# 객체지향프로그래밍 - 과제5

## 2019110634 컴퓨터공학과 이창렬

### A. Exercises (p.153-157)

1. In Listing 6.4(addnonnegatives.cpp) could the condition of the if statement have used > instead of >= and achieved the same results? Why?

if문에서 0을 포함해도 sum값에 추가할 것이 없어서 상관없습니다.

3. Use a loop to rewrite the following code fragment so that it uses just one std::cout and one '\n'.

```
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';
```

```
for (int i = 1; i<9; i++)
  std::cout << 2*i << '\n';
```

8. How many asterisks does the following code fragment print?

```
int a = 0;
while (a < 100) {
  int b = 0;
  while (b < 55) {
    std::cout << "*";
    b++;
  }
  std::cout << '\n';
}
```

무제한으로 나옵니다.

12. What is printed by the following code fragment?

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

0123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051 5

14. Rewrite the following code fragment using a break statement and eliminating the done variable.
Your code should behave identically to this code fragment.

```
bool done = false;
int n = 0, m = 100;
while (!done && n != m) {
  std::cin >> n;
  if (n < 0)
    done = true;
  std::cout << "n = " << n << '\n';
}
```

```
#include <iostream>
int main() {
  bool done = false;
  int n = 0, m = 100;
  while (!done && n != m) {
    std::cin >> n;
    if (n < 0)
      break;
    std::cout << "n = " << n << '\n';
  }
}
```

15. Rewrite the following code fragment so it eliminates the continue statement. Your new code's logic should be simpler than the logic of this fragment.

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

```
#include <iostream>
using namespace std;

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

19. Write a C++ program that allows the user to enter exactly twenty double-precision floating-point values. The program then prints the sum, average (arithmetic mean), maximum, and minimum of the values entered.

```
#include <iostream>
using namespace std;

int main() {
  double x = 0, sum = 0, min = 0, max = 0;
  for (int i = 0; i < 20; i++) {
    cout <<i+1 << "번째 숫자 : ";
    cin >> x;
    if (max <= x)
      max = x;
    if (min >= x)
      min = x;
    sum += x;
  }
  double avg = sum / 20;
  cout << "sum :" << sum << " avg : " << avg << " max : " << max << " min : " << min;
}
```

22. Redesign Listing 6.21 (startree.cpp) so that it draws a sideways tree pointing left; for example, if the user enters 7, the program would print

```
#include<iostream>
using namespace std;
int main()
{
  int n, i, j;
  cout << "Enter number of rows: ";
  cin >> n;
  for (i = 1; i <= n; i++) {
    for (j = i; j < n; j++) {
      cout << " ";
    }
    for (j = 1; j <= i; j++){
      cout << "*";
    }
    cout << "\n";
  }
  for (i = n; i >= 1; i--) {
    for (j = i; j <= n; j++) {
      cout << " ";
    }
    for (j = 1; j < i; j++) {
      cout << "*";
    }
    // ending line after each row
    cout << "\n";
  }
  return 0;
}
```

## B. Exercises (p.173-177)

2. Consider the following code fragment.

```
char ch;
std::cin >> ch;
switch (ch) {
case 'a':
std::cout << "*\n";
break;
case 'A':
std::cout << "**\n";
break;
case 'B':
case 'b':
std::cout << "***\n";
case 'C':
case 'c':
std::cout << "****\n";
break;
default:
std::cout << "*****\n";
}
```

(a) *

(b) **

(c) ***

(d) *** \n ****

(e) ****

(f) ****

(g) *****

3. What is printed by the following code fragment?

```
int x = 0;
do {
std::cout << x << " ";
x++;
} while (x < 10);
std::cout << '\n';
```

0 1 2 3 4 5 6 7 8 9

5. What is printed by the following code fragment?

```
for (int x = 0; x < 10; x++)
std::cout << "*";
std::cout << '\n';
```

*********

## C. additional exercises

C-1. Euler's number, e, is used as the base of natural logarithms. It can be approximated using the following formula. Write a program that approximate e using a loop that terminates when the difference between two successive values of e is less than 0.00001.

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
  double term = 1.0;
  double sum = 1.0;
  int n = 0;
  while (term >= 0.00001) {
    n++;
    term = term / n;
    sum = sum + term;
  }
  cout << "Approximate value of e is : " << sum;
}
```

C-2. Write a program that reads a positive integer less than 1,000,000 from the keyboard and then prints it out in reverse. For example, if the user enters 2576, it prints 6752.

```
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;

int main() {
  int n = 6, x;
  cout << "Enter number : ";
  cin >> x;
  string str = to_string(x);
  /*int len = str.length();
  for (int i = 0; i < (len / 2); i++) {
    swap(str[i], str[n]);
    n -= 1;
  }
  cout << str << endl;
  */
  reverse(str.begin(), str.end());
  cout << str;
}
```

* Write a for loop that will produce each of following sequences.

C-3. 6, 8, 10, 12, ... 60

```
for (int i = 1; i < 21; i++)
    std::cout << 3 * i << '\t';
```

C-4. 7, 9 ,11, 13, ..., 67

```
for (int i = 3; i < 34; i++)
    std::cout << 2 * i+1 << '\t';
```

C-5. The sum of the numbers between 1 and 15 inclusive.

```
int sum = 0;
  for (int i = 1; i < 16; i++)
    sum += i;
  std::cout << sum;
```

C-6. The first 50 numbers in the series, 1, 4, 7, 10, ....

```
for (int count = 0; count < 50; count++)
    std::cout << 1+3*count << '\n';
```