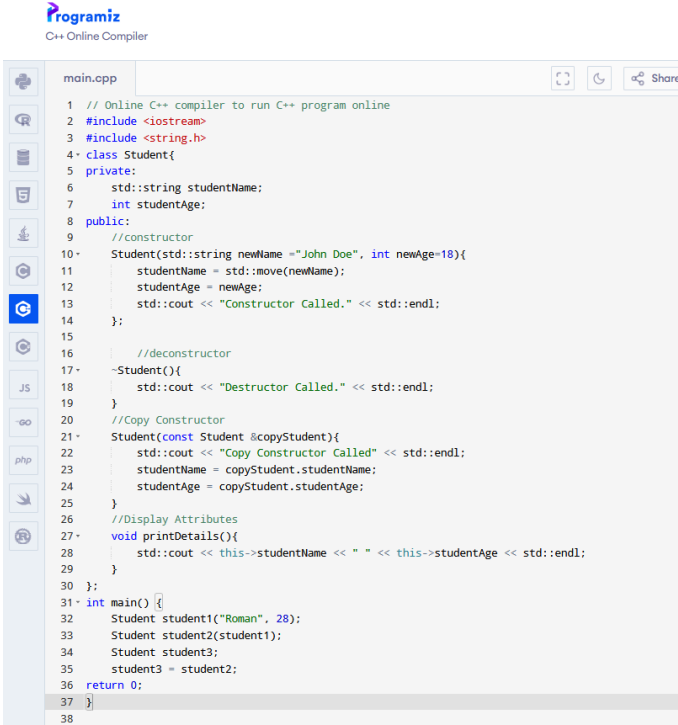


Activity No. 3	
Hands-on Activity 2.1 Arrays, Pointers and Dynamic Memory Allocation	
Course Code: CPE010	Program: Computer Engineering
Course Title: Data Structures and Algorithms	Date Performed: Sep 11, 2024
Section: CPE21S4	Date Submitted: Sep 12, 2024
Name(s): Carl Jervie B. Carag	Instructor: Maria Rizette Sayo
<div> <div>Screenshot</div> <div>  </div> </div> <div> <div>The modified part</div> <div> <pre> 31 int main() { 32 const size_t j = 5; 33 Student studentList[j] = {}; 34 std::string namesList[j] = {"Carly", "Freddy", "Sam", "Zack", "Cody"}; 35 int ageList[j] = {15, 16, 18, 19, 16}; 36 return 0; 37 } 38 </pre> </div> </div>	<div> <div>Output</div> <div> <pre> /tmp/ZESbjKqBNd.o Constructor Called. Copy Constructor Called Constructor Called. Destructor Called. Destructor Called. Destructor Called. === Code Execution Successful === </pre> </div> </div> <div> <div>The modified part</div> <div> <div>Output</div> <div> <pre> /tmp/YkvYqCClnS.o Constructor Called. Constructor Called. Constructor Called. Constructor Called. Constructor Called. Destructor Called. Destructor Called. Destructor Called. Destructor Called. Destructor Called. === Code Execution Successful === </pre> </div> </div> </div>
<div> <div>Observation</div> <div> </div> </div>	<div> <p>I observed that we have first put the needed library header such as iostream and string.h to be the basic foundation of the code. Then we have created our class with private attributes. We have made the body of the code constructor, and we have assigned default values for the parameters of our constructor. The Deconstructor serves as the output stream. Now the driver program is utilized for the main function to show the dynamic and static allocation.</p> </div>

For the Modified Part

I observed that the frequency of the constructor and destructor increased into 5 times before it was modified having the constant size set to 5.

Table 2-1 Initial Driver Program

Screenshot

```
main.cpp
6  std::string studentName;
7  int studentAge;
8  public:
9  //constructor
10 Student(std::string newName = "John Doe", int newAge=18){
11     studentName = std::move(newName);
12     studentAge = newAge;
13     std::cout << "Constructor Called." << std::endl;
14 };
15
16 //destructor
17 ~Student(){
18     std::cout << "Destructor Called." << std::endl;
19 }
20 //Copy Constructor
21 Student(const Student &copyStudent){
22     std::cout << "Copy Constructor Called" << std::endl;
23     studentName = copyStudent.studentName;
24     studentAge = copyStudent.studentAge;
25 }
26 //Display Attributes
27 void printDetails(){
28     std::cout << this->studentName << " " << this->studentAge << std::endl;
29 }
30 };
31 int main() {
32     const size_t j = 5;
33     Student studentList[j] = {};
34     std::string namesList[j] = {"Carly", "Freddy", "Sam", "Zack", "Cody"};
35     int ageList[j] = {15, 16, 18, 19, 16};
36     for(int i = 0; i < j; i++){ //loop A
37         Student *ptr = new Student(namesList[i], ageList[i]);
38         studentList[i] = *ptr;
39     }
40     for(int i = 0; i < j; i++){ //loop B
41         studentList[i].printDetails();
42     }
43     return 0;
44 }
45
```

Output

```
/tmp/R3JKRcMg6g.o
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Carly 15
Freddy 16
Sam 18
Zack 19
Cody 16
Destructor Called.
Destructor Called.
Destructor Called.
Destructor Called.
Destructor Called.
```

Observation

After we have shown the dynamic and static allocation of the values that we have assigned to our parameters. We have now dynamically allocated instances for the student class and we may store newly created objects in the specific allocations pointed by our made arrays.

Table 2-2 Modified Driver Program with St

Loop A

```
for(int i = 0; i < j; i++){ //loop A
Student *ptr = new Student(namesList[i], ageList[i]);
studentList[i] = *ptr;
}
```

Observation

In the loop A, the iteration of the variable name takes place such as the "namesList" and "ageList" while the studentList is set to pointer as a result the corresponding names in the namesList are combined with the variables in the ageList.

Loop B

<pre>for(int i = 0; i < j; i++){ //loop B studentList[i].printDetails(); }</pre>	
Observation	The Loop B then prints the iteration of the code from the loop A, the loop B contains the same iteration with A but it prints it in an iterated method.
Output	
Observation	
6. Output	
7. Supplementary Activity	
8. Conclusion	
<p>This Activity shows how we may utilize the use of static and dynamic memory allocation. I have learned the usage of the operators such as the reference operator which is used to retrieve memory addressed to a variable and the dereference operator which allows accessing a value at a particular memory allocation. Pointers were also used in this activity where it has the address of the memory location containing the data. I have learned the rule of three known as Destructors, Copy Constructor, Copy Assignment Operator. First, the Destructor is a function that mainly deletes an object. The second one is the Copy constructor that makes a copy of an existing instance. Last but not the least was the "Copy Assignment Operator" . We want to copy one object to another using the assignment operator. The driver is responsible for the printing and execution of this code. All of these elements were combined and are identified in this activity.</p>	
9. Assessment Rubric	

"I affirm that I will not give or receive any unauthorized help on this activity/exam and that all work will be my own."