

Assignment I

- 문제는 총 3문제
- 각 문항에서 발생하는 파일(폴더)은 저장하여 압축한 후 LMS에 정해진 시간내에 upload.
- 문장의 comment로 설명, 지정된 파일명 작성
- (Assignment I-1) 학번_이름_Coffee.zip
 - 만약 upgrade문제를 해결할 경우 학번_이름 _Coffee_up.zip으로 제출함 (둘 중 한가지를 선택하도록)
- (Assignment I-2) 학번_이름_Real.zip
- (Assignment I-3) 학번_이름_Com.zip

Assignment 1-1

- CoffeeMachine 클래스를 만들기
- 커피 제조 비율

에스프레소 : 커피와 물이 각각 1씩 소비

아메리카노 : 커피는 1, 물은 2가 소비

설탕커피 : 커피 1, 물 2, 설탕 1

- CoffeeMachine 클래스의 멤버 변수와 멤버 함수를 구성하여 .h 파일로 작성(CoffeeMachine.h)
- CoffeeMachine 클래스를 구현한 부분을 따로 작성하기 (CoffeeMachine.cpp)

Assignment 1-1

CoffeeMachine.cpp

CoffeeMachine.h

main.cpp

report1

(전역 범위)

```
1  #include "CoffeeMachine.h"
2
3  int main() {
4      CoffeeMachine java(5, 10, 6); // 커피량:5, 물량:10, 설탕:6으로 초기화
5      java.drinkEspresso(); // 커피 1, 물 1 소비
6      java.show(); // 현재 커피 머신의 상태 출력
7      java.drinkAmericano(); // 커피 1, 물 2 소비
8      java.show(); // 현재 커피 머신의 상태 출력
9      java.drinkSugarCoffee(); // 커피 1, 물 2, 설탕 1 소비
10     java.show(); // 현재 커피 머신의 상태 출력
11     java.fill(); // 커피 10, 물 10, 설탕 10으로 채우기
12     java.show(); // 현재 커피 머신의 상태 출력
13 }
```

선택 Microsoft Visual Studio 디버그 콘솔

```
Espresso
커피:4 물:9 설탕:6

Americano
커피:3 물:7 설탕:6

SugarCoffee
커피:2 물:5 설탕:5

FILL UP!!
커피:10 물:10 설탕:10
```

Assignment I-I (Upgrade Extra Prob.)

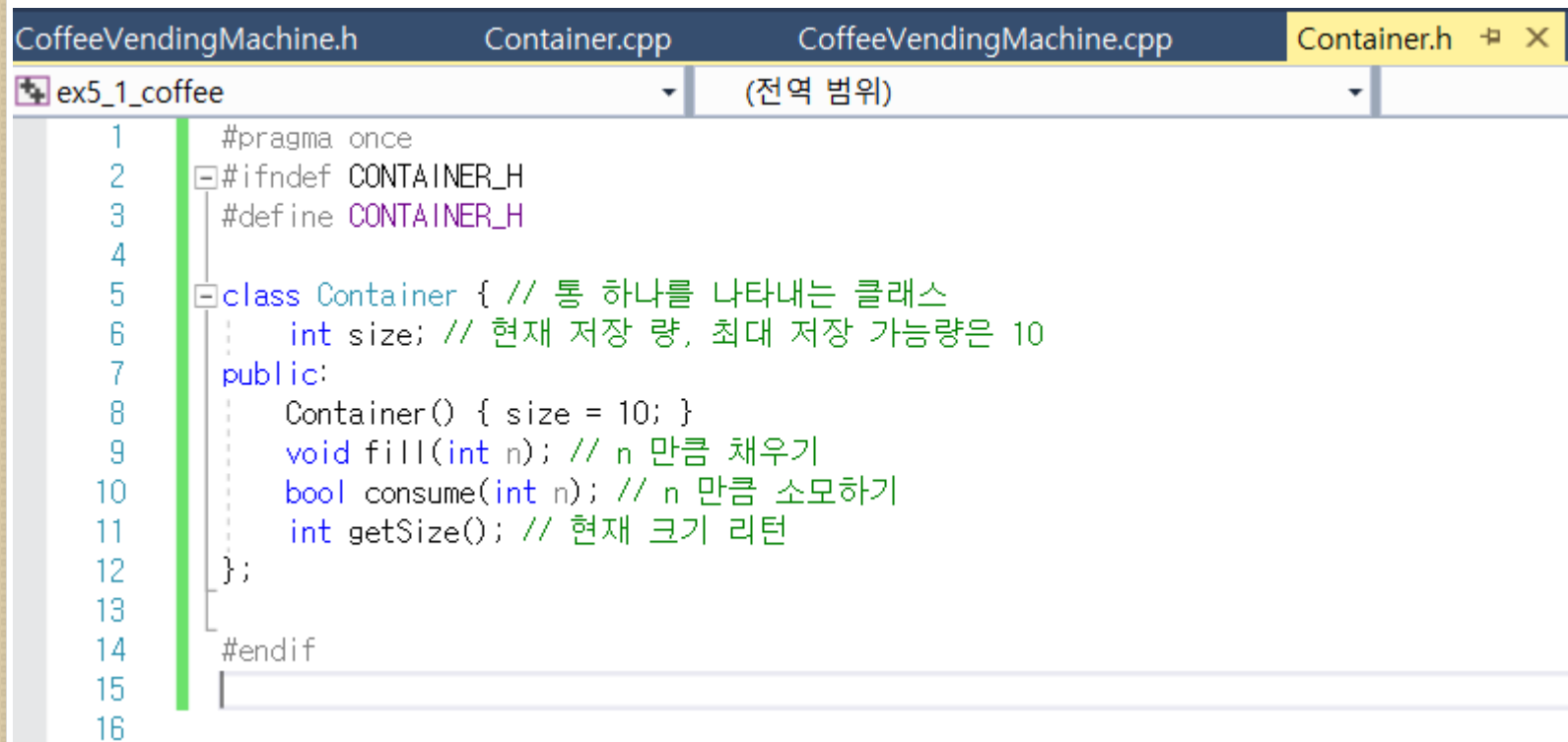
- 커피 자판기로 작동하는 프로그램
- CoffeeVendingMachine.h //커피 자판기를 표현하는 클래스
- 커피와 물, 설탕을 tong이라는 객체배열로 표현

```
CoffeeVendingMachine.h  Container.cpp  CoffeeVendingMachine.cpp  Container.h  mai
ex5_1_coffee  CoffeeVendingMachine  checkInputErrc

1  #pragma once
2  #ifndef COFFEEVENDINGMACHINE_H
3  #define COFFEEVENDINGMACHINE_H
4
5  #include "Container.h"
6
7  class CoffeeVendingMachine { // 커피 자판기를 표현하는 클래스
8      Container tong[3]; // tong[0]는 커피, tong[1]은 물, tong[2]는 설탕통을 나타냄
9      void fill(); // 3개의 통을 모두 10으로 채움
10     void selectEspresso(); // 에스프레소를 선택한 경우, 커피 1, 물 1 소모
11     void selectAmericano(); // 아메리카노를 선택한 경우, 커피 1, 물 2 소모
12     void selectSugarCoffee(); // 설탕커피를 선택한 경우, 커피 1, 물 2 소모, 설탕 1 소모
13     void show(); // 현재 커피, 물, 설탕의 잔량 출력
14     bool checkInputError(); // 오류 처리를 위해 추가한 멤버 함수
15 public:
16     void run(); // 커피 자판기 작동
17 };
18
19 #endif
20
```

Assignment I-I (Upgrade Extra Prob.)

- Container.h // 커피 자판기 내의 물품을 담는 통 하나를 나타내는 클래스



```
1  #pragma once
2  #ifndef CONTAINER_H
3  #define CONTAINER_H
4
5  class Container { // 통 하나를 나타내는 클래스
6      int size; // 현재 저장 량, 최대 저장 가능량은 10
7  public:
8      Container() { size = 10; }
9      void fill(int n); // n 만큼 채우기
10     bool consume(int n); // n 만큼 소모하기
11     int getSize(); // 현재 크기 리턴
12 };
13
14 #endif
15
16
```

Assignment I-I (Upgrade Extra Prob.)

- 다음과 같이 실행되도록 Container.cpp, coffeeVendingMachine.cpp파일을 구하기
- 잔량을 확인하고 만일 커피, 물, 설탕 중 하나라도 부족하면 '원료가 부족합니다'가 출력됨
- 실행을 계속할 경우 1, 아니면 0으로 중간에 중지할 수 있도록 함

```
main.cpp x CoffeeVendingMachine.h Cont
ex5_1_coffee (전'
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 #include "CoffeeVendingMachine.h"
6 #include "Container.h"
7
8 int main() {
9     CoffeeVendingMachine cvm;
10    cvm.run();
11 }
```

```
Microsoft Visual Studio 디버그 콘솔
***** 커피자판기를 작동합니다. *****
메뉴를 눌러주세요(1:에스프레소, 2:아메리카노, 3:설탕커피, 4:잔량보기, 5:채우기)>> 4
커피 10, 물 10, 설탕 10
계속하실려면 1, 마칠시려면 0을 누르세요 : 1
메뉴를 눌러주세요(1:에스프레소, 2:아메리카노, 3:설탕커피, 4:잔량보기, 5:채우기)>> 3
설탕커피 드세요
계속하실려면 1, 마칠시려면 0을 누르세요 : 1
메뉴를 눌러주세요(1:에스프레소, 2:아메리카노, 3:설탕커피, 4:잔량보기, 5:채우기)>> 2
아메리카노 드세요
계속하실려면 1, 마칠시려면 0을 누르세요 : 1
메뉴를 눌러주세요(1:에스프레소, 2:아메리카노, 3:설탕커피, 4:잔량보기, 5:채우기)>> 4
커피 8, 물 6, 설탕 9
계속하실려면 1, 마칠시려면 0을 누르세요 : 1
메뉴를 눌러주세요(1:에스프레소, 2:아메리카노, 3:설탕커피, 4:잔량보기, 5:채우기)>> 1
에스프레소 드세요
```

Assignment 1-2

- 실수의 지수표현을 클래스 Exp로 작성하기
- 클래스 구현부분을 Exp.cpp로 구성하고, 클래스 생성부분의 표현을 Exp.h 파일로 분리하여 작성하기

```
main.cpp  X  Exp.cpp  Exp.h
ex4_2_test1  (전역 범위)

1  #include <iostream>
2  #include "Exp.h"
3  using namespace std;
4
5  int main() {
6      Exp a(3, 2);
7      Exp b(9);
8      Exp c;
9
10     cout << a.getValue() << ' ' << b.getValue() << ' ' << c.getValue() << endl;
11     cout << "a의 베이스 " << a.getBase() << ',' << "지수 " << a.getExp() << endl;
12
13     if (a.equals(b))
14         cout << "same" << endl;
15     else
16         cout << "not same" << endl;
17 }
```

Microsoft Visual Studio 디버그 콘솔

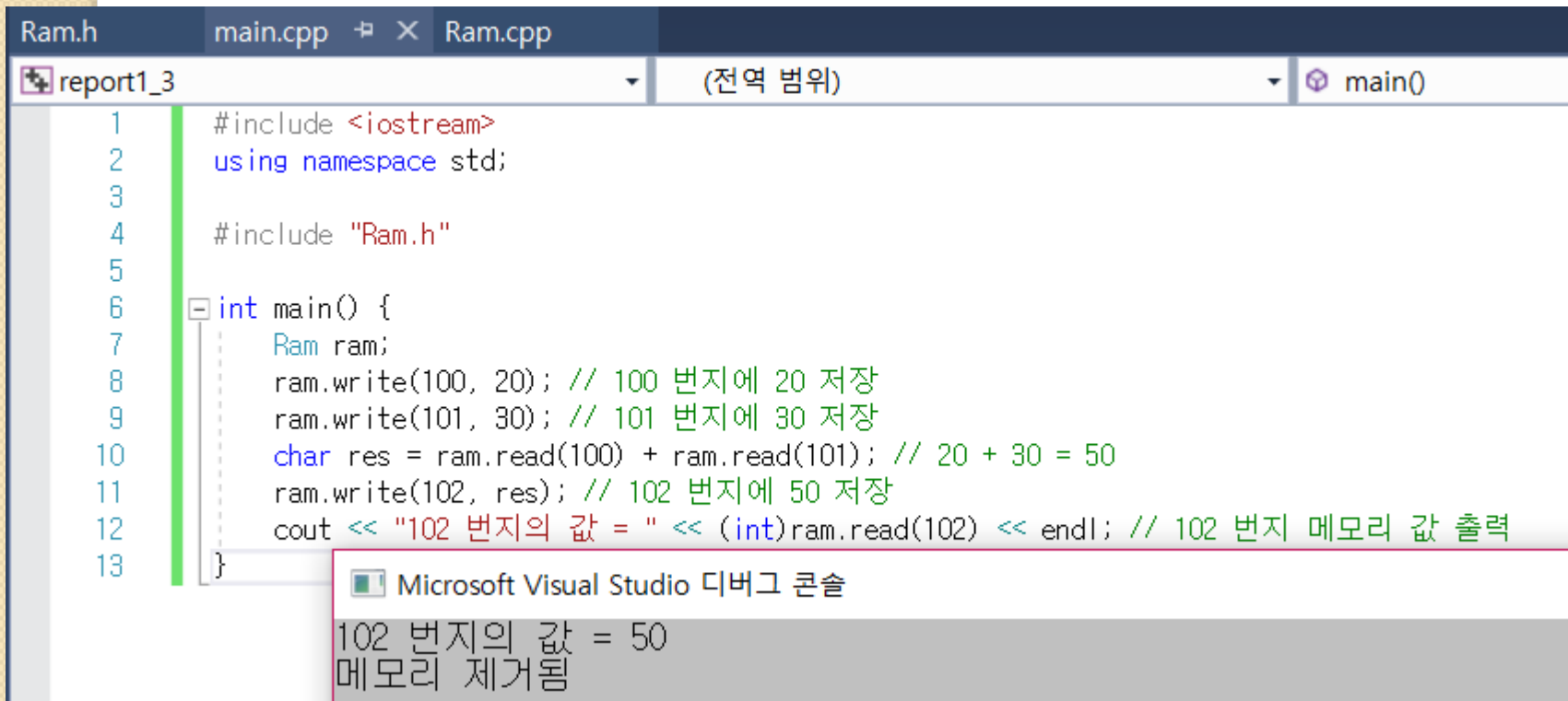
```
9 9 1
a의 베이스 3,지수 2
same
```

Assignment 1-2

- Exp 클래스는 3개의 생성자와 2개의 멤버 변수, 4개의 멤버 함수가 필요함
- `int getValue()` // 지수를 정수로 계산하여 리턴
- `int getBase()` // 베이스 값 리턴
- `int getExp()` // 지수값 리턴
- `int equals(Exp b)` // 이 객체와 객체 b의 값이 같은지 판별하여 리턴

Assignment 1-3

- 컴퓨터의 주 기억장치를 모델링하는 클래스 RAM 구현
- 데이터가 기록될 메모리 공간과 크기 정보를 가지고 주어진 주소에 데이터를 기록(write), 주어진 주소로부터 데이터를 읽어옴(read)
- Ram.h, Ram.cpp, main.cpp로 나누어서 저장하기



The screenshot shows a C++ IDE with three tabs: Ram.h, main.cpp, and Ram.cpp. The main.cpp tab is active, displaying the following code:

```
1  #include <iostream>
2  using namespace std;
3
4  #include "Ram.h"
5
6  int main() {
7      Ram ram;
8      ram.write(100, 20); // 100 번지에 20 저장
9      ram.write(101, 30); // 101 번지에 30 저장
10     char res = ram.read(100) + ram.read(101); // 20 + 30 = 50
11     ram.write(102, res); // 102 번지에 50 저장
12     cout << "102 번지의 값 = " << (int)ram.read(102) << endl; // 102 번지 메모리 값 출력
13 }
```

Below the code editor, the Microsoft Visual Studio 디버그 콘솔 (Debug Console) is visible, showing the output of the program:

```
102 번지의 값 = 50
메모리 제거됨
```

Assignment 1-3

- Ram클래스는 1개의 생성자와 1개의 소멸자, 2개의 멤버변수, 2개의 멤버 함수가 필요
- `char mem[100*1024]` // 100KB 메모리
- `char read(int addr)`// addr 주소의 메모리를 읽어 리턴
- `void write(int addr, char value)` // addr 주소에 value 저장
- 메모리 사용이 끝나면 메모리 제거 후 제거되었음을 알리는 메시지 출력