

Activity No. 2.1

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: 8/7/25
Section: CPE21S4	Date Submitted: 8/7/25
Name(s): Punay Heidee S.	Instructor: Engr. Jimlord Quejado

6. Output

Problem 1

Output

```
Vegetable Classes
Vegetables: Carrots Price: 3 Quantity: 10
Total price of Carrots is: 30
Vegetables: Broccoli Price: 5 Quantity: 10
Total price of Broccoli is: 50
Vegetables: Squash Price: 2 Quantity: 5
Total price of Squash is: 10
```

Output

```
Fruit Classes
Fruits: Banana Price: 3 Quantity: 10
Total price of Banana is: 30
Fruits: Manggo Price: 5 Quantity: 10
Total price of Manggo is: 50
Fruits: Guava Price: 2 Quantity: 5
Total price of Guava is: 10
```

Problem 2

Output

```
Item: Coffee
Quantity: 10
Price per unit: Php2.99
Total cost: Php29.9
Item: Bread
Quantity: 10
Price per unit: Php3.5
Total cost: Php35
Item: Milk
Quantity: 10
Price per unit: Php4.5
Total cost: Php45
```

```
==== Code Execution Successful ====
```

7. Supplementary Activity

```
main.cpp
1 #include <iostream>
2 #include <string>
3
4 class Vegetables{
5     public:
6
7
8     //Attributes
9     std::string name;
10    int price;
11    int quantity;
12
13    // Constructor
14    Vegetables(std::string newName, int newPrice, int newQuantity){
15        name = newName;
16        price = newPrice;
17        quantity = newQuantity;
18    }
19
20
21    // Destructor
22    ~Vegetables() {}
23
24
25    }
26    // Copy Constructor
27    Vegetables(const Vegetables &copyVegetables){
28        Vegetables::name = copyVegetables.name;
29        Vegetables::price = copyVegetables.price;
30        Vegetables::quantity = copyVegetables.quantity;
31    }
32
33
34    // Copy Assignment Operator
35    Vegetables &operator=(const Vegetables &copyVegetables){
36        if (this != &copyVegetables){
37            Vegetables::name = copyVegetables.name;
38            Vegetables::price = copyVegetables.price;
39            Vegetables::quantity = copyVegetables.quantity;
40        }
41    }
42
```

```
37     Vegetables::name = copyVegetables.name;
38     Vegetables::price = copyVegetables.price;
39     Vegetables::quantity = copyVegetables.quantity;
40 }
41 return *this;
42 }
43
44 void calculateSum() const {
45     std::cout << "Total price of " << name << " is: " << price * quantity << std::endl;
46 }
47 }
48 };
49
50
51 int main(){
52
53
54     std::cout << "Vegetable Classes" << std::endl;
55
56
57
58
59
60
61     Vegetables Carrots("Carrots", 3, 10);
62     Vegetables Broccoli("Broccoli", 5, 10);
63     Vegetables Squash("Squash", 2, 5);
64
65     std::cout << " Vegetables: " << Carrots.name << " Price: " << Carrots.price << " Quantity: " << Carrots
66         .quantity << std::endl;
67     Carrots.calculateSum();
68     std::cout << " Vegetables: " << Broccoli.name << " Price: " << Broccoli.price << " Quantity: " << Broccoli
69         .quantity << std::endl;
70     Broccoli.calculateSum();
71     std::cout << " Vegetables: " << Squash.name << " Price: " << Squash.price << " Quantity: " << Squash.quantity
72         << std::endl;
73     Squash.calculateSum();
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100 }
```

```
1 #include <iostream>
2
3 class GroceryItem {
4 private:
5     std::string name;
6     int quantity;
7     double price;
8
9 public:
10
11    GroceryItem(std::string itemName, int itemQuantity, double itemPrice) {
12        name = itemName;
13        quantity = itemQuantity;
14        price = itemPrice;
15    }
16
17
18    void displayDetails() const {
19        std::cout << "Item: " << name << std::endl;
20        std::cout << "Quantity: " << quantity << std::endl;
21        std::cout << "Price per unit: Php" << price << std::endl;
22        std::cout << "Total cost: Php" << quantity * price << std::endl;
23    }
24 }
25 };
26
27 int main() {
28     // Create an array of GroceryItem objects
29     GroceryItem GroceryList[3] = {
30         GroceryItem("Coffee", 10, 2.99),
31         GroceryItem("Bread", 10, 3.50),
32         GroceryItem("Milk", 10, 4.50)
33     };
34
35     // Display all item details
36     for (int i = 0; i < 3; ++i) {
37         GroceryList[i].displayDetails();
38     }
39
40     return 0;

```

main.cpp



```
1 #include <iostream>
2 #include <string>
3
4
5+ class Fruits {
6     public:
7
8
9     //Attributes
10    std::string name;
11    int price;
12    int quantity;
13
14    // Constructor
15+   Fruits(std::string newName, int newPrice, int newQuantity){
16        name = newName;
17        price = newPrice;
18        quantity = newQuantity;
19    }
20
21
22    // Destructor
23+   ~Fruits() {
24
25
26    }
27    // Copy Constructor
28+   Fruits(const Fruits &copyFruits){
29        Fruits::name = copyFruits.name;
30        Fruits::price = copyFruits.price;
31        Fruits::quantity = copyFruits.quantity;
32    }
33
34
35    // Copy Assignment Operator
36+   Fruits &operator=(const Fruits &copyFruits){
37        if (this != &copyFruits){
38            Fruits::name = copyFruits.name;
39            Fruits::price = copyFruits.price;
40            Fruits::quantity = copyFruits.quantity;
41        }
42    }
43
44
```

```
main.cpp
12     return 0;
13 }
14
15     void calculateSum() const {
16         std::cout << "Total price of " << name << " is: " << price * quantity << std::endl;
17
18     }
19 };
20
21
22 int main(){
23
24
25     std::cout << "Fruit Classes" << std::endl;
26
27
28
29
30
31
32 Fruits Banana("Banana", 3, 10);
33 std::cout << "Fruits: " << Banana.name << " Price: " << Banana.price << " Quantity: " << Banana.quantity
34     << std::endl;
35     Banana.calculateSum();
36
37
38
39 Fruits Mango("Mango", 5, 10)
40 std::cout << "Fruits: " << Mango.name << " Price: " << Mango.price << " Quantity: " << Mango.quantity << std
41     ::endl;
42     Mango.calculateSum();
43
44
45 Fruits Guava("Guava", 2, 5);
46 std::cout << "Fruits: " << Guava.name << " Price: " << Guava.price << " Quantity: " << Guava.quantity << std
47     ::endl;
48     Guava.calculateSum();
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79     return 0;
```

8. Conclusion:

To conclude, we were able to do some of the tasks. It was confusing and hard to it so what we did was watch some videos on how to do it and ask some help to my friends. The task were making a class together with the object and methods and how will we apply it in C++. Overall, it was hard to do it and I hope we get to learn more of this to help myself in making a code or to ourselves.

9. Assessment Rubric