

# Object-Oriented Programming Lab #03

Department: 화학공학과

Student ID: 2019101074

Name: 안용상

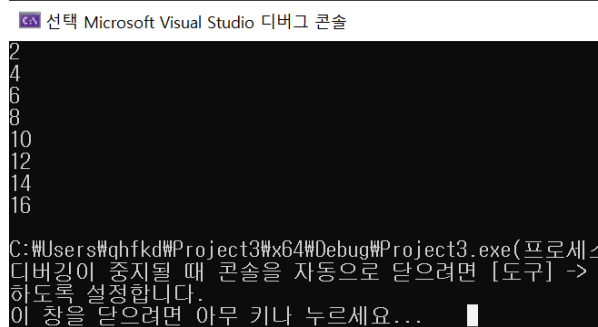
1. 아래 코드를 반복문을 이용하여 다시 작성하라.

```
std::cout << 2 << '\n';
std::cout << 4 << '\n';
std::cout << 6 << '\n';
std::cout << 8 << '\n';
std::cout << 10 << '\n';
std::cout << 12 << '\n';
std::cout << 14 << '\n';
std::cout << 16 << '\n';
```

- <다시 작성한 소스 코드>

```
#include <iostream>
#include <stdio.h>
int main(){
    for (int i = 1; i <=8;i++){ // for 구문을 이용해서 i를 1부터 8까지 순차적으로 증가시키며
        //총 8번의 실행을 하게 반복시키는 코드
        std::cout << i*2 << '\n';//i가 증가함에 따라 i*2도 2의 배수로 증가하게 되어 원하는
        //출력결과가 나오게됨
    }
}
```

- 출력결과



2. 아래 코드의 동작을 설명하라.

```
#include <iostream>
int main() {
    int a = 0;
    while (a < 20) {
        int b = 0;
        while (b < 10) {
            if ((a + b) % 2 == 0)
                std::cout << " ";
            b++;
        }
        std::cout << '\n';
        a++;
    }
}
```

[illegible]

```
#include <iostream>
int main() {
```

```
int x = 100, y;
while (x > 0) {
    std::cin >> y;
    if (y == 25) {
        x--;
        continue;
    }
    std::cin >> x;
    std::cout << "x = " << x << '\n';
}
}
```

#### <해답코드>

```
#include <iostream>
int main() {
    int x = 100, y;
    while (x > 0) {
        std::cin >> y;
        if (y != 25) { //이전 코드에서 continue때문에 if문이 참일때 건너뛰어졌던 부분을
//if로 먼저 조건에 넣는다.
//이부분이 이전에 if문이 참일때 건너뛰어졌고, if문이 False일때는 이 부분만 실행되었으니까
//이를 if / else구문으로 구사해도 똑같이 구현이 됨
            std::cin >> x;
            std::cout << "x = " << x << '\n';
        } else { // 이전 코드에서 continue가 있었던 if문을 else문으로 바꿈
            x--;
        }
    }
}
```

4. double형 자료를 20개 입력 받아서, 합, 평균, 최댓값과 최솟값을 출력하는 프로그램을 작성하라.

#### <작성한 프로그램 소스코드>

```
#include <stdio.h>
#include <iostream>

int main() {
    double input_data; //입력받은 double
    double sum; //입력받은 수들의 합을 할당하는 변수
    double max; //입력받은 수들 중에 MAX값을 할당하는 변수
    double min; //입력받은 수들 중에 MIN값을 할당하는 변수
    for (int i=1; i <= 20; i++) { //i를 1부터 20까지 증가시키며 20번의 시행을 함.
        std::cin >> input_data; //매 시행 마다 input_data를 새로 할당해줌
        if (i == 1) { //가장 첫번째 시행일 때, sum과 max와 min값에 처음 할당받은 input_data를
//할당해줌
            sum = input_data;
            max = input_data;
            min = input_data;
        }
        else { //두번째 시행부터는
            sum += input_data; //sum에는 매 시행마다 새로받은 input_data를 누적해서 합하고
            max = (max > input_data) ? max : input_data; //max에는 삼항연산자로, 현재 할당되어
//있는 max값과 지금 input받은 double수들 중 더 큰 수를 max변수에 재할당해주고
            min = (min < input_data) ? min : input_data; //min에는 반대로 삼항연산자로,
//현재 min에 할당된 값과, input받은 값들 중 더 작은 수를 min변수에 재할당해준다
        }
    }
    double mean = sum / 20; //최종적으로 for loop를 탈출한 뒤, 20번 누적된 총합 sum을 20으로
//나눠서 평균을 구해준다
    printf("합 : %f \n 평균 : %f \n 최댓값: %f \n 최솟값: %f \n ", sum, mean, max, min);
    //이를 출력해준다.
}
```

- 출력결과

```

Microsoft Visual Studio 디버그 콘솔

421
553
-21321
43
123
545
613
431
666
21321
3215
-3200
123
431
55
123
421
123
55
합 : 4864.000000
평균 : 243.200000
최댓값: 21321.000000
최솟값: -21321.000000

C:\Users\qhfkd\Project3\Debug\Project3.exe(프로세스 35892개)이
었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디
버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

```

5. 아래 코드의 동작을 설명하라.

```

int a = 0;
while (a < 100);
    std::cout << a++;
std::cout << '\n';

```

#### • <코드동작 설명>

```

int a = 0; //a를 정수형 0으로 초기화한다
while (a < 100); //a가 100보다 작은 값이면 while문 안의 소스코드를 실행한다
    //하지만 이경우 while문 안의 소스코드가 아무것도 작성된게 없는 상태로
    //를 찍었으므로 어떤 연산도 수행되지 않는다.
    //그러므로 a는 계속 0인 상태고, while문이 실행되는
    //때 시행에 (a<100)이라는 조건이 계속 True가 되어
    //while (a<100)의 루프가 계속 실행되고, 멈추지않게 된다.
    std::cout << a++; //a를 출력한뒤, a+=1연산을 수행한다. 하지만 이경우 while문에
    //감혀서 나오지 못하기에 코드가 수행되지않는다.
std::cout << '\n'; //이또한 while문에서 더 내려오지 못하므로 줄바꿈 출력이 수행되지
//않는다

```

6. 아래와 같이 출력되는 프로그램을 작성하라.

```

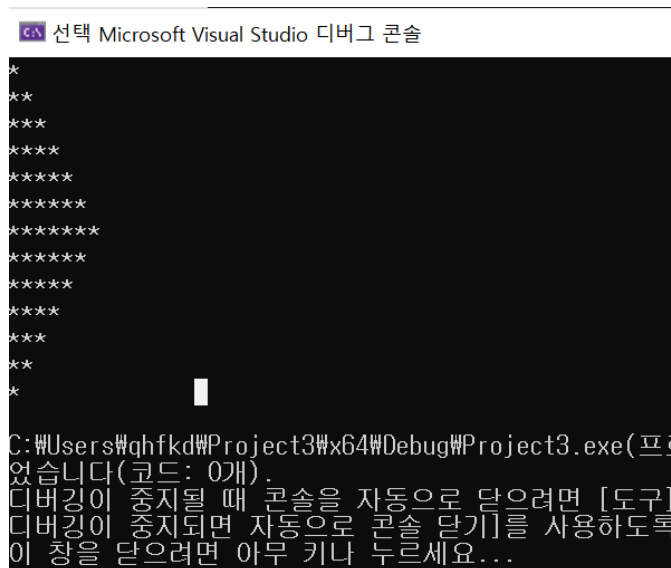
*
**
***
****
*****

```

- <작성한 소스코드>

```
#include <stdio.h>
#include <iostream>
#include <string> //문자 한개 char가 아니라 여러개를 이은 문자열을 이용할 것이므로 string을
//헤더파일로 호출한다.
int main() {
    std::string str = ""; //std::string으로 str를 string변수로 선언하고 *를 할당한다.
    int count; // *를 몇 번 이어붙힐지를 결정해주는 count변수를 생성한다.
    for (int i=1; i <= 13; i++) {
        if (i > 7) {
            count = 14 - i; //8번째 시행부터는 다시 *의 개수가 1개씩 작아지므로
                            //count, 즉 몇번 이어붙힐지에 대한 기준값을 14-i로 두어
                            //count를 순차적으로 차감시킨다
        }
        else {
            count = i; //1-7번째 시행까지는 count를 시행 회차와 같이 설정한다.
        }
        for (int j = 1; j <= count; j++) { //이 for문은 상위 for문 안에 위치하는 for문이다.
            //상위 for문에서 한 시행에서 결정된 count, 즉 몇번 *을 이어붙힐지에 대한
            //숫자값을 이용한다.
            //그 count를 이 for문의 반복횟수로 지정하기위해
            //for (int j = 1; j <= count; j++) 와 같이 for문을 지정해준다.
            std::cout << " "; //그리고 for문안에서 *한개를 출력하는 출력코드를 작성한다
            if (count == j) {
                std::cout << std::endl; //만약 count번째 *출력이라면, 줄바꿈을 해준다.
            }
        }
        //결과적으로 안에있는 for문이 끝나고 보면 count만큼의 *이 이어붙혀진 한줄이 출력되어있다.
    }
}
```

- 출력결과



7. 오일러의 수(Euler's number,  $e$ )는 자연로그의 밑(base of natural logarithms)이며, 아래와 같이 근사화할 수 있다.

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots \leftarrow$$

위 수식을 이용하여  $e$ 의 근사값을 계산하여 출력하는 프로그램을 작성하라. 단, 더해주는 항의 값이 0.00001보다 작은 값이 나타날 때까지 계산하라.

- 작성한 소스코드

```
#include <stdio.h>

int main() {
    double add = 1.0; //더해줄 값(첫번째 더해줄 값은 1로 정해져있으므로 1로 초기화)
    double sum = 1.0; //총 합 (첫번째 더해줄 값을 더해주고 시작하면 첫 누적합은
    //1로 되기때문에 1로 초기화
    int eter = 1; //시행 반복 횟수

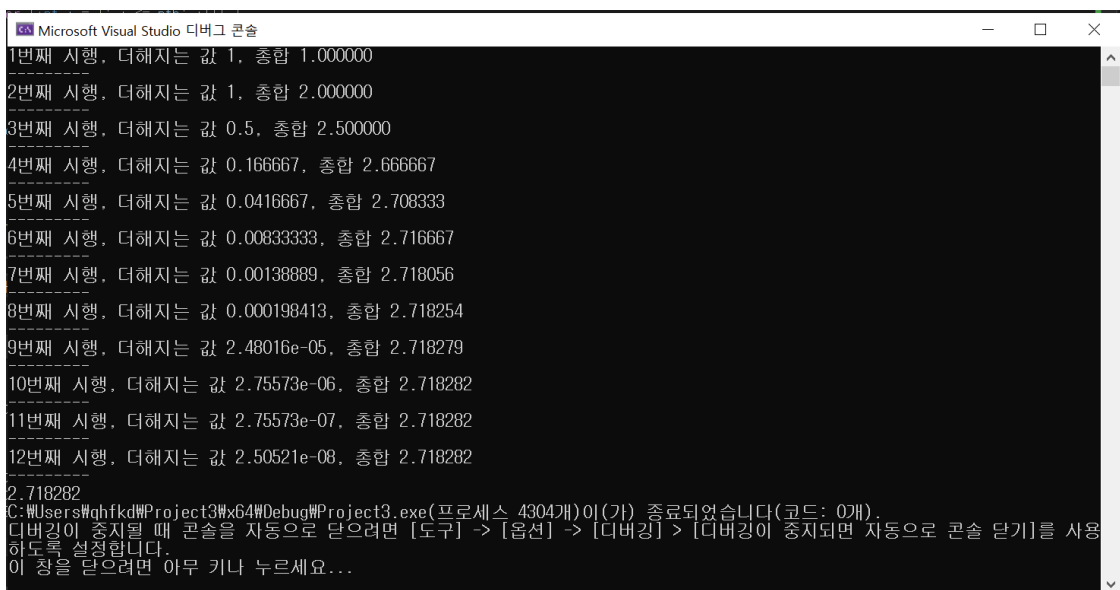
    while (add > 0.00000001) {
        printf("%d번째 시행, 더해지는 값 %g, 총합 %f \n-----\n", eter, add, sum);

        double all_dot = 1.0; //각 요소들의 분모. (팩토리얼, 즉 누적곱)
        for (int i = 2; i <= eter; i++) { //2부터 시작하는 i가 시행반복횟수보다 같거나
            //작은 값이면 for loop를 반복한다.(여기서 2부터 시작하는 이유는 1은 곱하는게 의미 없기 때문)
            all_dot *= i;
            //반복되는 동안 all_dot에 누적으로 i를 곱해준다
        }

        add = 1.0 / all_dot; //더해줄 값인 add에 1을 누적곱 all_dot로 나눠준 값을 할당.
        sum += add; // 총 합에 add값을 누적시킴
        eter++; //시행 횟수 변수를 1 증가시킴.
    }

    printf("%f", sum); // 합을 출력한다
    return 0;
}
```

- 출력결과



```
Microsoft Visual Studio 디버그 콘솔
1번째 시행, 더해지는 값 1, 총합 1.000000
2번째 시행, 더해지는 값 1, 총합 2.000000
3번째 시행, 더해지는 값 0.5, 총합 2.500000
4번째 시행, 더해지는 값 0.166667, 총합 2.666667
5번째 시행, 더해지는 값 0.0416667, 총합 2.708333
6번째 시행, 더해지는 값 0.00833333, 총합 2.716667
7번째 시행, 더해지는 값 0.00138889, 총합 2.718056
8번째 시행, 더해지는 값 0.000198413, 총합 2.718254
9번째 시행, 더해지는 값 2.48016e-05, 총합 2.718279
10번째 시행, 더해지는 값 2.75573e-06, 총합 2.718282
11번째 시행, 더해지는 값 2.75573e-07, 총합 2.718282
12번째 시행, 더해지는 값 2.50521e-08, 총합 2.718282
2.718282
C:\Users\ghfkd\Project3\Debug\Project3.exe(프로세스 4304개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```

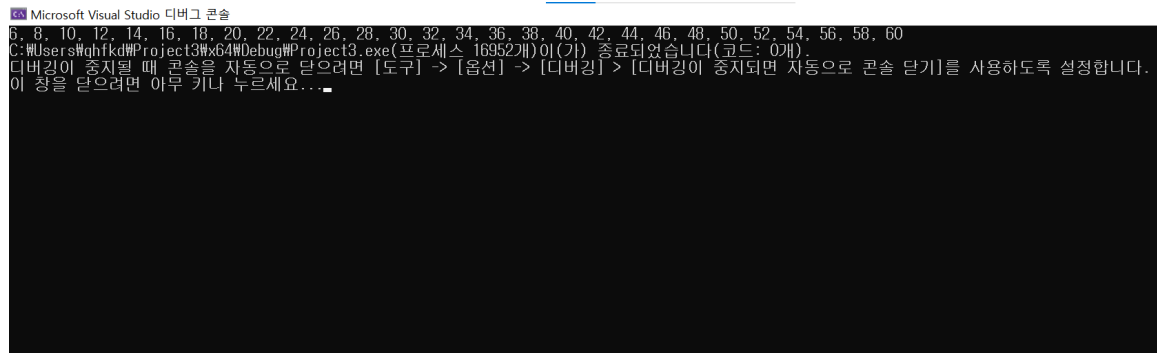
8. for 문을 이용하여 아래와 같이 출력되는 프로그램을 작성하라.

```
6, 8, 10, 12, ..., 60
```

- 작성소스코드

```
#include <iostream>
#include <stdio.h>
int main() {
    for (int i = 1; i <= 60; i++) { //i를 1부터 60까지 늘리며 60번 loop반복
        if ((i >= 6) && (i % 2 == 0)) { //i가 6과 같거나 크고, 짝수이면 if문 내부 소스코드를 실행
            if(i==60){//i가 6과 같거나 크고, 짝수이면서 i가 60이면, 즉 마지막 시행이면 i만 출력한다
                std::cout << i;
            }else{
                std::cout << i << " "; //i가 6과 같거나 크고, 짝수이면서 i가 60보다 작으면,
            } //즉 마지막 시행이 아니면 i를 출력하고 ,까지 같이 출력한다.
        }
    }
}
```

- 출력결과



9. for 문을 이용하여 아래와 같이 출력되는 프로그램을 작성하라.

```
7, 9, 11, 13, ..., 67
```

- 작성소스코드

```
#include <iostream>
#include <stdio.h>
int main() {
    for (int i = 1; i <= 67; i++) { //i를 1부터 67까지 늘리며 67번 loop반복
        if ((i >= 7) && (i % 2 == 1)) { //i가 7과 같거나 크고, 홀수이면 if문 내부 소스코드를 실행
            if (i == 67) { //i가 7과 같거나 크고, 홀수이면서 i가 67이면, 즉 마지막 시행이면 i만 출력한다
                std::cout << i;
            }
            else {
                std::cout << i << " "; //i가 7과 같거나 크고, 홀수이면서 i가 67보다 작으면,
            } //즉 마지막 시행이 아니면 i를 출력하고 ,까지 같이 출력한다.
        }
    }
}
```

- 출력결과

```
Microsoft Visual Studio 디버그 콘솔
7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67
C:\Users\qhfkd\Project3\Debug\Project3.exe(프로세스 35268개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```

10. for 문을 이용하여 아래와 같이 출력되는 프로그램을 작성하라.

1과 15까지의 합

- 작성소스코드

```
#include <iostream>
#include <stdio.h>
int main() {
    int sum = 0;
    for (int i = 1; i <= 15; i++) { //i를 1부터 15까지 증가시키면서 15번 루프를 반복
        sum += i; // sum변수에 i를 계속 누적 합해줌.
    }
    printf("1과 15까지의 합 : %d", sum); //누적합 결과를 출력
}
```

- 출력결과

```
Microsoft Visual Studio 디버그 콘솔
1과 15까지의 합 : 120
C:\Users\qhfkd\Project3\Debug\Project3.exe(프로세스 232
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션
이 창을 닫으려면 아무 키나 누르세요...
```

11. for 문을 이용하여 아래와 같이 출력되는 프로그램을 작성하라.

1, 4, 7, 10, ... 의 수열에서 50개의 출력

- 작성소스코드

```
#include <iostream>
#include <stdio.h>
int main() {
    int number = 1; //수열의 시작 숫자
    for (int i = 1; i <= 50; i++) { //i를 1부터 50까지 증가시키며, 50번 루프반복
```



```

    if (i < 50) { //i가 50보다 작으면 number를 출력하고 ", "도 같이 출력
        std::cout << number << ", ";
    }
    else {
        std::cout << number; //i가 50이면 즉 마지막 시행이면 ", "없이 그냥 number만 출력
    }
    number += 3; //등차수열의 간격이 3이므로, 3을 더해주고 number에 재할당
}
}

```

- 출력결과

