

HW1 강의 중 실습 코드를 loop 사용하여 변경

```
int main()
```

```
{
```

```
    int coffee;
```

```
    cout<<"어떤 커피 드릴까요? (1:보통, 2:설탕, 3:블랙) ";
```

```
    cin>>coffee;
```

```
    coffee_machine(coffee);
```

```
    cout<<"손님 커피 여기 있습니다. "<<endl;
```

```
    cout<<"어떤 커피 드릴까요? (1:보통, 2:설탕, 3:블랙) ";
```

```
    cin>>coffee;
```

```
    coffee_machine(coffee);
```

```
    cout<<"손님 커피 여기 있습니다. "<<endl;
```

```
    cout<<"어떤 커피 드릴까요? (1:보통, 2:설탕, 3:블랙) ";
```

```
    cin>>coffee;
```

```
    coffee_machine(coffee);
```

```
    cout<<"손님 커피 여기 있습니다. "<<endl;
```

```
    .  
    .  
    .
```

주문을 여러 명에게 받도록 강의 중 실습 [기본2]를 변경

Loop을 사용하여
끝없이 주문
받는 자판기를
만들어보세요.

HW2 구의 부피를 구하는 함수 작성 (미리 정의된 수학 함수 pow사용해보기)

```
1 // Exercise 6.10 Solution: Ex06_10.cpp
2 // Inline function that calculates the volume of a sphere.
3 #include <iostream>
4 #include <cmath> //수학함수를 사용하기 위한 미리 정의된 라이브러리를 포함시키세요
5 using namespace std;
6
14
15 int main()
16 {
17     double radiusValue = 0;
18
19     // prompt user for radius
20     cout << "Enter the length of the radius of your sphere: ";
21     cin >> radiusValue; // input radius
22
23     // use radiusValue to calculate volume of sphere and display result
24     cout << "Volume of sphere with radius " << radiusValue
25         << " is " << sphereVolume( radiusValue ) << endl;
26 }
```

TIP

- 1) 반지름 r인 구의 부피 구하는 공식: $4.0 / 3.0 * PI * r^3$
- 2) a^b 를 pow를 사용하면 pow(a,b)

HW3 내가 직접 짠 my_pow 함수를 이용하세요.

```
1  // Exercise 6.10 Solution: Ex06_10.cpp
2  // Inline function that calculates the volume of a sphere.
3  #include <iostream>
4  #include <cmath>    //수학함수를 사용하기 위한 미리 정의된 라이브러리를 포함시키세요
5  using namespace std;
6
14
15  int main()
16  {
17      double radiusValue = 0;
18
19      // prompt user for radius
20      cout << "Enter the length of the radius of your sphere: ";
21      cin >> radiusValue; // input radius
22
23      // use radiusValue to calculate volume of sphere and display result
24      cout << "Volume of sphere with radius " << radiusValue
25           << " is " << sphereVolume( radiusValue ) << endl;
26  } // end main
```

TIP

- 1) float 입력, float 리턴값을 가지는 my_pow(a,b)
- 2) for 문 사용하여 my_pow함수 구현할 수 있음

HW4 rand() 함수로 주사위 6,000,000번 굴리는 시뮬레이션

```
1 // Fig. 6.9: fig06_09.cpp
2 // Rolling a six-sided die 6,000,000 times.
3 #include <iostream>
4 #include <iomanip>
5 #include <cstdlib> // contains function prototype for rand
6 using namespace std;
7
8 int main()
9 {
10     unsigned int frequency1 = 0; // count of 1s rolled
11     unsigned int frequency2 = 0; // count of 2s rolled
12     unsigned int frequency3 = 0; // count of 3s rolled
13     unsigned int frequency4 = 0; // count of 4s rolled
14     unsigned int frequency5 = 0; // count of 5s rolled
15     unsigned int frequency6 = 0; // count of 6s rolled
16
17     // summarize results of 6,000,000 rolls of a die
18     for ( unsigned int roll = 1; roll <= 6000000; ++roll )
19     {
```

//int face 선언.

//**rand()** 사용하여 주사위를 6,000,000굴려서

// 1~6 중 하나의 값을 무작위로 얻어서 face 변수에 대입

//face의 값에 따라 frequency의 값 하나씩 증가 (if 나 switch 사용)

HW4 rand() 함수로 주사위 굴리는 시뮬레이션

```
47
48     cout << "Face" << setw( 13 ) << "Frequency" << endl; // output headers
49     cout << "    1" << setw( 13 ) << frequency1
50         << "\n    2" << setw( 13 ) << frequency2
51         << "\n    3" << setw( 13 ) << frequency3
52         << "\n    4" << setw( 13 ) << frequency4
53         << "\n    5" << setw( 13 ) << frequency5
54         << "\n    6" << setw( 13 ) << frequency6 << endl;
55 } // end main
```

Face	Frequency
1	999702
2	1000823
3	999378
4	998898
5	1000777
6	1000422