Object-Oriented Programming Lab #06

Department: 응용물리학과 Student ID: 2017103038 Name: 권인회

A. Code explanation & output analysis (Write the source code and results)

A-1. Listing 8.4

: < 코드 >

```
#include <iostream>
#include <cmath>
int main() {
// Location of orbiting point is (x,y)
double x; // These values change as the
double y; // satellite moves
const double PI = 3.14159;
// Location of fixed point is always (100, 0),
// AKA (p_x, p_y). Change these as necessary.
const double p_x = 100;
const double p_y = 0;
// Radians in 10 degrees
const double radians = 10 * PI/180;
// Precompute the cosine and sine of 10 degrees
const double COS10 = cos(radians);
const double SIN10 = sin(radians);
// Get starting point from user
std::cout << "Enter initial satellite coordinates (x,y):";
std::cin >> x >> y;
// Compute the initial distance
```

```
double d1 = sqrt((p_x - x)*(p_x - x) + (p_y - y)*(p_y - y));

// Let the satellite orbit 10 degrees

double x_old = x; // Remember x's original value

x = x*COS10 - y*SIN10; // Compute new x value

// x's value has changed, but y's calculate depends on

// x's original value, so use x_old instead of x.

y = x_old*SIN10 + y*COS10;

// Compute the new distance

double d2 = sqrt((p_x - x)*(p_x - x) + (p_y - y)*(p_y - y));

// Print the difference in the distances

std::cout << "Difference in distances: " << d2 - d1 << '₩n'; }
```

< 실행결과 예시 >

```
Enter initial satellite coordinates (x,y):10 0

Difference in distances: 0.168644

C:₩Users₩Administrator₩source₩repos₩Ex002₩Debug₩Ex002.exe(프로세스 26388개)이(가
) 종료되었습니다(코드: 0개).

이 창을 닫으려면 아무 키나 누르세요...
```

< 설명 >

이 문제는 두 지점과의 거리를 구하는 문제이다. 좌표상에서 관측자가 (100,0)에서 바라볼 때, 원점을 중심으로 원운동하는 궤도 상의 한 점 (x1,y1)를 입력받고 그 한 점에서 10도 이동한 만큼의 궤도의 한 점 (x2,y2)이라 할 때, 관측자를 기준으로 그 한점과의 거리가 얼마나 변하였는지 측정하는 코드이다. 따라서 입력받을 두 double형 변수 x,y를 선언하고 double형 상수로 PI값과 관측자의 위치 (p_x,p_y), 10도, cos10도, sin10도 값을 선언해준다. Cin을 통해 x,y 값을 입력받고 첫 두 점 사이의 거리를 구한다. 이것은 관측자와 입력받은 좌표 사이의 거리이다. 따라서 두 점 사이의

거리 공식에 따라 sqrt((p_x - x)*(p_x - x) + (p_y - y)*(p_y - y))를 계산하고 d1에 넣는다. 이후, x와 y 값을 원점에 대해 궤도운동하면서 10도 움직인 위치로 수정해주고 다시 d2를 계산한다. 그러고서 d2 – d1 값을 출력하면서 관측자가 볼 때 궤도상의 점까지의 거리가 얼마나 변하였는지 알 수 있다.

A-2. Listing 8.12

: < 코드 >

```
#include <iostream>

#include <cstdlib>

#include <ctime>

int main() {

// Set the random seed value

srand(static_cast(time(0)));

// Roll the die three times

for (int i = 0; i < 3; i++) {

// Generate random number in the range 1...6

int value = rand() % 6 + 1;

// Show the die

std::cout << "+-------+₩n";

switch (value) {

case 1:

std::cout << "| |₩n";

std::cout << "| * |₩n";

std::cout << "| |₩n";

break;

case 2:
```

```
std::cout << "| * |₩n";
std::cout << "| |₩n";
std::cout << "| * |₩n";
break;
case 3:
std::cout << "| * |₩n";
std::cout << "| * |₩n";
std::cout << "| * |₩n";
break;
case 4:
std::cout << "| * * |₩n";
std::cout << "| |₩n";
std::cout << "| * * |₩n";
break;
case 5:
std::cout << "| * * |₩n";
std::cout << "| * |₩n";
std::cout << "| * * |₩n";
break;
case 6:
std::cout << "| * * * |₩n";
std::cout << "| |₩n";
std::cout << "| * * * |₩n";
break;
default:
std::cout << " *** Error: illegal die value ***₩n";
```

```
break;
} std::cout << "+-------+₩n"; }
}
```

< 실행결과 예시 >

```
+-------+
|  * *  |
|   *   |
|  * *  |
+-------+
+-------+
|   *   |
|       |
|   *   |
+-------+
+-------+
| * * * |
|       |
| * * * |
+-------+
```

C:₩Users₩Administrator₩source₩repos₩Ex002₩Debug₩Ex002.exe(프로세스 32872개)이(가
) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

< 설명 >

난수 생성, 시간 라이브러리를 불러온 뒤 srand를 통해 난수 생성 초기 시드 값을 시스템 클럭으

로 설정한다. 그 뒤, 주사위를 for문을 통해 3번 굴린다. 이 때 주사위의 결과는 랜덤으로 나올 것이고 확률은 매우 비슷하다. value값을 rand() % 6 으로 설정하여 0~5사이의 난수를 입력받아 1을 더해주어 1~6사이의 값이 입력되도록 한다. Switch case문을 통해 1~6까지의 주사위 그림을 설정해놓고 value에 따라 1~6에 알맞은 그림이 출력되도록 한다. 그럴리는 없겠지만 1~6 외의 값이 입력되면 에러문구가 출력된다. 이 과정을 for문을 통해 3번 거쳐 주사위 3개의 그림이 랜덤으로 나오게 되는 것이다.

A-3. Listing 9.11

: < 코드 >

```
#include <iostream>
#include <cmath>
/*
* help_screen
* Displays information about how the program works
* Accepts no parameters
* Returns nothing
*/
void help_screen() {
std::cout << "Add: Adds two numbers₩n";
std::cout << " Example: a 2.5 8.0₩n";
std::cout << "Subtract: Subtracts two numbers₩n";
std::cout << " Example: s 10.5 8.0₩n";
std::cout << "Print: Displays the result of the latest operation₩n";
std::cout << " Example: p₩n";
std::cout << "Help: Displays this help screen₩n";
std::cout << " Example: h₩n";
std::cout << "Quit: Exits the program₩n";
```

```
std::cout << " Example: q₩n";
} /*
* menu
* Display a menu
* Accepts no parameters
* Returns the character entered by the user.
*/
char menu() {
// Display a menu
std::cout << "=== A)dd S)ubtract P)rint H)elp Q)uit ===₩n";
// Return the char entered by user
char ch;
std::cin >> ch;
return ch;
}
/*
* main
* Runs a command loop that allows users to
* perform simple arithmetic.
*/
int main() {
double result = 0.0, arg1, arg2;
bool done = false; // Initially not done
do {
switch (menu()) {
case 'A': // Addition
```

```
case 'a':

std::cin >> arg1 >> arg2;

result = arg1 + arg2;

std::cout << result << '₩n';

break;

case 'S': // Subtraction

case 's':

std::cin >> arg1 >> arg2;

result = arg1 - arg2;

// Fall through, so it prints the result

case 'P': // Print result

case 'p':

std::cout << result << '₩n';

break;

case 'H': // Display help screen

case 'h':

help_screen();

break;

case 'Q': // Quit the program

case 'q':

done = true;

break;

} } while (!done); }
```

< 실행결과 예시 >

```
=== A)dd S)ubtract P)rint H)elp Q)uit ===

a
```

1.1 2.2

3.3

=== A)dd S)ubtract P)rint H)elp Q)uit ===

s

2.2 1.1

1.1

=== A)dd S)ubtract P)rint H)elp Q)uit ===

p

1.1

=== A)dd S)ubtract P)rint H)elp Q)uit ===

h

Add: Adds two numbers

 Example: a 2.5 8.0

Subtract: Subtracts two numbers

 Example: s 10.5 8.0

Print: Displays the result of the latest operation

 Example: p

Help: Displays this help screen

 Example: h

Quit: Exits the program

 Example: q

=== A)dd S)ubtract P)rint H)elp Q)uit ===

q

C:₩Users₩Administrator₩source₩repos₩Ex002₩Debug₩Ex002.exe(프로세스 31700개)이(가

) 종료되었습니다(코드: 0개).

이 창을 닫으려면 아무 키나 누르세요...

< 설명 >

Help_screen, menu의 함수를 정의해두고 main 내에서 실행시킨다. 함수를 살펴보면 help_screen은 정보를 전달하기 위해 문장들을 출력해주는 것이고, menu에서는 정보를 출력해주고 문자형 ch를 선언하고 입력 받아 ch를 return 한다. 따라서 main 내를 보면 필요한 변수들을 double형으로 선언해주고 while 문을 벗어나올 때 사용하기 위해 done을 boolean형으로 선언해준다. Do while 문 내에 switch case문을 둬서 입력하는 것에 따라 case문이 계속 반복해서 돌아갈 수 있도록 해뒀다. Menu 함수의 return 값에 따라 case가 정해지며 각 case를 살펴보면, A나 a일 때는 두 변수를 입력받아 덧셈 계산후 출력, S나 s일 때는 두 변수를 입력받아 뺄셈 계산후 출력, H나 h일 때는 help_screen함수 실행하여 안에 있던 설명이 출력되도록 한다. P나 p일 때는 가장 최근의 결과값을 출력해주고 Q나 q를 입력하면 done을 true로 바꾸어 while문에서 빠져나가도록 한다.

A-4. Listing 9.12

: < 코드 >

```
#include <iostream>
/*
* get_int_range(first, last)
* Forces the user to enter an integer within a
* specified range
* first is either a minimum or maximum acceptable value
* last is the corresponding other end of the range,
* either a maximum or minimum * value
* Returns an acceptable value from the user
*/
int get_int_range(int first, int last) {
// If the larger number is provided first,
// switch the parameters
if (first > last) {
```

```
int temp = first;
first = last;
last = temp;
}
// Insist on values in the range first...last
std::cout << "Please enter a value in the range " << first << "..." << last << ": ";
int in_value; // User input value
bool bad_entry;
do {
std::cin >> in_value;
bad_entry = (in_value < first || in_value > last);
if (bad_entry) {
std::cout << in_value << " is not in the range " << first << "..." << last << '₩n';
std::cout << "Please try again: ";
}
}
while (bad_entry);
// in_value at this point is guaranteed to be within range
return in_value;
}
/*
* main
* Tests the get_int_range function
*/
int main() {
std::cout << get_int_range(10, 20) << '₩n';
```

```
std::cout << get_int_range(20, 10) << '₩n';

std::cout << get_int_range(5, 5) << '₩n';

std::cout << get_int_range(-100, 100) << '₩n';

}
```

< 실행결과 예시 >

```
Please enter a value in the range 10...20: 9

9 is not in the range 10...20

Please try again: 10

10

Please enter a value in the range 10...20: 15

15

Please enter a value in the range 5...5: 5

5

Please enter a value in the range -100...100: -99

-99


C:₩Users₩Administrator₩source₩repos₩Ex002₩Debug₩Ex002.exe(프로세스 46516개)이(가

) 종료되었습니다(코드: 0개).

이 창을 닫으려면 아무 키나 누르세요...
```

< 설명 >

이번엔 get_int_range 함수에 대해서 설명하겠다. 두 int형 변수를 parameter로 받아 두 정수의 크기를 비교하여 작은 것을 변수 first, 큰 것을 변수 last로 할당한다. 이후 first와 last 범위 내의 수를 cin을 통해 입력받고 이 in_value가 범위 내에 있으면 선언한 boolean형 bad_entry가 false가 되어 while문을 벗어난 뒤, in_value를 return한다. 만약 범위 밖에 있다면 범위 안에 없다는 안내 문구가 출력되며 while문을 돌아 다시 한번 in_value를 입력받을 수 있도록 한다. 아래 main에서는 이 함수에 대한 test를 한 것이며 실행결과 제대로 작동하는 것을 볼 수 있다.

A-5. Listing 9.17

: < 코드 >

```
#include <iostream>
#include <cmath>
/*
* equals(a, b, tolerance)
* Returns true if a = b or |a - b| < tolerance.
* If a and b differ by only a small amount
* (specified by tolerance), a and b are considered
* "equal." Useful to account for floating-point
* round-off error.
* The == operator is checked first since some special
* floating-point values such as HUGE_VAL require an
* exact equality check.
*/
bool equals(double a, double b, double tolerance) {
return a == b || fabs(a - b) < tolerance;
}
int main() {
for (double i = 0.0; !equals(i, 1.0, 0.0001); i += 0.1)
std::cout << "i = " << i << '₩n';
}
```

< 실행결과 >

```
i = 0
i = 0.1
```

i = 0.2

i = 0.3

i = 0.4

i = 0.5

i = 0.6

i = 0.7

i = 0.8

i = 0.9

C:₩Users₩Administrator₩source₩repos₩Ex002₩Debug₩Ex002.exe(프로세스 48492개)이(가

) 종료되었습니다(코드: 0개).

이 창을 닫으려면 아무 키나 누르세요...

< 설명 >

Equals 함수를 boolean형으로 return 되도록 정의하였다. 3개의 double형 변수를 파라미터로 받아 a와 b가 같거나, a와 b의 차가 tolerance보다 작을 경우에 true를 return한다. Main 내에서 for문을 살펴보자. double형인 변수 i가 0.0에서 0.1씩 증가하는 동안 조건이 true이면 for문 안에 있는 내용을 출력한다. 조건에서 !equals(I, 1.0, 0.0001)이므로 i와 1.0이 같거나 차가 0.0001보다 작을 경우 equals가 true가 되는 것이다. 따라서, equals가 false일 동안 for문이 실행되고 i가 커져 equals가 true가 되는 순간 프로그램이 종료되는 것이다. 따라서, i가 0.9 일때까지만 출력되고 i가 1.0이 되는 순간 프로그램이 종료되는 것을 볼 수 있다.

A-6. Listing 9.18

< 코드 >

```
#include <iostream>
#include <iomanip>
// Print the column labels for an n x n multiplication table.
void col_numbers(int n) {
        std::cout << " ";
        for (int column = 1; column <= n; column++)
                std::cout << std::setw(4) << column; // Print heading for this column.
```

```
        std::cout << '₩n';
}
// Print the table's horizontal line at the top of the table
// beneath the column labels.
void col_line(int n) {
        std::cout << " +";
        for (int column = 1; column <= n; column++)
                std::cout << "----"; // Print separator for this row.
        std::cout << '₩n';
}
// Print the title of each column across the top of the table
// including the line separator.
void col_header(int n) {
        // Print column titles
        col_numbers(n);
        // Print line separator
        col_line(n);
}
// Print the title that appears before each row of the table's
// body.
void row_header(int n) {
        std::cout << std::setw(4) << n << " |"; // Print row label.
}
// Print the line of text for row n
// This includes the row number and the
// contents of each row.
void print_row(int row, int columns) {
        row_header(row);
        for (int col = 1; col <= columns; col++)
                std::cout << std::setw(4) << row * col; // Display product
        std::cout << '₩n'; // Move cursor to next row
}
// Print the body of the n x n multiplication table
void print_contents(int n) {
        for (int current_row = 1; current_row <= n; current_row++)
                print_row(current_row, n);
}
// Print a multiplication table of size n x n.
void timestable(int n) {
        // First, print column heading
        col_header(n);
        // Print table contents
        print_contents(n);
}
// Forces the user to enter an integer within a
// specified range first is either a minimum or maximum
// acceptable value last is the corresponding other end
// of the range, either a maximum or minimum value
// Returns an acceptable value from the user
int get_int_range(int first, int last) {
        // If the larger number is provided first,
        // switch the parameters
        if (first > last) {
                int temp = first;
                first = last;
```

```
                last = temp;
        }
        // Insist on values in the range first...last
        std::cout << "Please enter a value in the range "
                << first << "..." << last << ": ";
        int in_value; // User input value
        bool bad_entry;
        do {
                std::cin >> in_value;
                bad_entry = (in_value < first || in_value > last);
                if (bad_entry) {
                        std::cout << in_value << " is not in the range "
                                << first << "..." << last << '₩n';
                        std::cout << "Please try again: ";
                }
        } while (bad_entry);
        // in_value at this point is guaranteed to be within range
        return in_value;
}
int main() {
        // Get table size from user; allow values in the
        // range 1...18.
        int size = get_int_range(1, 18);
        // Print a size x size multiplication table
        timestable(size);
}
```

< 실행결과 예시 >

Please enter a value in the range 1...18: 17

```
    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17

 +-----------------------------------------------------------------

   1|   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17

   2|   2   4   6   8  10  12  14  16  18  20  22  24  26  28  30  32  34

   3|   3   6   9  12  15  18  21  24  27  30  33  36  39  42  45  48  51

   4|   4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68

   5|   5  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80  85

   6|   6  12  18  24  30  36  42  48  54  60  66  72  78  84  90  96 102

   7|   7  14  21  28  35  42  49  56  63  70  77  84  91  98 105 112 119

   8|   8  16  24  32  40  48  56  64  72  80  88  96 104 112 120 128 136

   9|   9  18  27  36  45  54  63  72  81  90  99 108 117 126 135 144 153
```

```
 10 |  10  20  30  40  50  60  70  80  90 100 110 120 130 140 150 160 170
 11 |  11  22  33  44  55  66  77  88  99 110 121 132 143 154 165 176 187
 12 |  12  24  36  48  60  72  84  96 108 120 132 144 156 168 180 192 204
 13 |  13  26  39  52  65  78  91 104 117 130 143 156 169 182 195 208 221
 14 |  14  28  42  56  70  84  98 112 126 140 154 168 182 196 210 224 238
 15 |  15  30  45  60  75  90 105 120 135 150 165 180 195 210 225 240 255
 16 |  16  32  48  64  80  96 112 128 144 160 176 192 208 224 240 256 272
 17 |  17  34  51  68  85 102 119 136 153 170 187 204 221 238 255 272 289
```

C:₩Users₩Administrator₩source₩repos₩Ex002₩Debug₩Ex002.exe(프로세스 51492개)이(가

) 종료되었습니다(코드: 0개).

이 창을 닫으려면 아무 키나 누르세요...

< 설명 >

구구단을 표 형식으로 나타내는 코드이다. 함수를 차례대로 살펴보자. Col_numbers 함수는 정수형 변수 하나를 파라미터로 받아 실행된다. 핵심만 설명하면 for문인데 setw에 따라 일정한 간격을 유지하면서 1부터 n까지 숫자를 나열한다. 표에서 맨 위 행을 나타낼 것으로 보인다. Col_line 함수는 정수형 변수 하나를 파라미터로 받아 실행된다. 이것도 형식적인 것들을 출력하고 n의 크기만큼 "----"를 출력하여 첫 행을 구분해주는데 사용한다. 이 두 함수를 합쳐서 col_header 함수로 꾸려 동시에 실행될 수 있도록 한다. Row_header 함수는 setw에 따라 일정한 간격에서 n과 " | " 를 차례로 출력한다. 아마 첫 열과 구분선으로 보인다. Print_row 함수는 두 개의 정수형 파라미터 row와 colums를 받는다. 안에는 row를 파라미터로 받아 row_header 함수도 실행하게 되어있고 for문에 따라 columns의 크기만큼 row와 col을 곱해 일정한 간격을 유지하면서 출력하도록 되어 있다. 이 함수를 print_contents 함수에 의해 n번만큼 반복하도록 for문 안에 넣어 실행시킨다. 그렇다면 한 행을 채우는 print_row 함수를 n번 반복하여 n행까지 내용이 자동으로 증가되면서 출력된다. 이 과정을 col_header로 머리행을 만들고 print_contents로 내용행을 만들도록 timestable 함수 하나로 묶어준다. Get_int_range 함수는 위에서 설명하였으므로 생략한다. 따라서 main 내에서 get_int_range 함수에 의해 정해진 범위에 따라 size 입력값을 받아 그 size에 따른 구구단 표를 출력하게 되는 것이다.

B. Exercises (Write the questions down on your answer sheet)

(pp. 197-199), Exercises 3, 8, 9
(write output analysis for all exercises)

3. Which one of the following values could be computed by the rand function?

4.5 34 -1 RAND_MAX + 1

: 34이다. rand함수에서는 정수값만 나오므로 4.5는 계산될 수 없다. 또한, rand함수의 범위는 0~32767이기 때문에 -1도 나올수가 없고, 최대값을 넘기는 RAND_MAX + 1도 나올 수가 없다. RAND_MAX가 32767이다. 따라서, 34만 가능하다.

8. Consider each of the following code fragments below that could be part of a C++ program. Each fragment contains a call to a standard C/C++ library function. Answer each question in one of the following three ways: • If the code fragment contains a compile-time error, write the word error for the answer. • If the code fragment contains no compile-time errors and you can determine its output at compile-time, provide the fragment's literal output. • If the code fragment contains no compile-time errors but you cannot determine its exact output at compile-time, provide one possible evaluation and write the word example for the answer and provide one possible literal output that the code fragment could produce.

(a) std::cout << sqrt(4.5) << '₩n';

: 2.12132가 정상 출력된다.

(b) std::cout << sqrt(4.5, 3.1) << '₩n';

: 오버로드된 함수에서 2개의 인수를 사용하지 않습니다. 라는 경고가 뜨며 컴파일되지 않는다. 따라서 인수를 두개 사용할 수 없다.

(c) std::cout << rand(4) << '₩n';

: 함수는 1개의 인수를 사용하지 않습니다. 라는 경고가 뜨며 컴파일 되지 않는다. Rand 함수는 인수를 필요로 하지 않기 때문이다.

(d) double d = 16.0; std::cout << sqrt(d) << '₩n';

: 4가 정상적으로 출력된다. D의 제곱근이 출력된다.

(e) std::cout << srand() << '₩n';

: 함수는 0개의 인수를 사용하지 않습니다. 라는 경고가 뜨며 컴파일 되지 않는다. Srand 함수는 1

개의 인수를 받아야하며 std::cout이 가능하지 않아 인수를 채워도 << 연산자 오류가 발생한다.

(f) std::cout << rand() << '₩n';

: 처음 실행했던 난수 (본인의 경우 41)가 출력된다. 재실행하여도 결과는 같다.

(g) int i = 16; std::cout << sqrt(i) << '₩n';

: 4가 정상적으로 출력된다. I 값인 16의 제곱근이기 때문이다.

(h) std::cout << srand(55) << '₩n';

: 오른쪽 피연산자로 'void' 형식을 사용하는 연산자가 없거나 허용되는 변환이 없습니다. Srand 함수가 return값을 갖는 함수가 아니기 때문에 출력할 수 없는 것이다.

(i) std::cout << tolower('A') << '₩n';

: 97이 출력된다. 문자형 A가 소문자문자형 a로 바뀐 뒤, 그에 따른 아스키코드가 출력되는 것이다.

(j) std::cout << exp() << '₩n';

: 오버로드된 함수에서 0개의 인수를 사용하지 않습니다. 라는 경고가 나오며 컴파일 되지 않는다. Exp 함수는 1개의 인수가 필요하다.

(k) std::cout << sqrt() << '₩n';

: (j)와 마찬가지로 오버로드된 함수에서 0개의 인수를 사용하지 않습니다. 라는 경고가 나오며 컴파일 되지 않는다. sqrt 함수는 1개의 인수가 필요하다.

(l) std::cout << toupper('E') << '₩n';

: 69가 출력된다. 문자형 E는 이미 대문자이므로 그대로 E의 아스키코드인 69가 출력된다.

(m) std::cout << toupper('e') << '₩n';

: 69가 출력된다. 문자형 e가 대문자 E로 바뀌어 위와 같이 E의 아스키코드인 69가 출력된다.

(n) std::cout << toupper("e") << '₩n';

: 인수 1을 'const char [2]'에서 'int'로 변환할 수 없습니다.라는 경고가 뜨며 컴파일 되지 않는다. Toupper의 함수의 인수가 문자열인 경우 int형식의 매개변수와 호환되지 않기 때문이다.

(o) std::cout << exp(4.5) << '₩n';

: 90.0171이 출력된다. 지수함수 e의 4.5제곱의 값이 정상 출력되었다.

(p) std::cout << toupper('h', 5) << '₩n';

: 함수는 2개의 인수를 사용하지 않습니다. 라는 경고가 뜨며 컴파일되지 않는다. toupper함수는 1개의 인수를 받는다.

(q) std::cout << ispunct('!') << '₩n';

: 16이 출력된다. ispunct함수는 인수가 숫자, 알파벳을 제외한 문자인지 판단한다. 따라서 참인 경우 0 이외의 값을 return하고 아니라면 0을 return한다.

(r) std::cout << tolower("F") << '₩n';

: (n)과 마찬가지이다. 인수 1을 'const char [2]'에서 'int'로 변환할 수 없습니다.라는 경고가 뜨며 컴파일 되지 않는다. Tolower의 함수의 인수가 문자열인 경우 int형식의 매개변수와 호환되지 않기 때문이다.

(s) char ch = 'D'; std::cout << tolower(ch) << '₩n';

: 100이 출력된다. tolower함수가 문자열 D를 인수로 입력받아 소문자인 d의 아스키코드인 100을 return하여 출력한다.

(t) std::cout << exp(4.5, 3) << '₩n';

: 오버로드된 함수에서 2개의 인수를 사용하지 않습니다. 라는 경고가 뜨며 컴파일 되지 않는다. Exp 함수는 1개의 인수만 사용한다.

(u) std::cout << toupper('7') << '₩n';

: 문자형 7에는 대소문자가 없으므로 변경된게 없이 그대로 7의 아스키코드인 55가 출력된다.

(v) double a = 5, b = 3; std::cout << exp(a, b) << '₩n';

: 마찬가지로 exp 함수는 1개의 인수만 사용하기 때문에 경고 뜨며 컴파일 되지 않는다.

(w) std::cout << exp(3, 5, 2) << '₩n';

: 상동이다. Exp 함수는 1개의 인수만 사용한다. 인수가 필요 이상이면 경고뜨며 컴파일 안된다.

(x) std::cout << tolower(70) << '₩n';

: 102가 출력된다. 아스키코드에서 70과 F가 동일하므로 F의 소문자인 f의 아스키코드 102가 출력되는 것이다.

(y) double a = 5; std::cout << exp(a, 3) << '₩n';

: 마찬가지로 exp 함수는 1개의 인수만 사용하기 때문에 경고 뜨며 컴파일 되지 않는다.

(z) double a = 5; std::cout << exp(3, a) << '₩n';

: 마찬가지로 exp 함수는 1개의 인수만 사용하기 때문에 경고 뜨며 컴파일 되지 않는다.

9. From geometry: Write a computer program that given the lengths of the two sides of a right triangle adjacent to the right angle computes the length of the hypotenuse of the triangle. (See Figure 8.6.) If you are unsure how to solve the problem mathematically, do a web search for the Pythagorean theorem.

: < 코드 >

```
#include <iostream>
#include <cmath>

int main() {
        double a, b, c;
        std::cin >> a >> b;
        c = sqrt(a * a + b * b);
        std::cout << c << std::endl;
}
```

< 실행결과 예시 >

3 4

5

C:₩Users₩Administrator₩source₩repos₩Ex002₩Debug₩Ex002.exe(프로세스 78084개)이(가

) 종료되었습니다(코드: 0개).

이 창을 닫으려면 아무 키나 누르세요...

< 설명 >

함수를 사용하라는 소리가 없어 우선 코드를 최대한 간략하게 작성함. 두 직각변의 길이 a, b를 입력 받고 빗변의 길이인 c를 피타고라스 정리를 이용하여 계산하였다. C는 a제곱더하기b제곱의 제곱근이므로 sqrt함수를 이용하였다.

C. Exercises (Write the questions down on your answer sheet)

(pp. 236-240), Exercises 1-8 (If the code does not work, then explain what is wrong, and correct the code), 9, 10
(write output analysis for all exercises)

1. Is the following a legal C++ program?

D. Additional exercises (Write the questions down on your answer sheet)

D-1. Write a program that reads a series of numbers and calculates the average, geometric mean, and harmonic mean. While () 내에서 cin / standard library 함수 사용

D-2. Write a program to print Fibonacci series. (0, 1, 1, ..., 34) 0번부터 1번부터 노상관

(Use the user-defined function: `int Fibonacci(int n)`)

D-3. Write a program that calculates the real solution of the quadratic equation $ax^2+bx+c=0$

- Read in the values for the parameters a,b,c (type double).

- Then the program should calculate the solution considering the following circumstances:

- a=0 and b=0 ➔ Not a valid equation

- a=0 and b≠0 ➔ x=-c/b

- b^2 -4ac < 0 ➔ Not a Real Solution

- b^2 -4ac >= 0 ➔ $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

standard library 함수 사용

*** Write and test the following functions**

D-4. A function that returns the permutation of n and r. 순열 계산하고 테스트

```
// std::cout << get_int_range(20, 10) << '₩n';
std::cout << get_int_range(5, 5) << '₩n';
std::cout << get_int_range(-100, 100) << '₩n';
```

// 이런시긍로 테스트