

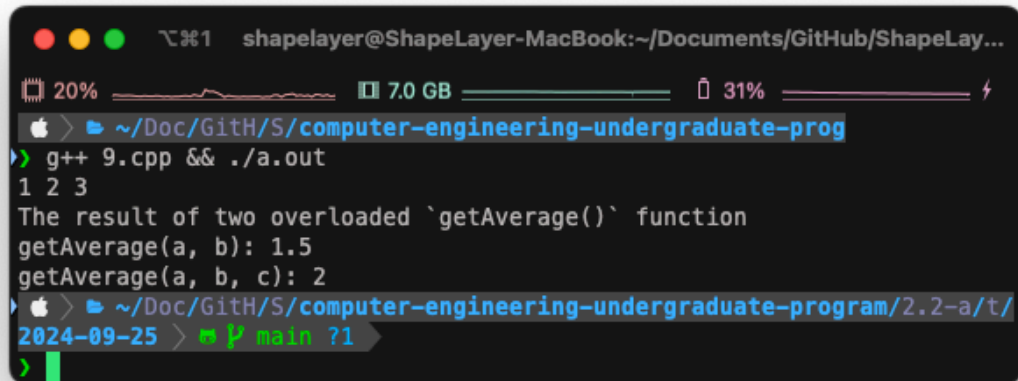
고급프로그래밍및실습 과제 #2

214823 박종현

#9 평균을 구하는 멤버 함수 `getAverage()`를 중복 정의해보자.

```
1  #include <iostream>
2
3  using namespace std;
4
5  struct Calculator {
6  public:
7      float getAverage(int a, int b) {
8          /**
9           * Calculate average of numbers.
10          *
11          *
12          * @param a, b Two integers to be averaged.
13          * @return average of `values`.
14          */
15          return ((float)(a + b)) / 2;
16      }
17
18      float getAverage(int a, int b, int c) {
19          /**
20           * Calculate average of numbers.
21          *
22          *
23          * @param a, b, c Three integers to be averaged.
24          * @return average of `values`.
25          */
26          return ((float)(a + b + c)) / 3;
27      }
28 };
29
30 int main() {
31     /**
32      * Input expected: 3 Integers seperated by whitespace.
33      * example:
34      * 7 8 10
35      */
36     Calculator calc = Calculator();
37     int a, b, c;
38
39     cin >> a >> b >> c;
40     float first = calc.getAverage(a, b);
41     float second = calc.getAverage(a, b, c);
42     cout << "The result of two overloaded `getAverage()` function" << endl;
43     cout << "getAverage(a, b): " << first << endl;
44     cout << "getAverage(a, b, c): " << second << endl;
```

```
45
46     return 0;
47 }
```



A terminal window on a MacBook. The title bar shows the user 'shapelayer' and the path '~/Documents/GitHub/ShapeLay...'. The terminal displays the following content:

```
Apple > ~/Doc/GitH/S/computer-engineering-undergraduate-prog
> g++ 9.cpp && ./a.out
1 2 3
The result of two overloaded `getAverage()` function
getAverage(a, b): 1.5
getAverage(a, b, c): 2
Apple > ~/Doc/GitH/S/computer-engineering-undergraduate-program/2.2-a/t/
2024-09-25 > main ?1
>
```

The terminal also shows system status at the top: 20% battery, 7.0 GB memory, and 31% disk space.

#10 MyArray라는 이름의 클래스를 작성하여 보자 ...

C++

```
1  #include <iostream>
2
3  using namespace std;
4
5  struct ArrayNode;
6  struct MyArray;
7
8  struct ArrayNode {
9      /**
10       * `ArrayNode` is data container for `MyArray` class to implement
11       * linkedlist-like array with dynamic allocation.
12       */
13  private:
14      /**
15       * `_p` indicates `MyArray` that uses this as data container.
16       * `_p` is declared because it may be useful later, but this
17       */
18      MyArray *_p;
19  public:
20      int value;
21      ArrayNode *next = NULL;
22      ArrayNode *prev = NULL;
23      ArrayNode(MyArray *p, ArrayNode *prev, int e) {
24          this->_p = p;
25          this->prev = prev;
26          this->value = e;
27      }
28      void setNext(ArrayNode *node) {
29          this->next = node;
30      }
31  };
32
33  struct MyArray {
34  private:
35      // 멤버 변수로는 배열의 크기를 나타내는 size와
36      int size = 0;
37
38      // 정수들이 실제로 저장된 메모리를 가리키는 `ptr`을 가진다.
39      ArrayNode *ptr = NULL; // Head
40      ArrayNode *ptr_tail = NULL; // Tail
41
42  public:
43      // `size` is encapsulized: This `getSize()` is getter.
44      int get_size() { return this->size; }
45
46      // 멤버 함수로는 정수를 추가하는 append()
47      void append(int e) {
48          ArrayNode *last = this->ptr_tail;
49          // 새 노드 생성
50          ArrayNode *_new = new ArrayNode(this, last, e);
51          // 마지막 노드가 null이라면 빈 리스트인 것
52          if (last == NULL)
53              this->ptr = _new; // 이 경우 시작 노드를 새로 생성한 노드로 지정
54          else
```

```

55     last->next = _new; // 빈 리스트가 아니라면 마지막 노드의 다음 노드를 새로 생성한 노드로 지정
56     // 마지막 노드 수정
57     this->ptr_tail = _new;
58     // 배열 크기 증가
59     this->size++;
60 }
61
62 // 마지막 정수를 삭제하는 delete()
63 // delete() is reserved.
64 void delete_last() {
65     if (size == 0) return; // 길이가 0이면 처리 종료
66
67     // 마지막 노드를 가져와서
68     // 마지막 노드의 이전 노드를 현재 배열의 마지막으로 지정
69     ArrayNode *last = this->ptr_tail;
70     this->ptr_tail = last->prev;
71     // 마지막 노드 제거
72     delete last;
73     // 길이 1 감소
74     this->size--;
75 }
76
77 // 배열 안의 정수를 출력하는 print() 등을 가진다.
78 void print() {
79     cout << "[" << this->size << "]" ";
80
81     // 배열 시작점을 데이터를 가져올 대상으로 지정
82     ArrayNode *ptr = this->ptr;
83     // 더 이상 가져올 대상이 없을 때까지 처리 반복
84     while (ptr != NULL) {
85         // 데이터를 가져올 대상의 값 출력
86         cout << ptr->value << " ";
87         // 데이터를 가져올 대상, 다음으로 지정
88         ptr = ptr->next;
89     }
90     cout << endl;
91 }
92 };
93
94 int main() {
95     MyArray arr = MyArray();
96     arr.append(1);
97     arr.append(2);
98     arr.append(3);
99     arr.print();
100    arr.delete_last();
101    arr.print();
102    return 0;
103 }

```

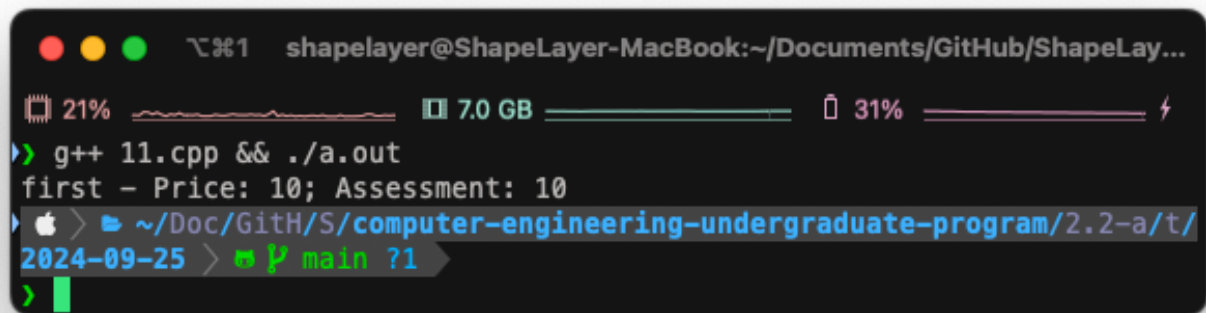
```
shapelayer@ShapeLayer-MacBook:~/Documents/GitHub/ShapeLay...
24% 6.9 GB 31%
> g++ 10.cpp -w && ./a.out
[3] 1 2 3
[2] 1 2 2043
> ~/Doc/GitH/S/computer-engineering-undergraduate-program/2.2-a/t/
2024-09-25 > main ?1
>
```

#11 Product라는 이름의 클래스를 작성하여 보자 ...

C++

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  struct Product {
7  public:
8      // 멤버 변수로는 제품의 이름을 나타내는 `name`,
9      string name;
10
11     // 제품의 가격을 나타내는 `price`,
12     int price;
13
14     // 제품에 대한 평가를 나타내는 `assessment`를 가진다.
15     int assessment;
16
17     /**
18      * Product 생성자: name, price, assessment를 매개변수로 받는다.
19      *
20      */
21     Product(string name, int price, int assessment) {
22         this->name = name;
23         this->price = price;
24         this->assessment = assessment;
25     }
26
27     // 멤버 함수로는 제품에 대한 정보들을 읽는 `getInfo()`,
28     string getInfo() {
29         return name + " - Price: " + to_string(price) + "; Assessment: " + to_string(assessment);
30     }
31
32     // 두 개의 제품을 비교하는 `bool isBetter(Product another)`,
33     /**
34      * `assessment` 속성을 기준으로 비교, `assessment`가 크면 더 나은 것으로 판단
35      */
36     bool isBetter(Product *other) {
37         return this->assessment > other->assessment;
38     }
39
40     // 제품에 대한 정보를 출력하는 `print()`를 가진다.
41     /**
42      * getInfo 메서드를 호출해 받은 정보를 그대로 출력 스트림으로 전달
43      */
44     void print() {
45         cout << this->getInfo() << endl;
46     }
47 };
48
49 int main() {
50     // Product 객체를 생성하여 테스트하여라
51     /**
52      * 테스트를 위해 가격 10, 평가 10의 first 상품과 가격 100, 평가 -1의 second 상품 생성
53      * first는 second보다 평가가 높음이 보장되므로(10 > -1 == true)
54      * first의 정보를 항상 출력함.
```

```
55  */
56  Product first = Product("first", 10, 10), second = Product("second", 100, -1);
57  if (first.isBetter(&second)) {
58      first.print();
59  } else {
60      return 1; // 만약 이 코드가 실행되면 문제가 있는 것.
61  }
62  return 0;
63 }
```



A terminal window on a MacBook. The title bar shows the name 'shapelayer@ShapeLayer-MacBook' and the path '~/Documents/GitHub/ShapeLay...'. The terminal displays the output of a C++ compilation and execution. The command 'g++ 11.cpp && ./a.out' is entered, followed by the output 'first - Price: 10; Assessment: 10'. The prompt changes to a shell prompt '2024-09-25 >'. The user enters 'main ?1' and the prompt changes to a green prompt '>'. The terminal also shows system status at the top: 21% battery, 7.0 GB memory, and 31% battery.

```
shapelayer@ShapeLayer-MacBook:~/Documents/GitHub/ShapeLay...
21% 7.0 GB 31%
> g++ 11.cpp && ./a.out
first - Price: 10; Assessment: 10
> main ?1
>
```