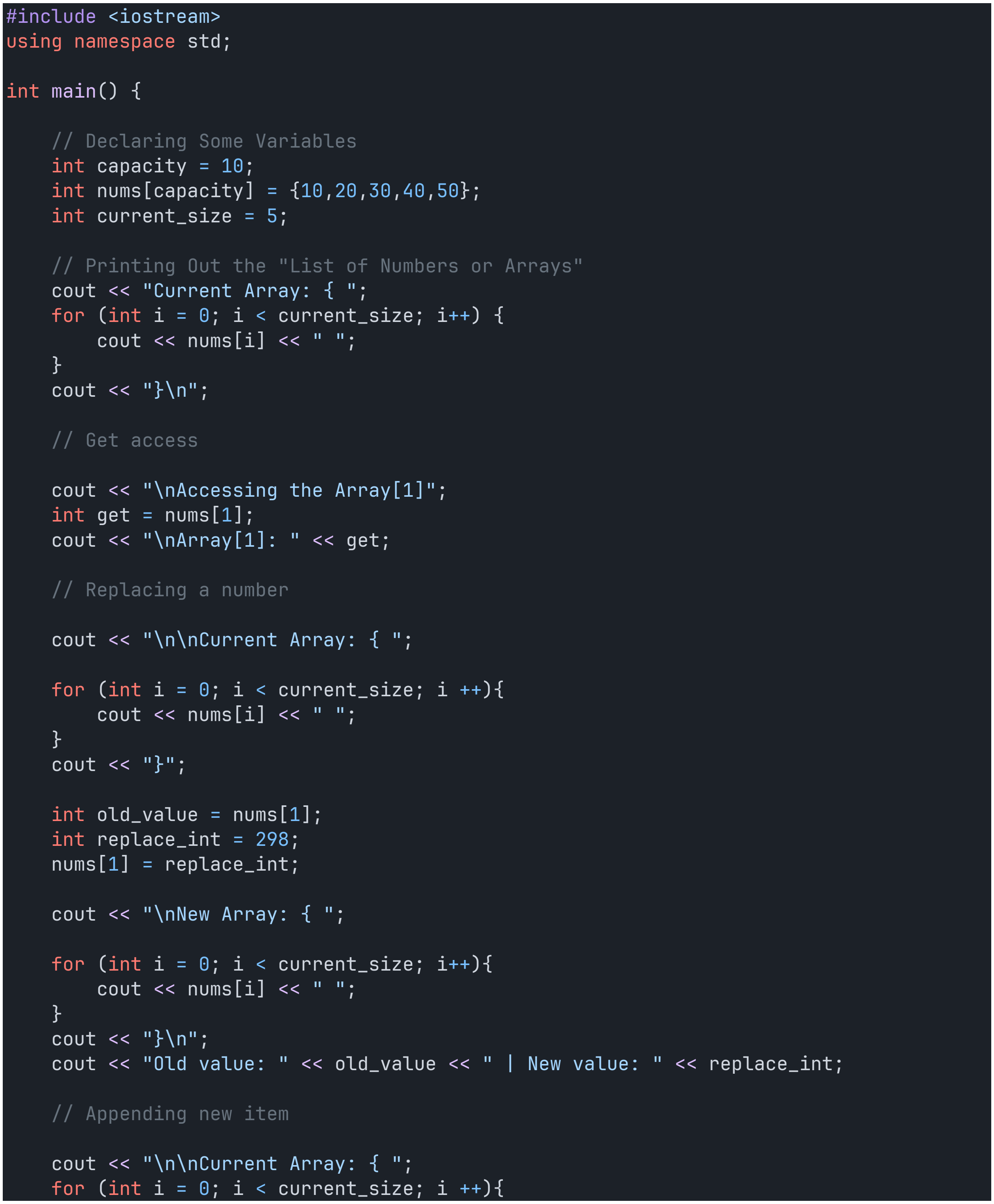
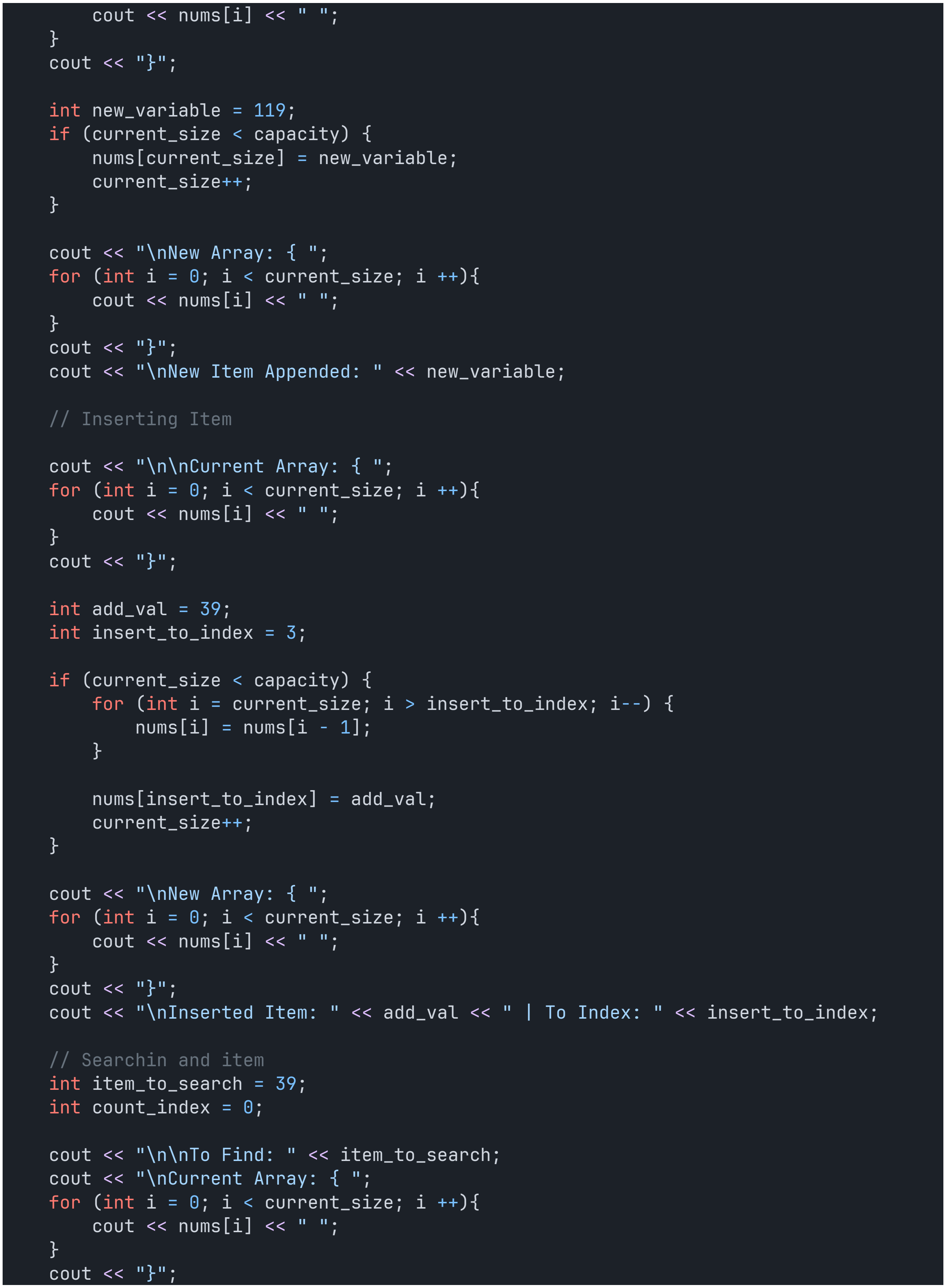
AGUIRRE, PAUL VINCENT S.
M029 - CP103: Data Structures And Algorithm

ACTIVITY 2

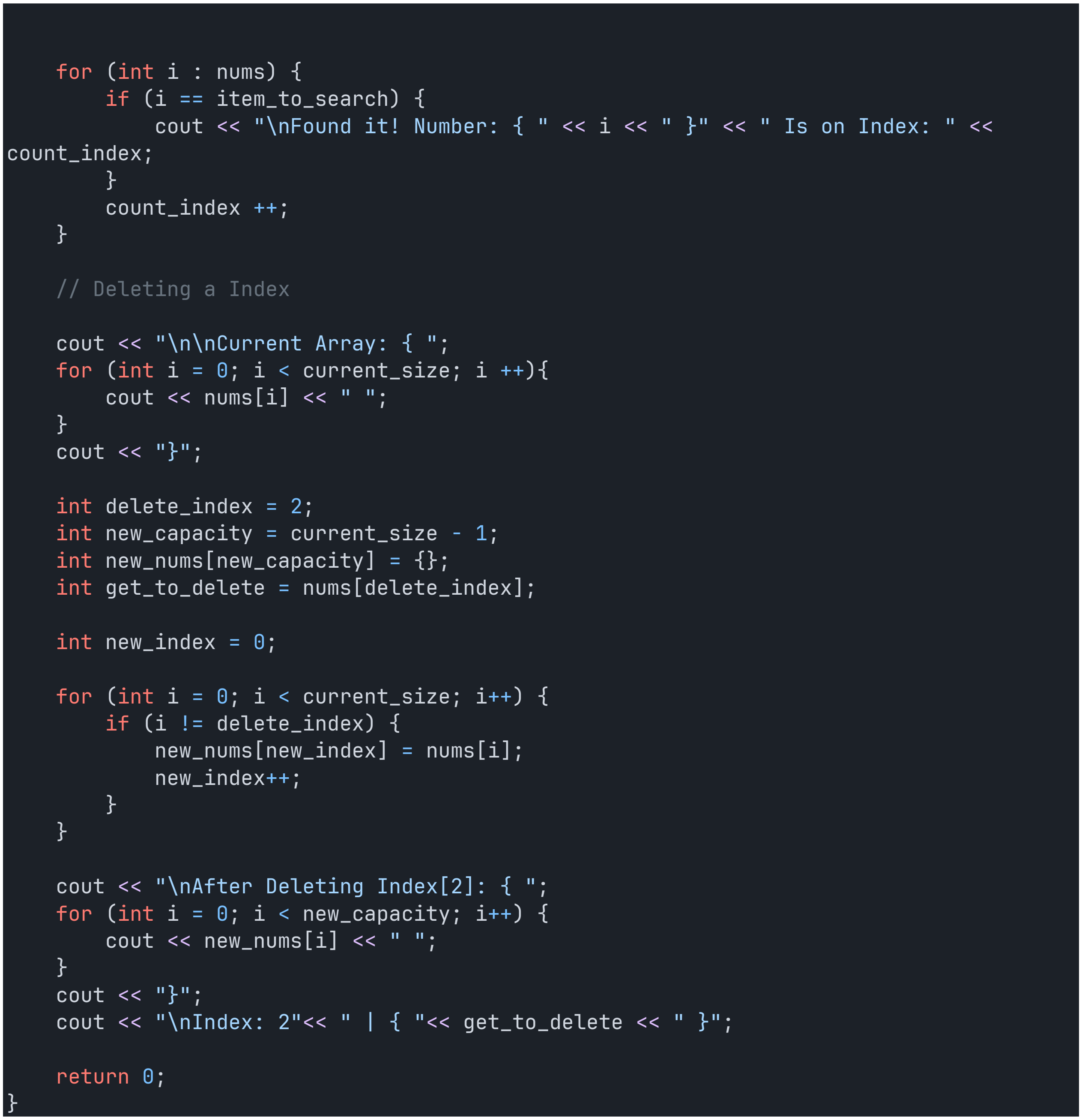
[SOURCE CODE]



```cpp
#include <iostream>
using namespace std;

int main() {

    // Declaring Some Variables
    int capacity = 10;
    int nums[capacity] = {10,20,30,40,50};
    int current_size = 5;

    // Printing Out the "List of Numbers or Arrays"
    cout << "Current Array: { ";
    for (int i = 0; i < current_size; i++) {
        cout << nums[i] << " ";
    }
    cout << "}\n";

    // Get access

    cout << "\nAccessing the Array[1]";
    int get = nums[1];
    cout << "\nArray[1]: " << get;

    // Replacing a number

    cout << "\n\nCurrent Array: { ";

    for (int i = 0; i < current_size; i ++){
        cout << nums[i] << " ";
    }
    cout << "}";

    int old_value = nums[1];
    int replace_int = 298;
    nums[1] = replace_int;

    cout << "\nNew Array: { ";

    for (int i = 0; i < current_size; i++){
        cout << nums[i] << " ";
    }
    cout << "}\n";
    cout << "Old value: " << old_value << " | New value: " << replace_int;

    // Appending new item

    cout << "\n\nCurrent Array: { ";
    for (int i = 0; i < current_size; i ++){
```



```
        cout << nums[i] << " ";
    }
    cout << "}";

    int new_variable = 119;
    if (current_size < capacity) {
        nums[current_size] = new_variable;
        current_size++;
    }

    cout << "\nNew Array: { ";
    for (int i = 0; i < current_size; i ++){
        cout << nums[i] << " ";
    }
    cout << "}";
    cout << "\nNew Item Appended: " << new_variable;

    // Inserting Item

    cout << "\n\nCurrent Array: { ";
    for (int i = 0; i < current_size; i ++){
        cout << nums[i] << " ";
    }
    cout << "}";

    int add_val = 39;
    int insert_to_index = 3;

    if (current_size < capacity) {
        for (int i = current_size; i > insert_to_index; i--) {
            nums[i] = nums[i - 1];
        }

        nums[insert_to_index] = add_val;
        current_size++;
    }

    cout << "\nNew Array: { ";
    for (int i = 0; i < current_size; i ++){
        cout << nums[i] << " ";
    }
    cout << "}";
    cout << "\nInserted Item: " << add_val << " | To Index: " << insert_to_index;

    // Searchin and item
    int item_to_search = 39;
    int count_index = 0;

    cout << "\n\nTo Find: " << item_to_search;
    cout << "\nCurrent Array: { ";
    for (int i = 0; i < current_size; i ++){
        cout << nums[i] << " ";
    }
    cout << "}";
```

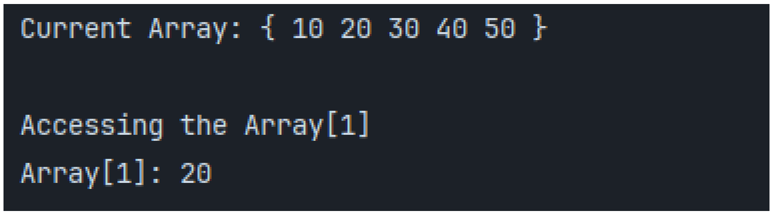


```
    for (int i : nums) {
        if (i == item_to_search) {
            cout << "\nFound it! Number: { " << i << " }" << " Is on Index: " <<
count_index;
        }
        count_index ++;
    }

    // Deleting a Index

    cout << "\n\nCurrent Array: { ";
    for (int i = 0; i < current_size; i ++){
        cout << nums[i] << " ";
    }
    cout << "}";

    int delete_index = 2;
    int new_capacity = current_size - 1;
    int new_nums[new_capacity] = {};
    int get_to_delete = nums[delete_index];

    int new_index = 0;

    for (int i = 0; i < current_size; i++) {
        if (i != delete_index) {
            new_nums[new_index] = nums[i];
            new_index++;
        }
    }

    cout << "\nAfter Deleting Index[2]: { ";
    for (int i = 0; i < new_capacity; i++) {
        cout << new_nums[i] << " ";
    }
    cout << "}";
    cout << "\nIndex: 2"<< " | { "<< get_to_delete << " }";

    return 0;
}
```

AGUIRRE, PAUL VINCENT S.
M029 - CP103: Data Structures And Algorithm

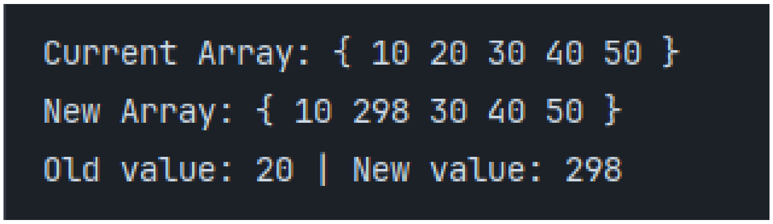
[OUTPUT]

Accessing:



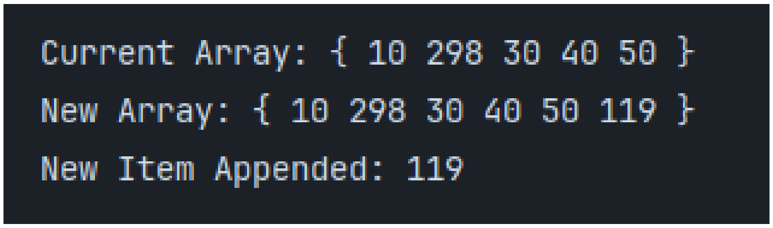
```
Current Array: { 10 20 30 40 50 }


Accessing the Array[1]
Array[1]: 20
```

Replacing:



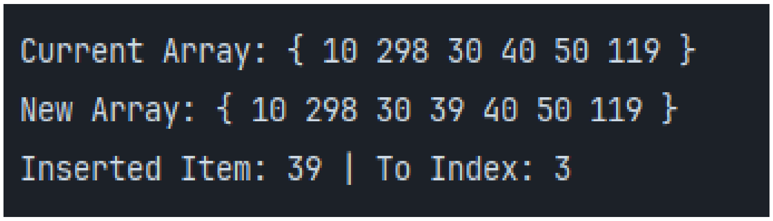
```
Current Array: { 10 20 30 40 50 }
New Array: { 10 298 30 40 50 }
Old value: 20 | New value: 298
```

Adding / Appending:



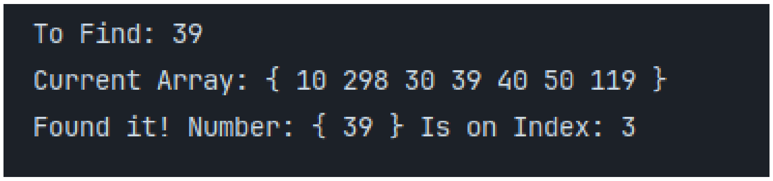
```
Current Array: { 10 298 30 40 50 }
New Array: { 10 298 30 40 50 119 }
New Item Appended: 119
```

Inserting:



```
Current Array: { 10 298 30 40 50 119 }
New Array: { 10 298 30 39 40 50 119 }
Inserted Item: 39 | To Index: 3
```

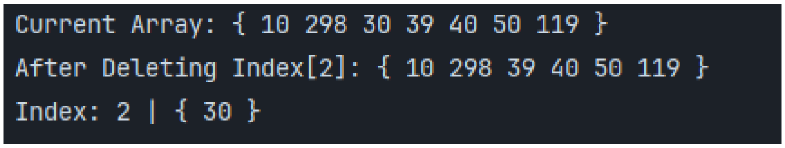
Searching:



```
To Find: 39
Current Array: { 10 298 30 39 40 50 119 }
Found it! Number: { 39 } Is on Index: 3
```

AGUIRRE, PAUL VINCENT S.
M029 - CP103: Data Structures And Algorithm

Deleting:



```
Current Array: { 10 298 30 39 40 50 119 }
After Deleting Index[2]: { 10 298 39 40 50 119 }
Index: 2 | { 30 }
```

[Explanation]

This C++ program works with a static array to show different ways of handling its elements. It starts with an array of five numbers and a set capacity of ten, then prints the current contents. It accesses a specific element using its index, replaces a value with a new one, appends a number at the end if there’s space, and inserts a number at a chosen index by shifting other elements. It also searches for a specific value and displays its index if found. Lastly, it removes an element by creating a new array without the chosen index. Throughout, it manages the current size and capacity manually while demonstrating indexing and shifting elements.