

Activity No. <n>	
<Replace with Title>	
Course Code: CPE010	Program: Computer Engineering
Course Title: Data Structures and Algorithms	Date Performed: 08/06/25
Section: CPE 010-CPE21S4	Date Submitted: 08/06/25
Name(s): Catungal Kerwin Jan B	Instructor: Jimlord Quejado

6. Output

Discussion:

```

1.cpp Discussion.cpp Discussion array.cpp
1  #include <iostream>
2  #include <string>
3
4  int main(){
5      int x = 10;
6      std::cout << x << std::endl;
7      std::cout << &x << std::endl;
8      std::cout << *&x << std::endl;
9
10 }
11

```

```

1.cpp Discussion.cpp Discussion array.cpp
1  #include <iostream>
2  #include <string>
3
4  int main(){
5      int array[] = {1, 2, 3, 4};
6      int *ptrArray
7      //array = ptrArray; //compiler error
8      ptrArray = array; //no errors
9
10 }

```

pp Discussion.cpp Discussion array.cpp Untitled4

```
#include <iostream>
#include <string>

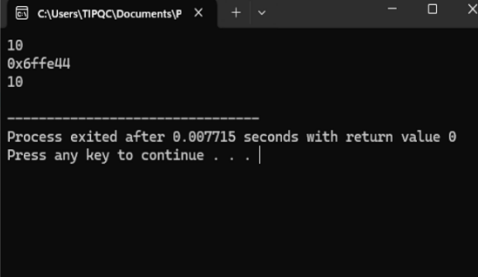
int main(){

    int array[] = {1, 2, 3, 4};
    int *ptrArray;
    //array = ptrArray; //compiler error
    ptrArray = array; //no errors

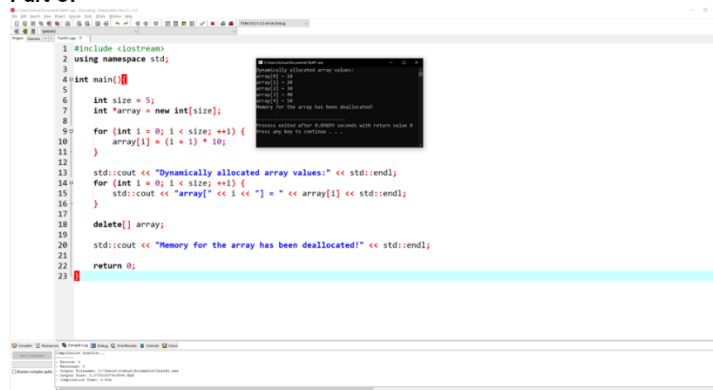
    return 0;
}
```

Part B.

```
test1.cpp Part B.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int var = 10;
6     int *ip;
7     ip = &var;
8
9     cout << var << endl;
10    cout << ip << endl;
11    cout << *ip << endl;
12
13    return 0;
14 }
```

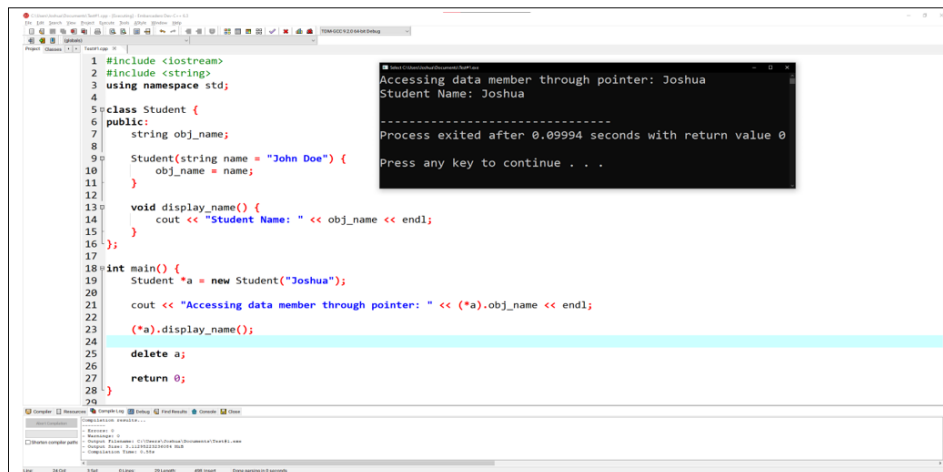


Part C.



```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int size = 5;
7     int *array = new int[size];
8     for (int i = 0; i < size; ++i) {
9         array[i] = (i + 1) * 10;
10    }
11
12    std::cout << "Dynamically allocated array values:" << std::endl;
13    for (int i = 0; i < size; ++i) {
14        std::cout << "array[" << i << "] = " << array[i] << std::endl;
15    }
16
17    delete[] array;
18
19    std::cout << "Memory for the array has been deallocated!" << std::endl;
20
21    return 0;
22 }
```

Part D.

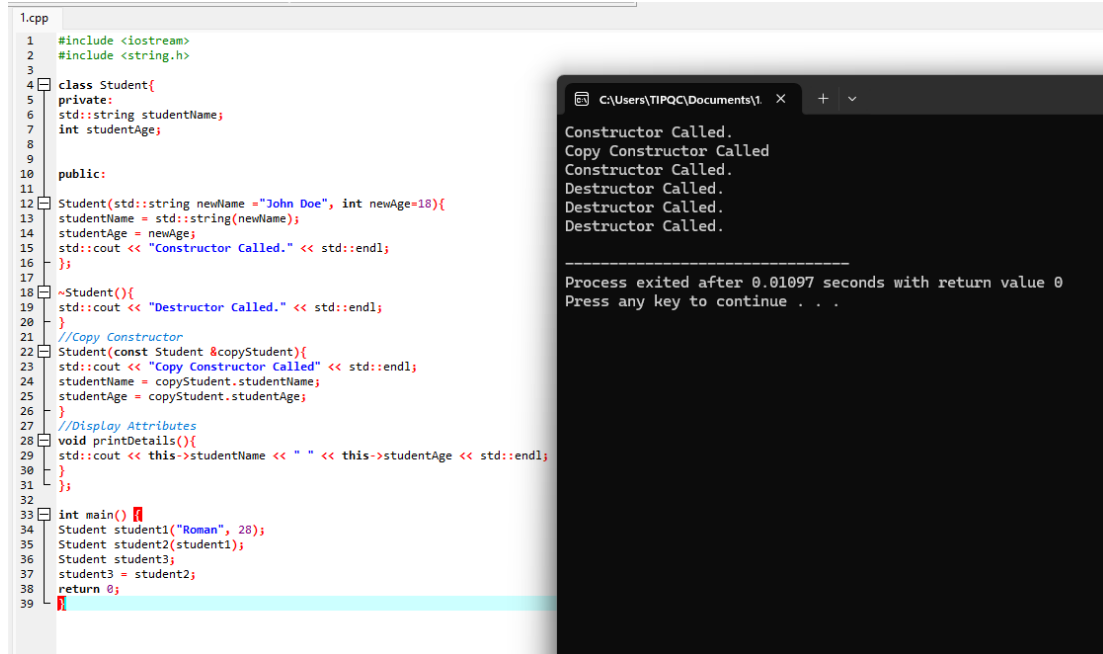


```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 class Student {
6 public:
7     string obj_name;
8
9     Student(string name = "John Doe") {
10         obj_name = name;
11     }
12
13     void display_name() {
14         cout << "Student Name: " << obj_name << endl;
15     }
16 };
17
18 int main() {
19     Student *a = new Student("Joshua");
20
21     cout << "Accessing data member through pointer: " << (*a).obj_name << endl;
22     (*a).display_name();
23
24     delete a;
25     return 0;
26 }
```

Accessing data member through pointer: Joshua
Student Name: Joshua

Process exited after 0.09994 seconds with return value 0
Press any key to continue . . .

Procedure:



```
1.cpp
1 #include <iostream>
2 #include <string.h>
3
4 class Student{
5 private:
6     std::string studentName;
7     int studentAge;
8
9 public:
10
11     Student(std::string newName = "John Doe", int newAge=18){
12         studentName = std::string(newName);
13         studentAge = newAge;
14         std::cout << "Constructor Called." << std::endl;
15     }
16
17     ~Student(){
18         std::cout << "Destructor Called." << std::endl;
19     }
20
21     //Copy Constructor
22     Student(const Student &copyStudent){
23         std::cout << "Copy Constructor Called" << std::endl;
24         studentName = copyStudent.studentName;
25         studentAge = copyStudent.studentAge;
26     }
27
28     //Display Attributes
29     void printDetails(){
30         std::cout << this->studentName << " " << this->studentAge << std::endl;
31     }
32
33     int main() {
34         Student student1("Roman", 28);
35         Student student2(student1);
36         Student student3;
37         student3 = student2;
38         return 0;
39     }
```

C:\Users\TIPQC\Documents\1 X + v
Constructor Called.
Copy Constructor Called
Constructor Called.
Destructor Called.
Destructor Called.
Destructor Called.

Process exited after 0.01097 seconds with return value 0
Press any key to continue . . .

```

cpp
1 #include <iostream>
2 #include <string.h>
3
4 class Student{
5 private:
6     std::string studentName;
7     int studentAge;
8
9 public:
10
11     Student(std::string newName = "John Doe", int newAge=18){
12         studentName = std::string(newName);
13         studentAge = newAge;
14         std::cout << "Constructor Called." << std::endl;
15     };
16
17     ~Student(){
18         std::cout << "Destructor Called." << std::endl;
19     }
20
21     //Copy Constructor
22     Student(const Student &copyStudent){
23         std::cout << "Copy Constructor Called" << std::endl;
24         studentName = copyStudent.studentName;
25         studentAge = copyStudent.studentAge;
26     }
27
28     //Display Attributes
29     void printDetails(){
30         std::cout << this->studentName << " " << this->studentAge << std::endl;
31     }
32
33 int main() {
34     const size_t j = 5;
35     Student studentList[j] = {};
36     std::string namesList[j] = {"Carly", "Freddy", "Sam", "Zack", "Cody"};
37     int ageList[j] = {15, 16, 18, 19, 16};
38     return 0;
39 }

```

```

C:\Users\TIPQC\Documents\1. X + v
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Destructor Called.
Destructor Called.
Destructor Called.
Destructor Called.
Destructor Called.

-----
Process exited after 0.008833 seconds with return value 0
Press any key to continue . . . |

```

```

cpp
1 #include <iostream>
2 #include <string.h>
3
4 class Student{
5 private:
6     std::string studentName;
7     int studentAge;
8
9 public:
10
11     Student(std::string newName = "John Doe", int newAge=18){
12         studentName = std::string(newName);
13         studentAge = newAge;
14         std::cout << "Constructor Called." << std::endl;
15     };
16
17     ~Student(){
18         std::cout << "Destructor Called." << std::endl;
19     }
20
21     //Copy Constructor
22     Student(const Student &copyStudent){
23         std::cout << "Copy Constructor Called" << std::endl;
24         studentName = copyStudent.studentName;
25         studentAge = copyStudent.studentAge;
26     }
27
28     //Display Attributes
29     void printDetails(){
30         std::cout << this->studentName << " " << this->studentAge << std::endl;
31     }
32
33 int main() {
34     const size_t j = 5;
35     Student studentList[j] = {};
36     std::string namesList[j] = {"Carly", "Freddy", "Sam", "Zack", "Cody"};
37     for(int i = 0; i < j; i++){ //Loop A
38         Student *ptr = new Student(namesList[i], ageList[i]); studentList[i]
39     }
40     for(int i = 0; i < j; i++){ //Loop B studentList[i].printDetails();
41     }
42     return 0;
43 }

```

```

C:\Users\TIPQC\Documents\1. X + v
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Constructor Called.
Destructor Called.
Destructor Called.
Destructor Called.
Destructor Called.
Destructor Called.

-----
Process exited after 0.01065 seconds with return value 0
Press any key to continue . . . |

```

7. Supplementary Activity

Data supp.cpp

```
1 #include <iostream>
2 #include <string>
3
4
5 class GroceryItem {
6     protected:
7         std::string name;
8         double price;
9         int quantity;
10
11     public:
12         GroceryItem(std::string n = "", double p = 0.0, int q = 0)
13         {
14             name = n;
15             price = p;
16             quantity = q;
17         }
18
19         virtual ~GroceryItem() {}
20
21         virtual void show()
22         {
23             std::cout << name << " - PHP " << price << " x" << quantity
24             << " = PHP " << price * quantity << std::endl;
25         }
26
27         virtual double getTotal()
28         {
29             return price * quantity;
30         }
31
32         std::string getName()
33         {
34             return name;
35         }
36     };
37
38
39
40 class Fruit : public GroceryItem
```

C:\Users\Olaco\Downloads\Data supp.exe

Grocery List:
[Fruit] Apple - PHP 10 x7 = PHP 70
[Fruit] Banana - PHP 10 x8 = PHP 80
[Vegetable] Broccoli - PHP 60 x12 = PHP 720
[Vegetable] Lettuce - PHP 50 x10 = PHP 500

Total Price: PHP 1370
Lettuce removed.

Updated Grocery List:
[Fruit] Apple - PHP 10 x7 = PHP 70
[Fruit] Banana - PHP 10 x8 = PHP 80
[Vegetable] Broccoli - PHP 60 x12 = PHP 720

Process exited after 0.257 seconds with return value 0
Press any key to continue . . .

rces

Compile Log

Debug

Find Results

Console

Close

Compilation results...

8. Conclusion

So in this exercise, I was able to delve further into how arrays, pointers, and memory function in C++ in general, the entire idea of using new and delete to allocate dynamic memory. Working through it really gave me a sense of how the program was operating, and doing additional things like deleting an object or printing out the total price made it all the more hands-on and applicable. The entire process really showed me the step-by-step reasoning of programming, and that extra material like Jenna's Grocery List made it come so much more alive and interactive. I think I did fairly well although it's a bit hard given my situation but I'll definitely need to practice tracing the code better and keeping my memory management nice and tidy. This exercise seriously taught me how data is stored and manipulated in more complex programs.

9. Assessment Rubric