

CHAPTER 1

Solutions

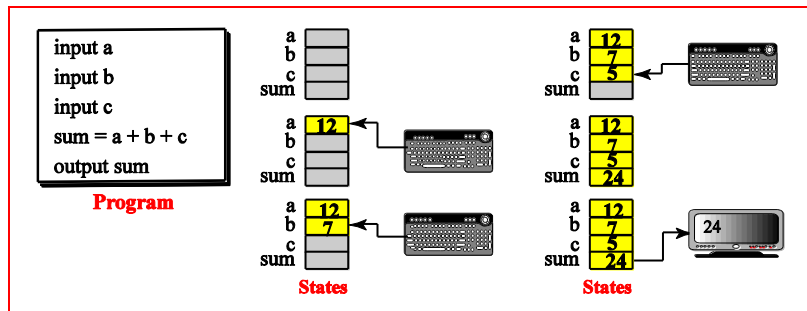
Problems

- 01 [그림 1-11]을 참고하여 절차형 패러다임에서 다음과 같은 코드를 실행했을 때의 메모리 상태를 설명하세요.

```
input a
input b
input c
sum = a + b + c
output sum
```

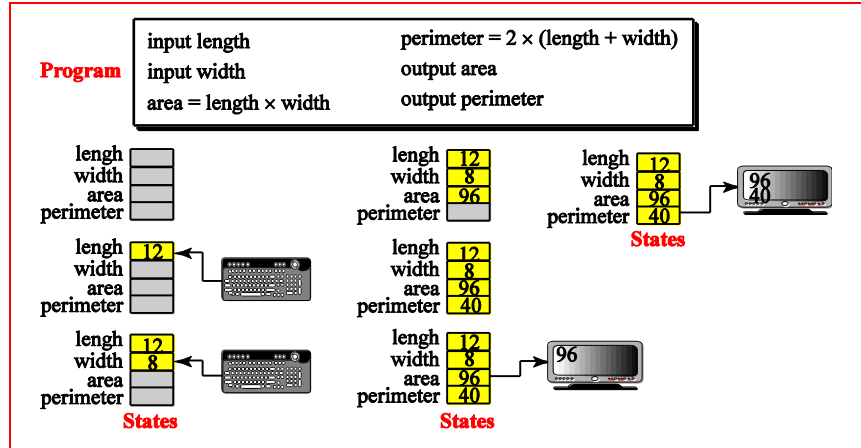
Assume that the value of a, b, c are 12, 7, and 5 respectively. Figure 1.1 shows the state of memory. We have shown memory in two columns to save space.

Figure 1.1 Solution to PR-1



- 02 [그림 1-11]을 참고하여 절차형 패러다임에서 다음과 같은 코드를 실행했을 때의 메모리 상태를 설명하세요. length는 12, width는 8로 가정합니다. 이때 area는 직사각형의 면적을 나타내며, parameter는 직사각형의 둘레를 나타냅니다.

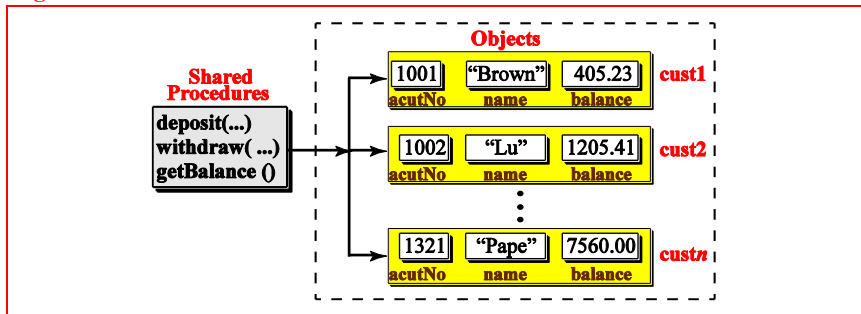
```
input length
input width
area = length × width
parameter = 2 × (length + width)
```



03 객체 지향형 패러다임에서 은행 계좌 객체를 만든다고 가정합니다. [그림 1-12]를 참고하여 데이터로 캡슐화해야 한다고 생각되는 데이터와 절차가 무엇이 있는지 설명하세요.

We assume only three data member for each object (account no, name, and balance). Figure 1.3 shows the situations of shared procedures and objects.

Figure 1.3 Solution to PR-3



04 [그림 1-13]을 참고하여 함수형 패러다임에서 다음 함수의 결과를 설명하세요.

`first (rest (rest (a, b, c)))`

`first (rest (rest (a, b, c))) = first (rest (b, c))`
`first (rest (b, c)) = first (rest (c))`
`first (rest (c)) = first (c)`
`first (c) = c`

05 [그림 1-13]을 참고하여 함수형 패러다임에서 다음 함수의 결과를 설명하세요. 이때 list(...)는 주어 진 것들로 목록을 만든다고 가정합니다.

`list (first (rest (a, b)), first (a, b))`

```
list (first (rest (a, b)), first (a, b)) = list (first (b), a)
list (first (b), a) = list (b, a)
```

06 [그림 1-14]를 참고하여 다음 코드의 결과가 무엇일지 설명하세요.

```
Parent (Benji, Tara)?
GrandParent (Fay, Willi)?
```

The first one is not a fact according to Figure 1.14. The answer is false. For the second query, we need to use the rule defined in Figure 1.14 as shown below:

```
GrandParent (Fay, Willi) = Parent (Fay, Tara) AND Parent (Tara, Willie)
```

Since both of the two facts are true, the result is true.

07 [그림 1-14]를 참고하여 다음 코드의 결과가 무엇일지 설명하세요.

```
Parent (Fay, Tara)?
GrandParent (Tara, Willi)?
```

The first one is a fact according to Figure 1.14. The answer is true. For the second query, we need

```
GrandParent (Tara, Willi) = Parent (Tara, x) AND Parent (x, Willie)
```

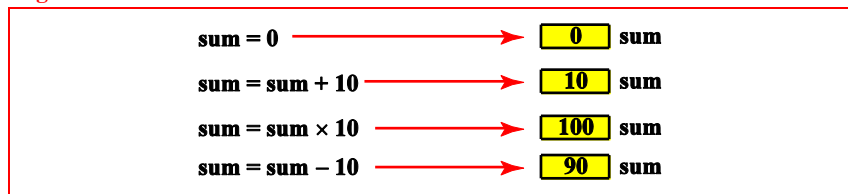
Since x needs to be the same in both facts, we cannot find either of the facts and the result is false.

08 다음 알고리즘이 실행된 뒤에 `sum`에 어떤 값이 들어갈지 설명하세요.

```
sum = 0
sum = sum + 10
sum = sum × 10
sum = sum - 10
```

Figure 1.4 shows the contents of `sum` after each line. Note that the value of `sum` will be changed only if it is at the left-hand side of the assignment. It is copied if it is at the right-hand side of the assignment.

Figure 1.4 Solution to PR-8

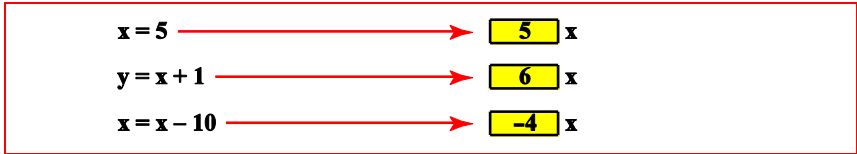


09 다음 알고리즘이 실행된 뒤에 `x`에 어떤 값이 들어갈지 설명하세요.

```
x = 5
x = x + 1
x = x - 10
```

Figure 1.5 shows the contents of x after each line. Note that the value of x will be changed only if it is at the left-hand side of the assignment. It is copied if it is at the right-hand side of the assignment.

Figure 1.5 Solution to PR-9

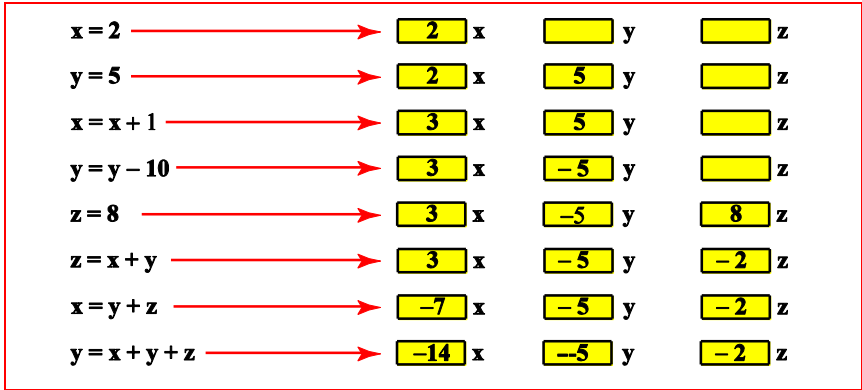


10 다음 알고리즘이 실행된 뒤에 x, y, z에 어떤 값이 들어갈지 설명하세요.

```
x = 2
y = 5
x = x + 1
y = y - 10
z = 8
z = x + y
x = y + z
y = x + y + z
```

Figure 1.6 shows the contents of x, y, and z after each line. Note that the value of x, y, or z will be changed only if they are at the left-hand side of the assignment. The values are copied if they are at the right-hand side of the assignment.

Figure 1.6 Solution to PR-10



11 다음 단위 변환식을 참고하여 센티미터(cm)를 인치(inch) 단위로 변환하는 알고리즘을 설계하세요.

```
1 inch = 2.54 centimeters
```

The following shows the algorithm:

```
Input value in centimeters
Divide value in centimeters by 2.54
Output value in inches
```

Solutions

Problems

01 다음 프로그램에서 오류가 있는 경우, 오류를 찾으세요.

```
1  #include <iostream>
2  using namespace std
3
4  int main ()
5  {
6      cout << 25;
7      cout << first;
8      return 0;
9  }
```

The following are the two errors.

Missing semicolon at the end of line 2.

The variable *first* is not declared (line 7).

02 다음 프로그램에서 오류가 있는 경우, 오류를 찾으세요

```
1  #include <iostream>
2
3  int main ()
4  {
5      cout << 35;
6      cout << 45;
7      return 0;
8  }
```

The following are the errors.

- a. The namespace is not defined.
- b. The values 34 and 45 are printed next to each other as 3445 without space between them.

03 다음 프로그램에서 오류가 있는 경우, 오류를 찾으세요

```
1  #include <iostream>
2  using namespace std
3
```

```

4  int main ();
5  {
6      std :: cout << 35 << endl;
7      std :: cout << 45;
8  }

```

There should not be a semicolon after the header of the *main* function in line 4.

04 다음 프로그램에서 오류가 있는 경우, 오류를 찾으세요.

```

1  using namespace std;
2
3  int main ()
4  {
5      double = 24;
6      cout << double ;
7      return 0
8  }

```

The header `<iostream>` needs to be included before line 1.

05 다음 프로그램에서 오류가 있는 경우, 오류를 찾으세요.

```

1  #include <iostream>
2  using namespace std
3
4  int main ()
5  {
6      int a = 25;
7      a + 30 = a ;
8      cout << a;
9      return 0;
10 }

```

The following are the errors.

- a. Missing semicolon after `std` in line 2.
- b. A literal (30) cannot be on the right of the assignment operator in line 7

06 다음 프로그램에서 오류가 있는 경우, 오류를 찾으세요.

```
1 #include <iostream>
2 using namespace std;
3
4 int main ()
5 {
6     string name;
7     cout << name;
8     return 1;
9 }
```

The following are the errors.

- a. Output will be “garbage” values because *name* was not assigned a value in line 6.
- b. The return statement in line 8 needs to return 0 to indicate success.

07 다음 프로그램에서 오류가 있는 경우, 오류를 찾으세요.

```
1 #include <iostream>
2 using namespace std;
3
4 int main ()
5 {
6     int y = 32.34;
7     cout << y ;
8     return 0;
9 }
```

The variable *y* needs to be assigned an integer, not a double value.

08 다음 프로그램에서 오류가 있는 경우, 오류를 찾으세요

```
1 #include <iostream>
```

```
2 using namespace std;
3
4 int main ()
5 {
6     float y = 32.34;
7     cout << y ;
8     return 0;
9 }
```

There is no compilation error, but the literal 32.34 is better to be 32.34F to match the type (float)

09 다음 프로그램에서 오류가 있는 경우, 오류를 찾으세요.

```
1 #include <iostream>
2 using namespace std;
3
4 int main ()
5 {
6     double y = 32.34;
7     cout << ;
8     return 0;
9 }
```

In line 7, the *cout* object needs a value to print.

CHAPTER 3

Solutions

Problems

01 다음 정수 표현식에서 오류가 발생하는 경우를 찾으세요

- | | |
|----------|----------|
| a. 0723 | b. 0491 |
| b. -0241 | d. -0412 |

- a. 0723 is evaluated as 467 in decimal.
- b. 0491 is not a valid octal integer (9 is not an octal digit).
- c. -0241 is evaluated as -161 in decimal.
- d. -0412 is evaluated as -266 in decimal.

02 다음 정수 표현식에서 오류가 발생하는 경우를 찾으세요

- | | |
|------------|------------|
| a. -0x1A12 | b. -0xH21B |
| c. 0xA2EF | d. 0x4 |

- a. -0x1A12 is evaluated as -6674 in decimal.
- b. -0xH21B is not a valid hexadecimal integer (H is not hex digit)
- c. 0xA2EF is evaluated as 41711 in decimal.
- d. 0x4 is evaluated as 4 in decimal.

03 다음 부동 소수점 표현에서 오류가 발생하는 경우를 찾으세요

- | | |
|------------|------------|
| a. 0.0712F | b. 6.0712 |
| c. 1.234E2 | d. 1123E-2 |

- a. 0.0712F is evaluated as 0.0712F in decimal.
- b. 6.0712 is evaluated as 6.0712
- c. 1.234E2 is evaluated as $1.234 * 10^2$ or 123.4
- d. 1123E-2 is evaluated as $11.23 * 10^{-2}$ or 11.23
- d. 1123E-2 is evaluated as $11.23 * 10^{-2}$ or 11.23

04 다음 표현식의 결과를 예측하세요.

$$3 + 7 * 2$$

$$4 * 2 + 7$$

$$7 / 4 + 3 * 4$$

$$3 + (7 * 2)$$

$$(4 * 2) + 7$$

$$(7 / 4) + 3 * 4$$

$$(3 + 14)$$

$$(8 + 7)$$

$$1 + (3 * 4)$$

$$17$$

$$15$$

$$(1 + 12)$$

$$13$$

05 다음 표현식의 결과를 예측하세요.

$$4 / 3 * 6 / 2 + 8 \% 3$$

$$24 \% 5 + 16 \% 5 * 4 / 2$$

$$(4 / 3) * 6 / 2 + 8 \% 3$$

$$(24 \% 5) + 16 \% 5 * 4 / 2$$

$$1 * (6 / 2) + 8 \% 3$$

$$4 + (16 \% 5) * 4 / 2$$

$$1 * 3 + (8 \% 3)$$

$$4 + (1 * 4) / 2$$

$$(1 * 3) + 2$$

$$4 + (4 / 2)$$

$$(3 + 2)$$

$$(4 + 2)$$

$$5$$

$$6$$

06 $x = 3, y = 5, z = 6$ 이라고 할 때, 다음 각 표현식의 결과를 예측하세요.

$$x - 3 + y * 2 - z / 2$$

$$(x + 2) \% (y * y)$$

$$x - 3 + y * 2 - z / 2$$

$$(3 + 2) \% (5 * 5)$$

$$3 - 3 + (5 * 2) - 6 / 2$$

$$5 \% (5 * 5)$$

$$3 - 3 + 10 - (6 / 2)$$

$$5 \% 25$$

$$(3 - 3) + 10 - 3$$

$$5$$

$$(0 + 10) - 3$$

$$(10 - 3)$$

$$7$$

07 $x = 3, y = 5, z = 6$ 이라고 할 때, 다음 각 표현식의 결과를 예측하세요.

$$(x + 2) \% (y * y)$$

$$y - 2 - x * y$$

08 $x = 3, y = 5, z = 6$ 이라고 할 때, 다음 각 표현식의 결과를 예측하세요.

$z /= x + y + 4;$

$z \% = y * 2 + 3;$



09 다음 표현식의 결과를 예측하세요

<code>2 + 4</code>	<code>3 % 5;</code>	<code>3.2 * 2;</code>	<code>5 + 3.2;</code>
--------------------	---------------------	-----------------------	-----------------------

<code>2 + 4</code>	<code>3 % 5</code>	<code>3.2 * 2</code>	<code>5 + 3.2</code>
6	3	<code>3.2 * 2.0</code>	<code>5.0 + 3.2</code>
		5.2	8.2

10 다음 표현식의 결과를 예측하세요.

<code>'A' - 2</code>	<code>'B' + 3.2;</code>	<code>4.5 + 'D';</code>	<code>'C' * 2;</code>
----------------------	-------------------------	-------------------------	-----------------------

<code>('A') - 2</code>	<code>('B') + 3.2</code>	<code>4.5 + ('D')</code>	<code>('C') * 2</code>
<code>(65 - 2)</code>	<code>(66) + 3.2</code>	<code>4.5 + (68)</code>	<code>(67 * 2)</code>
63	<code>(66.0 + 3.2)</code>	<code>(4.5 + 68.0)</code>	134
	69.2	72.5	

11 다음 중에서 제대로 선언된 변수를 고르세요.

a. `char letter = 'A';`
b. `int first, second;`
c. `double average, tax = 8.5;`
d. `long double 3x;`

a. valid;
b. valid.
c. valid;
d. not valid: 3x is not a valid identifier

12 다음 중에서 제대로 선언된 변수를 고르세요.

a. `char letter, grade;`
b. `int sum = 5;`
c. `double average * 3;`
d. `long double 34.123;`

- a. valid;
- b. valid.
- c. invalid: *3 is not a valid initialization
- d. invalid: there is no identifier

13 다음 코드를 차례차례 실행했을 때, 최종적으로 x, y, z에 어떤 값이 들어가는지 예측하세요.

```
int x = 2, y = 4, z = 5;
x = y - 2;
y += x;
z /= x + y + 4;
z %= y * 2 + 3;
```

int x = 2, y = 4, z = 5;	x : 2	y : 4	z : 5
x = y - 2;	x : 2	y : 4	z : 5
y += x;	x : 2	y : 6	z : 5
z /= x + y + 4;	x : 2	y : 6	z : 0
z %= y * 2 + 3;	x : 2	y : 6	z : 0

14 다음 코드를 차례차례 실행했을 때, 최종적으로 x, y, z에 어떤 값이 들어가는지 예측하세요.

```
double x = 2.5, y = 3.2, z = 5.1;
x = y + 2 ;
y += x - 1.5;
z += x + y * 4;
```

```
x : 5.2
y : 6.9
z : 37.9
```

15 다음 중에서 컴파일 오류를 발생시키는 코드를 모두 고르고 그 이유를 설명하세요.

- | | |
|-------------------|-------------------|
| a. 2 + x = 4; | b. z = z + 2 = 6; |
| c. y = x + 2 % 5; | d. t = z = 4 + 5; |

- a. Error: right operand must be a single variable.
- b. Error: a value may not be assigned to an expression.
- c. No error.
- d. No error.

16 다음 프로그램의 문장이 위부터 아래로 차근차근 실행된다고 할 때, 최종적인 x와 y의 값이

바뀌는지 예측하세요.

```
int x = 2, y = 10;  
y = x;  
x = y;
```

Both *x* and *y* contain 2 because the value of *y* is lost when *x* is assigned to it.

17 다음 프로그램의 문장이 위부터 아래로 차근차근 실행된다고 할 때, 최종적인 *x*와 *y*의 값이 바뀌는지 예측하세요.

```
int x = 2, y = 10;  
int temp;  
temp = x;  
x = y;  
y = temp;
```

The variable *x* contains 10 and the variable *y* contains 2. The values were swapped (exchanged) through the variable *temp* which preserves the value of *y*.

18 다음 코드의 출력 결과를 예측하세요.

```
cout << fixed << setprecision (2) << 124.78560 << endl;  
cout << fixed << setprecision (2) << 0.14 << endl;  
cout << fixed << setprecision (2) << 20 << endl;  
cout << fixed << setprecision (2) << 14767.0 << endl;
```

```
124.79  
0.14  
20  
14767.00
```

19 다음 한글로 표현된 코드를 실제 코드로 하나하나 옮기세요.

```
변수 x의 값을 1만큼 증가시킵니다.  
변수 y의 값을 1만큼 감소시킵니다.  
변수 t의 값을 5.14로 할당합니다.  
변수 first의 값을 변수 second에 복사합니다.
```

20 다음 코드 중 발생할 수 있는 오류를 찾으세요(x, y, z, t는 숫자 자료형이라고 가정).

```
x += 7;  
y *= y;  
z + 6 = z;  
3t = 12;
```

```
x += 7;          no error  
y *= y;          no error
```

z + 6 = z;	error: the left operand must be a single variable
3t = 12;	error: 3t is an invalid variable name

21 x가 bool, y가 char, z가 short, t가 float라고 가정할 때, 다음 표현식에서 발생하는 암묵적 자료형 변환이 무엇인지, 그리고 표현식의 결과가 무엇인지 예측하세요.

```
x + 4
y * 2
z - 24;
t + 23.4;
```

```
x is promoted to int
y is promoted to int
z is promoted to int
t is promoted to double
```


Solutions

Problems

01 초기에 $x = 4$, $y = 0$ 일 때, 다음 코드를 실행한 이후의 x , y 값을 예측하세요.

```
if (x != 0)
{
    y = 3;
}
```

The condition is true, so the body of the *if* statement is executed and the value of y will be changed to 3. The values of x remains unchanged. We have:

$x \rightarrow 4$ $y \rightarrow 3$

02 초기에 $x = 4$, $y = 0$, $z = 2$ 일 때, 다음 코드를 실행한 이후의 x , y , z 값을 예측하세요.

```
if ( z == 0 || y != 0 )
{
    if ( z <= 2 )
    {
        z = 4;
    }
}
else
{
    y = 5;
    z = y + x;
}
```

The condition of outer *if* is (**false** || **false**) which is evaluated to **false**. The *else* statement is executed. Therefore, we have :

$x \rightarrow 4$ $y \rightarrow 5$ $z \rightarrow 9$

03 초기에 $x = 4$, $y = 0$, $z = 2$ 일 때, 다음 코드를 실행한 이후의 x , y , z 값을 예측하세요.

```
if ( z == 2 )
{
    y = 1;
}
```

```
else
{
    x = 3;
}
```

The condition is true, so the *if*-section is executed and the value of y will be changed to 1. The value of x and z remain unchanged. We have:

$$x \rightarrow 4 \qquad y \rightarrow 1 \qquad z \rightarrow 2$$

04 초기에 $x = 4, y = 0, z = 2$ 일 때, 다음 코드를 실행한 이후의 x, y, z 값을 예측하세요.

```
if ( x > y || y < z )
{
    x = 10;
}
```

SPR-3. The condition is (**true** || **true**), which is evaluated to *true*, so the body of *if*-statement is executed and the value of x will be changed to 10. The value of y and z remain unchanged. We have:

$x \rightarrow 10$ $y \rightarrow 0$ $z \rightarrow 2$

05 초기에 `x = true`, `y = false`, `z = true`일 때, 다음 코드를 실행한 이후의 `x`, `y`, `z` 값을 예측하세요

```
if ( x )
{
    if ( y )
    {
        z = false;
    }
    else
    {
        y = true;
    }
}
```

Since `x` is true, we enter the body of the outer *if*, but since `y` is false, the *else* section is executed. The value of `y` is changed to **true**. Therefore, we have

$x \rightarrow \text{true}$ $y \rightarrow \text{true}$ $z \rightarrow \text{true}$

06 초기에 $x = 0, y = 0, z = 1$ 일 때, 다음 코드를 실행한 이후의 x, y, z 값을 예측하세요.

```
switch ( x )
{
    case 0 : x = 2;
            y = 3;
    case 1 : x = 4;
    default : y = 5;
            x = 1;
}
```

The values will be changed in a waterfall. We have the three changes:

After case 0	$x \rightarrow 2$	$y \rightarrow 3$	$z \rightarrow 1$
--------------	-------------------	-------------------	-------------------

After case 1	x → 4	y → 3	z → 1
After default	x → 1	y → 5	z → 1

07 초기에 $x = 2$, $y = 1$, $z = 1$ 일 때, 다음 코드를 실행한 이후의 x , y , z 값을 예측하세요.

```
switch ( x )
{
    case 0 : x = 3;
            y = 2;
    case 1 : x = 2;
    default : y = 3;
            x = 4;
}
```

We enter through the default case.

After default	x → 4	y → 3	z → 1
---------------	-------	-------	-------

08 다음 코드를 switch 조건문을 사용하는 형태로 변경하세요.

```
if ( x == 2 )
{
    x++;
}
else if ( x == 3 )
{
    x--;
}
else
{
    cout << "End!";
}
```

The following shows the code.

```
switch ( x )
{
    case 2: x++;
            break;
    case 3: x--;
            break;
    default: cout << "End!";
}
```

09 다음 코드를 switch 조건문을 사용하는 형태로 변경하세요.

```
if ( ch == 'A' || ch == 'a' )
    countA++;
else if ( ch == 'E' || ch == 'e' )
    countE++;
else if ( ch == 'I' || ch == 'i' )
    countI++;
else
    cout << "Error--Not A, E, or I " << endl;
```

The following shows the code.

```
switch ( ch )
{
    case 'A':
    case 'a':    countA++;
                break;

    case 'E':
    case 'e':    countE++;
                break;

    case 'I':
    case 'i':    countI++;
                break;

    default:     cout << "Error--Not A, E, or I "<< endl;
}
}
```

Solutions

Problems

01 다음 코드에서 while 반복문 내부의 코드가 몇 번 실행될지 예측하세요. 또한 코드가 모두 종료된 뒤 x의 값을 예측하세요.

```
int x = 5;
while ( x < 9 )
{
    x++;
}
```

Four times (when $x = 5, 6, 7,$ and 8). When we come out of the loop, the value of x is 9 .

02 다음 코드에서 while 반복문 내부의 코드가 몇 번 실행될지 예측하세요. 또한 코드가 모두 종료된 뒤 x의 값을 예측하세요.

```
int x = 5;
while ( x < 11 )
{
    x += 2;
}
```

Three times (when $x = 5, 7, 9$). When we come out of the loop, x is 11 .

03 다음 코드에서 while 반복문 내부의 코드가 몇 번 실행될지 예측하세요. 또한 코드가 모두 종료된 뒤 x의 값을 예측하세요.

```
int x = 7;
while ( x < 3 )
{
    x++;
}
```

The condition of the while loop is false the first time and the loop is skipped. The body of the loop is never executed. This means that that value of x is always 7 .

04 다음 코드에서 while 반복문 내부의 코드가 몇 번 실행될지 예측하세요. 또한 코드가 모두 종료된 뒤 x의 값을 예측하세요.

```
int x = 7;
int y = 5;
while ( x < 11 && y > 3 )
{
```

```
    x++;  
    y--;  
}
```

The body is executed twice. At the end, x is 9 and y is 3 (which terminates the loop).

05 다음 코드에서 while 반복문 내부의 코드가 몇 번 실행될지 예측하세요. 또한 코드가 모두 종료된 뒤 x의 값을 예측하세요.

```
int x = 7;  
while ( false )  
{  
    x++;  
}
```

The loop is never executed because the loop condition is false when it is first tested. The variable x is unchanged.

06 다음 코드에서 while 반복문 내부의 코드가 몇 번 실행될지 예측하세요. 또한 코드가 모두 종료된 뒤 x의 값을 예측하세요.

```
int x = 10;  
while ( true )  
{  
    x--;  
}
```

The loop is executed an infinite number of times because the while condition is not changed. The variable x will become infinity.

07 다음 코드에서 do-while 반복문 내부의 코드가 몇 번 실행될지 예측하세요. 또한 코드가 모두 종료된 뒤 x의 값을 예측하세요.

```
int x = 5;  
do  
{  
    x -= 2;  
} while (x != 4);
```

It will be executed an infinite number of times because x never becomes 4. (Its values are 5, 3, 1, -1, ...) The value of x is undeterminable.

08 다음 코드에서 do-while 반복문 내부의 코드가 몇 번 실행될지 예측하세요. 또한 코드가 모두 종료된 뒤 x의 값을 예측하세요.

```
int x = 15;  
do  
{  
    x -= 2;  
    if (x < 9)  
        break;  
} while (true);
```

It is executed four times. The value of x is 7 at the end of the loop.

09 다음 코드의 실행 결과를 예측하세요

```
int x = 13;
while (x > 7)
{
    cout << x << " ";
    x--;
}
```

The following is printed: 13 12 11 10 9 8

10 다음 코드의 실행 결과를 예측하세요.

```
for (int x = 13; x > 7; x--)
{
    cout << x << " ";
}
```

The following is printed.

13 12 11 10 9 8

11 다음 do-while 반복문을 while 반복문으로 변경하세요.

```
int x = 13;
do
{
    cout << x << " ";
    x--;
} while (x > 7);
```

The following is printed.

13 12 11 10 9 8 7

12 다음 while 반복문을 for 반복문으로 변경하세요.

```
int x = 10;
do
{
    cout << x << end;
    x++;
} while (true);
```

The following shows the *while* loop.

```
int x = 10;
while (true)
{
    cout << x << end;
    x++;
}
```

13 다음 for 반복문을 while 반복문으로 변경하세요

```
int x = 10;
while (x < 20)
{
    cout << x << endl;
    x++;
}
```

The following shows the *for* loop.

```
for (x = 10, x < 20, x++)
{
    cout << x << endl;
}
```

14 다음 코드의 실행 결과를 예측하세요.

```
int x = 13;
while (x > 7)
{
    cout << x << "    ";
    x--;
}
```



Solutions

Problems

01 다음 함수 선언에서 시그니처와 관계된 요소를 찾으세요.

```
int firstFunction (int x, float y, int z);  
void secondFunction (int x, boolean y);  
void thirdFunction (double x, double y);  
void fourthFunction ();
```

The following shows the signature of each function.

```
firstFunction (int, float, int)  
secondFunction (int, boolean)  
thirdFunction (double, double)  
fourthFunction ()
```

02 다음 두 함수 선언이 오버로딩될 수 있는지 확인하세요.

```
int fun (int x, int y);  
void fun (int a, int b);
```

The names are the same and the signatures are identical: *fun* (*int*, *int*). They are not overloaded. A compiler error will result because the compiler thinks that the second function is the re-declaration of the first.

03 다음 두 함수 선언이 오버로딩될 수 있는지 확인하세요.

```
int fun (int x, int y);  
void fun (float a, float b);
```

The names are the same, but the signatures of the two functions are different : *fun* (*int*, *int*) and *fun* (*float*, *float*). They are overloaded.

04 다음 두 함수 선언이 오버로딩될 수 있는지 확인하세요.

```
int fun (int x, int y, int z);  
float fun (int a, int b);
```

The names are the same, but the signatures are different: *fun* (*int*, *int*, *int*) and *fun* (*int*, *int*). They are overloaded.

05 다음 두 함수 선언이 오버로딩될 수 있는지 확인하세요.

```
int functionOne (int x, int y);
```

```
int functionTwo (int a, int b);
```

The two function have different names. The first condition for overloading is not met. These two function are completely two separate functions (not overloaded).

06 다음 중에서 정상적인 함수 선언을 고르세요.

```
a. float one (int a, int b);  
b. boolean two (int a, b);  
c. float (int a, int b);  
d. void three (void);  
e. int ();
```

- a. It is valid.
- b. It is not valid: no type for parameter *b*.
- c. It is not valid because it does not contain a function name.
- d. It is valid.
- e. It is not valid because it does not contain a function name.

Find any errors in the following function definition:

07 다음 프로그램에서 발생할 수 있는 오류를 찾으세요.

```
void one (int a)  
{  
    return a;  
}
```

A void function cannot return a value.

08 다음 프로그램에서 발생할 수 있는 오류를 찾으세요.

```
int two (int a)  
{  
    int b = a * a;  
}
```

The function header indicates that it will return an integer. Nothing is returned.

09 다음 프로그램에서 발생할 수 있는 오류를 찾으세요.

```
int three (int a, int b)  
{  
    c = a * b;  
    return c;  
}
```

The function will not compile because the variable *c* is not declared.

10 다음 프로그램에서 발생할 수 있는 오류를 찾으세요.

```
void one ()
```

```

{
    cout << "In One" << endl;
    void two ()
    {
        cout << "In Two" << endl;
        return;
    }
    return;
}

```

A function cannot be defined inside another function. The function **two** has been defined inside the body of function **one**.

11 다음 프로그램에서 발생할 수 있는 오류를 찾으세요.

```

int wrong (int x)
{
    double x = 2.7;
    return x;
}

```

The function header defines *x* as an integer. The body redefines it as a double. The declaration of the double *x* shadows the declaration of the parameter *x* and a compilation error occurs.

12 다음 함수를 호출했을 때 무엇을 리턴하는지 예측하세요.

```

int test ()
{
    return 3.25;
}

```

The integer value 3 will be returned. The double value 3.25 is changed to integer value 3.

13 다음 함수를 호출했을 때 무엇을 리턴하는지 예측하세요.

```

char test ()
{
    return 67;
}

```

The character C will be returned.

14 다음 함수를 호출했을 때 무엇을 리턴하는지 예측하세요.

```

double test ()
{
    return 9;
}

```

The value 9.0 will be returned.

15 다음 프로그램의 실행 결과를 예측하세요.

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

int x;

int main ()
{
    cout << x;
    return 0;
}
```

The integer 0 is printed. The global integer x is initialized to the default value (0).

16 다음 프로그램의 실행 결과를 예측하세요.

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

int x;

int main ()
{
    int x;
    cout << x;
}
```

A garbage value (leftover value in the address (x) is printed because the local variables shadows the global variable. The *main* function sees only the local variable x.

Solutions

Problems

01 다음과 같은 생성자가 있다고 할 때 어떤 것이 매개변수 생성자인지, 기본 생성자인지, 복사 생성자인지 구분해 보세요.

```
Fun ();  
Fun (int x);  
Fun (const Fun& fun);
```

The following shows the answers:

```
Fun ();                // Default constructor  
Fun (int x);           // Parameter constructor  
Fun (const Fun& fun);  // Copy constructor
```

02 다음과 같이 Rectangle 클래스의 생성자를 선언하면 오류가 발생합니다. 왜 발생하는지 설명해 보세요.

```
int Rectangle (int length, int height);
```

The constructor cannot have a return value.

03 다음과 같이 Rectangle 클래스의 생성자를 선언하면 오류가 발생합니다. 왜 발생하는지 설명해 보세요.

```
int ~Rectangle ();
```

The destructor cannot have a return value.

04 다음과 같은 생성자를 할당하는 코드 대신 초기화 리스트를 사용하는 코드로 변경하세요.

```
Rectangle :: Rectangle (int len, int wid)  
{  
    length = len;  
    height = wid;  
}
```

The following shows the answer.

```
Rectangle :: Rectangle (int len, int wid)
```

```

: length (len), width (wid)
{
}

```

05 다음과 같은 클래스 정의가 있을 때

```

class Sample
{
    private:
        int x;
    public:
        int getX () const;
};

```

다음 인스턴스화 코드 중 오류가 발생하는 것을 고르세요.

```

Sample first (4);
Sample second ( );
Sample third;

```

The following shows the answer:

```

Sample first (4);           // Error. Class has no parameter constructor
Sample second ( );         // Wrong syntax for a default constructor
Sample third;               // Correct default constructor

```

06 다음 클래스 정의에서 발생하는 오류를 찾으세요

Braces missing after class name and at end. Declaration of a class also needs a semicolon at the end. The following shows the correct class definition.

```

class First
{
    private:
        double x;
        double y;
    public:
        double getX () const;
        double getY () const;
};

```

// Brace needed

// Brace and semicolon needed

07 x가 클래스의 데이터 멤버라고 할 때, 다음 클래스의 멤버 함수 정의에서 발생하는 오류를 찾으세요.

```

Fun :: int getX ( ) const

```

```
{
    return x;
}
```

The return type must be before the Class name as shown below:

```
int Fun :: getX ( ) const
{
    return x;
}
```

08 다음 중에서 Sample 클래스의 접근자 함수를 올바르게 정의한 것을 찾으세요.

The left function is a proper accessor function; the right function should not be used because it is not supposed to change the host object.

09 다음 Sample 클래스의 멤버 함수 중에서 설정자 멤버 함수가 될 수 있는 것은 어떤 것인가요?

The left function cannot be a mutator function because of the constant modifier. The right function can be a mutator function.

10 다음과 같은 클래스 정의가 있을 때, 두 멤버 함수의 정의를 작성하세요.

The following shows the definition of the parameter constructor and copy constructor.

```
Fun :: Fun (int xx)
: x (xx)
{
}

Fun :: Fun (const Fun& f)
: x (f.x)
{
}
```

11 10번에서 만든 매개변수가 있는 생성자와 복사 생성자를 사용하는 애플리케이션 코드를 작성해주세요.

```
int main ()
{
    Fun fun1 (5);
    Fun fun2 (fun1);
}
```

Solutions

Problems

01 다음과 같은 배열이 주어졌을 때, 홀수 위치에 있는 배열의 요소만 출력하는 코드를 작성하세요.

```
int arr [10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

SPR-1. The following is the code:

```
for (int i = 0; i < 10; i = i + 2)
{
    cout << arr[i] << " ";
}
```

02 다음과 같은 배열이 주어졌을 때, 앞의 요소 5개를 출력하는 코드를 작성하세요.

```
int arr [12] = {0, 10, 20, 39, 40, 50, 60, 70, 80, 90};
```

SPR-2. The following is the code:

```
for (int i = 0; i < 5; i++)
{
    cout << arr[i] << " ";
}
```

03 다음과 같은 배열 선언이 있을 때, 짝수 인덱스의 요소는 0으로, 홀수 인덱스의 요소는 1로 채우는 코드를 작성하세요.

```
int arr [10];
```

SPR-3. The following is the code:

```
for (int i = 0; i < 10; i++)
{
    if (i % 2 == 0)
    {
        arr[i] = 0;
    }
    else
    {
        arr[i] = 1;
    }
}
```

04 다음과 같은 배열 선언이 있을 때, 앞의 요소 5개는 정수 5, 나머지는 정수 10으로 채우는 프로그램을 만드세요.

```
int arr [10];
```


SPR-4. The following is the code:

```
for (int i = 0; i < 10; i++)
{
    if (i < 5)
    {
        arr[i] = 5
    }
    else
    {
        arr[i] = 10;
    }
}
```

05 다음 코드의 실행 결과를 예측하세요.

```
int arr [10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
for (int i = 0; i < 5; i++)
{
    cout << arr [i * 2] << " ";
}
```

SPR-5. The following is the result

0 2 4 6 8

06 매개변수로 arr1과 arr2라는 정수 배열을 전달받고, 두 배열의 요소들이 같은지 확인한 뒤 true 또는 false를 리턴하는 함수를 만드세요. 두 배열의 크기는 동일하다고 가정합니다.

```
int arr [10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
for (int i = 0; i < 5; i++)
{
    arr [i] = arr [i * 2 + 1];
    cout << arr [i] << endl;
}
```

SPR-6. The following is the result:

1 3 5 7 9

07 매개변수로 arr1과 arr2라는 정수 배열을 전달받고, arr1의 모든 요소를 2배해서 arr2에 할당하는 함수를 만드세요. 두 배열의 크기는 동일하다고 가정합니다.

```
int arr1 [5] = {0, 1, 2, 3, 4};
int arr2 [5];
for (int i = 0; i < 5; i++)
{
    arr2 [i] = arr1 [i];
    cout << arr1 [i] + arr2 [i] << endl;
}
```

SPR-7. The following is the result:

0 2 4 6 8

08 매개변수로 크기 n, 정수 배열 arr을 전달받고, 배열의 요소가 오름차순으로 정렬되었는지

확인하고 true 또는 false를 리턴하는 함수를 만드세요.

```
int arr [5] = {0, 1, 2, 3, 4};
for (int i = 1; i <= 4; i++)
{
    cout << arr [i] << " ";
}
cout << endl;
```

SPR-8. The following is the result:

```
1 2 3 4
```

09 매개변수로 크기 n, 정수 배열 arr, 정수 value를 전달받고, 배열에 정수 value가 있는지 확인하고 true 또는 false를 리턴하는 함수를 만드세요.

```
int arr1 [10] = {0, 10, 20, 39, 40, 50, 60, 70, 80, 90};
```

SPR-9. The following is the code:

```
int arr2 [10] = {0};
for (int i = 0, i < 10; i++,)
{
    arr2 [i] = arr1 [10 - i];
}
```

10 매개변수로 크기 size, 정수 배열 arr, 인덱스 index1과 index2를 전달받고, 배열의 index1과 index2 위치의 요소를 스왑하는 함수를 만드세요. 매개변수로 전달받은 인덱스가 유효한 범위에 있는지 확인하기 바랍니다.

```
bool cmpArys (int ary1 [ ], int ary2[ ], int size)
{
    for (int i = 0; i < size; i++)
    {
        if (ary1[i] != ary2[i])
        {
            return false;
        }
    }
    return true;
}
```

11 매개변수로 크기 size, 정수 배열 arr, 인덱스 index를 전달받고, 인덱스 위치의 요소를 제거하는 함수를 만드세요. 매개변수로 전달받은 인덱스가 유효한 범위에 있는지 확인하기 바랍니다.

```
void doubleArr (int arr1[ ], int arr2[ ], int n)
{
    for (int i = 0; i < n; i++)
    {
        arr2[i] = arr1[i] * 2;
    }
}
```

```

    }
    return;
}

```

12 매개변수로 크기 size, 정수 배열 arr, 인덱스 index, 정수 data를 전달받고, 인덱스 위치에 정수 data를 삽입하는 함수를 만드세요. 매개변수로 전달받은 인덱스가 유효한 범위에 있는지 확인하기 바랍니다.

```

bool seqArys (int arr[ ], int n)
{
    for (int i = 1; i < size; i++)
    {
        if (arr [i] < arr [i - 1])
            return false;
    }
    return true;
}

```

13 용적이 10인 2개의 배열을 선언하고, 회사 직원의 이름과 봉급을 평행하게 갖게 초기화하세요.

```

string names [10] = {"Tom", "Dick", "Harry", "Mary",
                    "Pat", "Jane", "Joy", "Sam", "Betty", "Peter"};
double salaries [10] = {1050, 1700, 1450, 1675, 2120,
                        1150, 1475, 990, 1600, 2500};

```

14 table이라는 이름의 n행 3열을 갖는 2차원 배열의 요소를 출력하는 함수를 만드세요. 매개변수로 전달받은 r이 유효한 값인지 확인하기 바랍니다.

```

void printRow (int table [ ][3], int n, int r)
{
    assert (r > 0 || r < n);
    for (int col = 0; col < 3; col++)
    {
        cout << table [r] [col] << "    ";
    }
    cout << endl;
    return;
}

```

15 table이라는 이름의 n행 3열을 갖는 2차원 배열에서 c열의 요소를 출력하는 함수를 만드세요. 매개 변수로 전달받은 c가 유효한 값인지 확인하기 바랍니다.

```

void printRow (int table [ ][3], int n , int c)
{
    assert (c > 0 || c < 3);
    for (int row = 0; row < n; row++)

```

```
{  
    cout << table[row][c] << endl;  
}  
return;  
}
```

Solutions

Problems

01 다음 코드의 실행 결과를 예측하세요.

```
int x = 10;  
int& y = x;  
cout << x << " " << y;
```

The following shows the output. We print the value of variable x directly and through the reference variable.

```
10 10
```

02 다음 코드의 실행 결과를 예측하세요.

```
int x = 100;  
int& y = x;  
int& z = x;  
cout << x << " " << y << " " << z;
```

The following shows the output. We print the value of variable x directly and through two reference variables.

```
100 100 100
```

03 다음 코드에서 잘못된 부분을 찾으세요.

```
int x = 1;  
int y = 2;  
int& z = x;  
z = y;  
cout << x << " " << y << " " << z;
```

The following shows the output. Although variable z is a reference to x, we copy the value of y (which is 2) to variable z which is a reference to x. The value 2 can be reached through x, through z (as a reference to x), and through y, which holds this value.

```
2 2 2
```

04 다음 코드에서 잘못된 부분을 찾으세요.

```
int x = 1000;  
int& y = 2000;
```

The first statement is valid. The second statement is not valid; we cannot bind a value to a reference variable.

05 다음 코드에서 잘못된 부분을 찾으세요.

```
const int x = 100;
double& y = x;
```

The variable *x* is of type *int*. The variable *y* is a reference (alias name) to a variable of type *double*. We cannot make *y* to be bound to *x*.

06 다음 코드의 실행 결과를 예측하세요.

```
int x = 10;
int* y = &x;
cout << x << " " << *y;
```

The following shows the output. We are accessing the value of variable *x* in two ways: first directly and second indirectly through pointer variable *y* (by using the dereference operator).

```
10 10
```

07 다음 코드의 실행 결과를 예측하세요.

```
int x = 100;
int* y = &x;
int* z = &x;
cout << x << " " << *y << " " << *z
```

The following shows the output. We access the value of variable *x* directly, through the pointer *y*, and through pointer *z*.

```
100 100 100
```

08 다음 코드의 잘못된 부분을 찾으세요.

```
const int x = 100;
double* y = &x;
```

The address of an integer cannot be assigned to a pointer to double.

09 다음 코드의 잘못된 부분을 찾으세요.

```
const int x = 100;
int& y = x;
```

We get a compilation error. The variable *x* is of type *const int*; the variable *y* is of type reference to *int*. We cannot bind *x* to *y* because the value of *x* cannot be changed. The compiler does not allow this binding because it means we can change the value *x* through *y*.

10 다음 코드의 잘못된 부분을 찾으세요.

```
int* x = 25;
```

A literal cannot be assigned to a pointer.

11 다음 코드의 실행 결과를 예측하세요.

```
int sample [5] = {0, 10, 20, 30, 40};  
cout << *(sample + 2);
```

We are accessing the third element of the array using the pointer arithmetic. If *sample* is pointing to the element with value 0, then *sample* [2] is pointing to the element with value 20. The following is printed.

20

12 다음 코드의 실행 결과를 예측하세요.

```
int sample [5] = {5, 10, 15, 20, 25};  
cout << *sample + 2 << endl;  
cout << *(sample + 2);
```

The second line accesses the value of the first element, which is 5, adds 2 to it, and prints the result, which is 7. The third line access the third element (at index 2), which is 15 and prints it. The output is shown below:

7
15

13 다음 프로그램의 실행 결과를 예측하세요.

```
#include <iostream>  
using namespace std;  
  
void fun (int* x)  
{  
    cout << *(x + 2);  
}  
  
int main ()  
{  
    int sample [5] = {0, 10, 20, 30, 40};  
    fun (sample);  
    return 0;  
}
```

The identifier *sample* as the address to the first element of the array is passed to the function *fun* and is bound to variable *x*. This means the variable *x* is playing the same role as the identifier *sample*. It can print the value of all elements of the array. In other words, *x* is pointing to the *sample* [0] and (*x* + 2) is pointing to *sample* [2]. The following is printed.

14 다음 문장을 실행했을 때 힙에 어떤 형태로 객체가 생성될지 그림으로 그리세요.

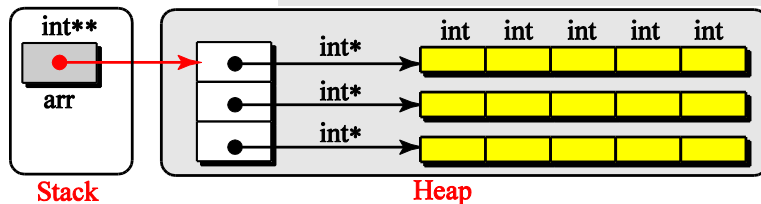
What is created in the heap from the following statement? Draw a picture to show the object in the heap.

```
int** arr = new int* [3];
```

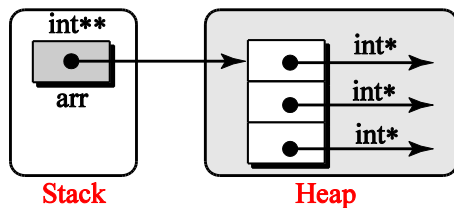
An array of three pointers to integers as shown below:

15 다음 문장을 실행했을 때 힙에 어떤 형태로 객체가 생성될지 그림으로 그리세요.

```
int** arr = new int* [3];  
for (int i = 0; i < 3; i++)  
{  
    arr[i] = new int [5];  
}
```



16 15번 코드로 만든 2차원 배열에 사용자로부터 키보드 입력을 받아 값을 채우는 코드를 작성하세요.



```
for (int row = 0; row < 3; row++)  
{  
    for (int col = 0; col < 5; col++)  
    {  
        cout << "Enter next value: ";  
        cin >> arr [row][col];  
    }  
}
```



```
}
```

17 15번 코드로 만든 2차원 배열을 delete 연산자로 어떻게 힙에서 제거하는지 코드를 작성하세요.

```
for (int i = 0; i < 3; i++)  
{  
    delete [ ] arr [i];  
}  
delete [ ] arr;
```

Solutions

Problems

01 아래의 C 문자열을 활용하는 코드를 같은 결과가 나오도록 C++ 문자열을 이용해서 코드를 작성하세요.

```
const char* str = "This is a string.";
cout << strlen (str) << endl;
```

SPR-1. The following shows the C++ code.

```
string strg ("This is a string.") ;
cout << strg.length () << endl;
```

02 아래의 C 문자열을 활용하는 코드를 같은 결과가 나오도록 C++ 문자열을 이용해서 코드를 작성하세요.

```
const char* str1 = "This is a string.";
const char* str2 = "This is another one.";
cout << strcmp (str1, str2) << endl;
```

SPR-2. The following shows the C++ code.

```
string strg1 ("This is a string.");
string strg2 ("This is another one.");
cout << strg1.compare (strg2) << endl;
```

03 아래의 C 문자열을 활용하는 코드를 같은 결과가 나오도록 C++ 문자열을 이용해서 코드를 작성하세요.

```
char str1 [ ] = "This is the first string.";
const char* str2 = "Here is another one.";
strcpy (str1, str2);
cout << str1 << endl;
```

SPR-3. The *strcpy* function for C strings can be simulated using the *assign* member function in C++. The following code has the same result.

```
string strg1 ("This is the first string.");
string strg2 ("Here is another one.");
strg1.assign (strg2);
cout << strg1 << endl;
```

04 아래의 C 문자열을 활용하는 코드를 같은 결과가 나오도록 C++ 문자열을 이용해서 코드를

작성하세요.

```
char str1 [ ] = "This is the first string.";
const char* str2 = "Here is another one.";
strncpy (str1, str2, 4);
cout << str1 << endl;
```

SPR-4. The *strncpy* function cannot be simulated using the *assign* member function in C++ because the *assign* function erases the whole destination string before copying the source. The *strncpy*, however, can be simulated using the *replace* member function as shown below.

```
string strg1 ("This is the first string.");
string strg2 ("Here is another one.");
strg1.replace (0, 4, strg2, 0, 4);
cout << strg1 << endl;
```

05 아래의 C 문자열을 활용하는 코드를 같은 결과가 나오도록 C++ 문자열을 이용해서 코드를 작성하세요.

```
char str1 [40] = "The time has come. ";
const char* str2 = "Are you ready?";
strcat (str1, str2);
cout << str1 << endl;
```

SPR-5. The *strcat* function can be simulated in C++ using the *append* member function.

```
string strg1 ("The time has come. ");
string strg2 ("Are you ready?");
strg1.append (strg2);
cout << strg1 << endl;
```

06 아래의 C 문자열을 활용하는 코드를 같은 결과가 나오도록 C++ 문자열을 이용해서 코드를 작성하세요.

```
char str1 [40] = "The time has changed. ";
const char* str2 = "Do you know? My dear friend.";
strncat (str1, str2, 12);
cout << str1 << endl;
```

SPR-6. The *strncat* function can be simulated in C++ using the *append* member function.

```
string strg1 ("The time has changed. ");
string strg2 ("Do you know? My dear friend.");
strg1.append (strg2, 0, 12);
cout << strg1 << endl;
```

07 아래의 C 문자열을 활용하는 코드를 같은 결과가 나오도록 C++ 문자열을 이용해서 코드를 작성하세요

```
char str [ ] = "This is a long string.";
```

```
*strchr (str, 's') = 'S';
*strchr (str, 's') = 'S';
cout << str << endl;
```

SPR-7. The following code uses the *find* function in C++ to achieve the same code.

```
string strg("This is a long string.");
strg [strg.find('s')] = 'S';
strg [strg.rfind('s')] = 'S';
cout << strg << endl;
```

08 다음 코드는 C 문자열의 첫 번째 문자를 제거하는 코드입니다. 같은 결과가 나오도록 C++ 문자열을 활용해 코드를 작성하세요.

```
const char* str = "This is a long string.";
char* p = strstr (str, "is");
cout << *p << endl;
```

SPR-8. The following code fragment uses the *find* function to do the same.

```
string strg ("This is a long string.");
int loc = strg.find ("is");
cout << strg [loc] << endl;
```

09 다음 코드는 C 문자열의 마지막 문자를 제외한 다른 모든 문자를 제거하는 코드입니다. 같은 결과가 나오도록 C++ 문자열을 활용해 코드를 작성하세요.

```
const char* str = "ABCDEFGH";
str = str + 1;
cout << str << endl;
```

SPR-9. The following code uses the *erase* function to do the same..

```
string strg ("ABCDEFGH");
strg.erase (0, 1);
cout << strg << endl;
```

10 다음과 같이 C++ 문자열을 활용하는 코드를 같은 결과가 나오도록 C 문자열로 코드를 작성하세요

```
const char* str = "ABCDEFGH";
str = str + strlen (str) - 1;
cout << str << endl;
```

SPR-10. The following code uses the C++ string to delete all but the last character in a string.

```
string strg ("ABCDEFGH");
strg.erase (0, strg.length () - 1);
cout << strg << endl;
```

11 C++ 문자열을 절반으로 나누어서 동일한 크기의 두 문자열로 나누는 코드를 작성하세요. 만약 문자 수가 홀수라면, 앞부분 문자열 끝에 빈 문자를 하나 추가하세요.

```
string strg;
cout << "Enter a string: " ;
getline (cin, strg);
if (strg.size () % 2 == 1)
{
    strg.push_back(' ');
}
int count = strg.size() / 2;
string strg1 (strg, 0, count) ;
string strg2 (strg, count, count);
cout << strg1 << endl;
cout << strg2 << endl;
```

12 C++ 문자열의 처음 4문자와 마지막 4문자를 추출하는 코드를 작성하세요. 이어서 추출한 문자열을 출력하세요.

```
string strg;
cout << "Enter a string: " ;
getline (cin, strg);
string strg1 (strg, 0, 4) ;
string strg2 (strg, strg.size() - 4);
cout << strg1 << endl;
```

CHAPTER 11

Solutions

Problems

01 다음과 같이 2개의 클래스를 정의했을 때, 위임(delegation)을 사용해서 두 클래스의 set 함수와 print 함수를 정의해주세요.

```
// Declaration of the class First
class First
{
    private:
```

```

    int a;
public:
    void set (int a);
    void print () const;
};
// Declaration of the class Second
class Second : public First
{
private:
    int b;
public:
    void set (int a, int b);
    void print () const;
};

```

```

// Definitions for the first class
void First :: set (int aa)
{
    a = aa;
}
void First :: print () const
{
    cout << a << endl;
}
// Definitions for the second class
void Second :: set (int aa, int bb)
{
    First :: set (aa);
    b = bb;
}
void Second :: print () const
{
    First :: print ();
    cout << b << endl;
}

```

02 다음과 같은 First 클래스의 정의가 있을 때, First 클래스를 public 상속하는 Second 클래스를 만 들었다고 합시다. Second 클래스에서 one, two, set에 대한 접근성(접근 할 수 있는지 없는지)이 어떻게 되는지 설명하세요.

```

class First
{
private:
    int one;
protected:
    int two;
public:
    void set (int one, int two);
}

```

```
}
```

The member *one* is *hidden* in class *Second*. The member *two* is *private* in class *Second*. The member *set* is *public* in class *Second*.

03 다음과 같은 3개의 클래스 정의가 있을 때 생성자와 print 함수를 정의하세요.

```
// Definition of class First
class First
{
    private:
        int a;
    public:
        First (int a);
        void print ( ) const;
};

// Definition of class Second
class Second : public First
{
    private:
        int b;
    public:
        Second (int a, int b);
        void print ( ) const;
};

// Definition of class Third
class Third : public Second
{
    private:
        int c;
    public:
        Third (int a, int b, int c);
        void print ( ) const;
};
```

```
// Definition of constructor and print function for class First
First :: First (int aa)
:a (aa)
{
}
void First :: print ( ) const
{
    cout << a << endl;
}

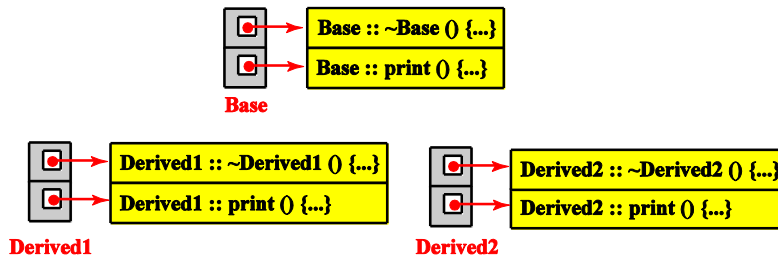
// Definition of constructor and print function for class Second
Second :: Second (int aa, int bb )
: First (aa), b (bb)
{
}
void Second :: print ( ) const
{
    First :: print ();
}
```

```
        cout << b << endl;
    }
    // Definition of constructor and print function for class Third
    Third :: Third (int aa, int bb, int cc)
    : Second (aa, bb), c (cc)
    {
    }
    void Third:: print () const
    {
        Second :: print ();
        cout << c << endl;
    }
```


Solutions

Problems

01 Base 클래스와 Base 클래스를 상속받는 Derived1 과 Derived2 클래스가 있다고 합시다. 이 클래스를 다형성 관계로 활용하려고 합니다. 각 클래스에 가상 print 함수와 가상 소멸자가 있는 경우의 가상 테이블을 그림으로 표현하세요.



02 다음과 같은 두 클래스가 있다고 합시다. 두 클래스가 다형성을 갖게 스택 메모리 위에 인스턴스화하는 코드를 작성하세요. 다형성을 활용한다는 것은 부모 클래스에 대한 포인터를 사용한다는 것입니다. 다만 객체의 실체를 스택 메모리에 만들어서 활용하라는 것이 문제의 포인트입니다.

```
int main ( )
{
    // Instantiation of pointer and two class objects
    First* ptr;
    First first (5);
    Second second (5, 7);
    // Let the pointer point to the first object
    ptr = &first;
    ptr -> print ();
    // Let the pointer point to the second object
    ptr = &second;
    ptr -> print();
    return 0;
}
```

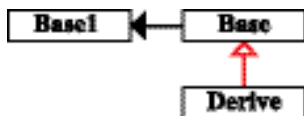
```
}
```

03 02 번을 변경해서, 객체를 힙 메모리 위에 만드세요.

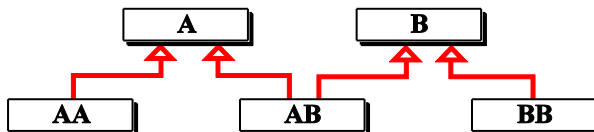
```
int main ( )  
{  
    // Instantiation of pointer  
    First* ptr;  
    // Creation of First object in the heap  
    ptr = new First (5);
```

```
    ptr -> print ();  
    // Creation of Second object in the heap  
    ptr = new Second (5, 7);  
    ptr -> print();  
    return 0;  
}
```

04 상속 관계를 갖는 Base 클래스와 Derived 클래스라는 클래스가 있고, Base 클래스가 Base1 클래스라는 다른 클래스를 포함(composition)하고 있다고 할 때, 세 클래스의 UML 다이어그램을 그리세요.



05 다음과 같은 클래스가 있을 때, 가상 베이스 클래스 또는 믹스인 클래스가 필요한지 생각해보고, 왜 그런지 설명하세요.



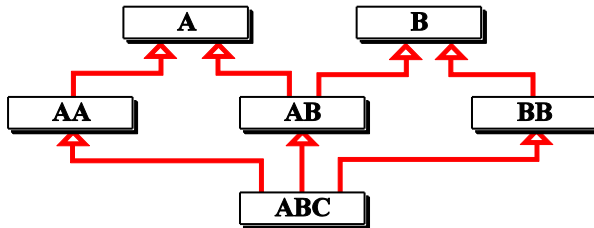
No. None of the derived classes inherits more than once from any of the base classes. Class AA inherits only from class A. Class B inherits only from class B. Class AB inherits once from class A and once from class B.

06 다음과 같은 코드가 있을 때, 클래스 A와 B가 다형성을 활용하게 해서 적절한 print 함수를 호출하게 하려면, 무엇을 추가해야 할지 설명하세요.

```
class A {...};  
class B: public A {...};  
  
int main()  
{  
    A* ptr  
    ptr = new A ();  
    ptr -> print ();  
    ptr = new B ();  
    ptr -> print ();  
    return 0;  
}
```

Out of three conditions for polymorphism (pointer or reference, exchangeable of objects, and virtual functions) we have the first two. We need to make the print function virtual to achieve polymorphism.

07 다음과 같은 클래스가 있을 때, 다중 상속 문제가 어떤 클래스에서 발생하는지 찾고, 왜 발생하는지 설명하세요.



The answer is yes. The class ABC creates multiple inheritance. Class ABC inherits class A once through AA and once through AB. Class ABC also inherits class B once through AB and once through BB.

Solutions

Problems

01 Fun 이라는 클래스 내부에서 다음과 같은 단항 연산자 오버로드 프로토타입을 만들었을 때 발생할 수 있는 오류를 찾으세요.

```
Fun operator+ ( );
```

Two errors. The host object must be constant and the returned type must also be constant. The correct declaration is shown below:

```
const Fun operator+( ) const;
```

02 Fun 이라는 클래스 내부에서 다음과 같은 단항 연산자 오버로드 프로토타입을 만들었을 때 발생할 수 있는 오류를 찾으세요.

```
Fun operator? (int x, int y, int z)
```

The symbol ? cannot be used as an operator.

03 Fun 이라는 클래스 내부에서 다음과 같은 이항 연산자 오버로드 프로토타입을 만들었을 때 발생할 수 있는 오류를 찾으세요.

```
void operator+ (const Fun& fun, const Fun& fun);
```

There are two errors. First, the function needs to return a constant Fun object. Second, the function should be defined as a friend function. The following is the correct prototype:

```
friend const Fun operator+ (const Fun& fun, const Fun& fun);
```

04 Fun 이라는 클래스 내부에서 다음과 같은 연산자 오버로드 프로토타입을 만들었을 때 발생할 수 있는 오류를 찾으세요.

```
double operator [ ] (int x, int y);
```

Three errors: The return type is missing (cannot be void); assuming accessor function (const), the key word **const** should be at the end; too many parameters (the array is implied by the function name).

05 Fun 이라는 클래스 내부에서 다음과 같은 연산자 오버로드 프로토타입을 만들었을 때 발생할 수 있는 오류를 찾으세요.

```
void operator( ) (int x, int y);
```

There are no errors.

06 Fun 이라는 클래스가 있을 때, 다음 두 연산자 오버로드 프로토타입의 차이를 설명하세요. 또한 Fun 클래스 내부에 두 프로토타입을 모두 만들 수 있을까요?

```
Fun operator += (const Fun& fun);  
Fun operator += (int x);
```

The first adds an instance of the class to the host. The second adds an integer value. Since the signatures are different, both can be present.

07 fun1 과 fun2 가 Fun 클래스의 인스턴스라고 할 때, 다음과 같은 연산이 가능하게 하려면, 어떤 연산자 오버로드 프로토타입이 필요한지 설명하세요.

```
fun1 += fun2  
fun2 = fun1  
fun1++  
fun1(a)
```

- a. `const Fun operator+= (Fun fun2);`
- b. `Fun& operator= (const Fun& fun1);`
- c. `const Fun operator++ (Fun fun1);`
- d. `Fun operator() (int a);`

08 다음과 같은 클래스에 덧셈 복합 할당 연산자(+=)를 오버로드하세요.

```
// Declaration  
int& Overload2 :: operator+=(const int& plusNum);  
// Definition  
int& Overload2 :: operator+=(const int& plusNum)  
{  
    value += plusNum  
    return value;  
}
```

Solutions

Problems

01 다음 프로그램의 실행 결과를 예측하세요.

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

int main ()
{
    int value = 30;
    try
    {
        if (value > 30) throw value;
    }
    cout << value;
    return 0;
}
```

The compilation fails because there is no *catch* clause after the *try* clause. We need to have a complete *try-catch* block made of two clauses.

02 다음 프로그램의 실행 결과를 예측하세요.

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

int main ()
{
    int value = 30;
    try
    {
        if (value < 20) throw value;
    }
    catch (int value)
```

```

    {
        cout << "In the catch clause." << endl;
    }
    cout << value << endl;
    return 0;
}

```

No exception is thrown. The catch block is ignored and the value 30 is printed.

03 다음 프로그램의 실행 결과를 예측하세요.

```

#include <iostream>
using namespace std;

void fun (int x )
{
    if (x < 10) throw 10.0;
}

int main ()
{
    try
    {
        fun (5);
    }
    catch (int value)
    {
        cout << value << endl;
    }
    return 0;
}

```

The function *fun* is called with argument of 5. It throws an exception of type *double*, but there is no *catch* block that can catch a value of type *double*. The program terminates ungracefully.

04 다음 프로그램에서 잘못된 부분을 찾으세요.

```

void fun (int x) throw ()
{
    if (x < 10) throw 10.0;
}

```

```
}
```

This is an example of bad coding. The function defines that the function is not supposed to throw any exception, but the function throws an exception of type *double*. If the user and the designer of the function are two different entities, this is a case of confusion.

05 다음 프로그램에서 잘못된 부분을 찾으세요.

```
void fun (double x) throw (double)
{
    if (x < 10.0) throw 10.0;
}
```

There is nothing wrong with this function definition. It defines that the function may throw a *double* value and it does so.

06 다음 프로그램의 실행 결과를 예측하세요.

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

void fun (int x ) throw (int)
{
    if (x > 1000) throw 10000;
}

int main ()
{
    try
    {
        fun (1002 );
    }
    catch (int value)
    {
        cout << value << endl;
    }
    return 0;
}
```

The value 10000 is printed and the program terminates. An exception (10000) is thrown, caught, and handled.

07 다음 프로그램의 실행 결과를 예측하세요.

```
#include <iostream>
```



```
using namespace std;

int main ()
{
    int value = 30;
    if (value > 20) throw value;
    cout << value;
    return 0;
}
```

The program terminates with an uncaught exception of type `int`.

Solutions

Problems

01 다음과 같은 코드가 있다고 합시다. 컴파일 시점에 `fun` 함수가 내부적으로 몇 가지 형태로 만들어지는 지 예측하세요.

```
template <typename T>
void fun (T x)
{
    ...
}
int main ()
{
    fun (7);
    fun (12.5);
    fun ("Hello");
    return 0;
}
```

Three. One for *int*, one for *double*, and one for *string*.

02 다음과 같은 코드가 있다고 합시다. 컴파일 시점에 `fun` 함수가 내부적으로 몇 가지 형태로 만들어지는 지 예측하세요.

```
template <typename T>
void fun (T x)
{
    ...
}
```

```
int main ()
{
    Sample (7);
    Sample (9) ;
    return 0;
}
```

Just one. An instance of the function *fun* that has an integer parameter.

03 두 숫자(int, long int, double, long double 자료형 등)의 평균을 구하는 템플릿 함수를 작성하 세요.

```
template <typename T>
T average (T first, T second)
{
    T avg = (first + second) / 2.0;
    return avg;
}
```

04 03 번의 함수를 오버로드해서, 세 숫자의 평균을 구하는 템플릿 함수를 작성하세요

```
template <typename T>
T average (T first, T second, T third)
{
    T avg = (first + second + third) / 3.0;
    return avg;
}
```

05 서로 다른 2 개의 자료형을 쌍으로 가질 수 있는 pair 라는 구조체가 있습니다. 이를 활용해서 int 와 double 자료형의 pair 구조체, double 과 char 자료형의 pair 구조체 등을 만드세요.

```
template <typename T1,  typename T2>
struct Pair
{
    T1 first;
    T2 second;
};
```

Solutions

Problems

01 파일 이름을 매개변수로 전달하고, 입력 모드로 파일을 여는 openInput 이라는 함수를 작성하세요.

```
void openInput (const char* fName)
{
    ifstream stream;
    stream.open (fName);
    if (stream.fail ())
    {
        cout << "File cannot be open. Program terminates.";
        assert (false);
    }
    stream.close ( );
}
```

02 파일 이름을 매개변수로 전달하고, 출력 모드로 파일을 여는 openOutput 이라는 함수를 작성하세요.

```
void openOutput (const char* fName)
{
    ofstream stream;
    stream.open (fName);
    if (stream.fail ())
    {
        cout << "File cannot be open. Program terminates.";
        assert (false);
    }
    stream.close ( );
}
```

03 파일 이름과 문자의 위치를 매개변수로 전달하고, 파일의 특정 위치에 있는 문자를 리턴하는 함수를 작성하세요.

```
char getRandom (const char* fileName, int location)
{
    char c;
    ifstream stream;
    stream.open (fileName);
    if (stream.fail ())
    {
        cout << "File cannot be open. Program terminates.";
        assert (false);
    }
    stream.seekg (location);
    stream >> c;
    stream.close ( );
    return c;
}
```

04 파일의 이름, 문자의 위치, 교체할 문자를 매개변수로 전달하고, 파일의 특정 위치에 있는 문자를 변경하는 함수를 작성하세요.

```
void setRandom (const char* fileName, int loc, char c)
{
    fstream stream;
    stream.open (fileName, ios :: in | ios :: out);
    if (stream.fail ())
    {
        cout << "File cannot be open. Program terminates.";
        assert (false);
    }
    stream.seekp (loc);
    stream.put (c);
    stream.close ( );
}
```

05 어떤 파일의 내용을 다른 파일에 복사하는 함수를 작성하세요.

```

void copy (const char* fileName1, const char* fileName2)
{
    ifstream stream1;
    ofstream stream2;
    stream1.open (fileName1, ios :: in);
    stream2.open (fileName2, ios :: out);
    if (stream1.fail () || stream2.fail ())
    {
        cout << "File (s) cannot be open. Program terminates.";
        assert (false);
    }
    char c;
    while (!stream1.eof())
    {
        c = stream1.get();
        stream2.put (c);
    }
    stream1.close();
    stream2.close();
}

```

06 두 파일의 내용을 비교해서 파일의 내용이 같을 경우 true, 아닐 경우 false 를 리턴하는 함수를 작성해하세요.

```

bool compare (const char* fileName1, const char* fileName2)
{
    ifstream stream1;
    ifstream stream2;
    stream1.open (fileName1, ios :: in);
    stream2.open (fileName2, ios :: in);
    if (stream1.fail () || stream2.fail ())
    {
        cout << "File (s) cannot be open. Program terminates.";
        assert (false);
    }
    char c1, c2;
    bool equal = true;

```

```

do
{
    c1 = stream1.get();
    c2 = stream2.get();
    if (c1 != c2)
    {
        equal = false;
    }
} while (!stream1.eof() || !stream2.eof());
stream1.close();
stream2.close();
return equal;
}

```

07 어떤 파일의 내용을 다른 파일 뒤에 추가하는 함수를 작성하세요.

```

void append (const char* fileName1, const char* fileName2)
{
    ifstream stream1;
    ofstream stream2;
    stream1.open (fileName1, ios :: in);
    stream2.open (fileName2, ios :: out | ios :: app);
    if (stream1.fail () || stream2.fail ())
    {
        cout << "File (s) cannot be open. Program terminates.";
        assert (false);
    }
    char c;
    while (!stream1.eof())
    {
        c = stream1.get();
        stream2.put(c);
    }
    stream1.close();
    stream2.close();
}

```


08 어떤 파일의 홀수 번째 문자를 다른 파일에, 짝수 번째 문자를 또 다른 파일에 복사하는 함수를 작성하세요.

```
void split (const char* file1, const char* file2, const char* file3)
{
    ifstream stream1;
    ofstream stream2;
    ofstream stream3;
    stream1.open (file1, ios :: in);
    stream2.open (file2, ios :: out);
    stream3.open (file3, ios :: out);
    if (stream1.fail () || stream2.fail () || stream3.fail())
    {
        cout << "File (s) cannot be open. Program terminates.";
        assert (false);
    }
    char c1 , c2;
    while (!stream1.eof())
    {
        c1 = stream1.get();
        c2 = stream1.get();
        stream2.put (c1);
        stream3.put (c2);
    }
    stream1.close();
    stream2.close();
    stream3.close();
}
```

09 어떤 파일을 읽어 들이고, 그 뒤에 글자 수를 세어서 글자 수를 출력하는 함수를 작성하세요. 글자 수 앞에는 띄어쓰기 하나를 넣으세요. 예를 들어 글자 수가 24 라면, '24'를 넣습니다.

```
void addCount (const char* file)
{
    ifstream stream1;
    stream1.open (file, ios :: in);
    if (stream1.fail ())
    {
```

```

        cout << "File cannot be open. Function terminates.";
        assert (false);
    }
    int count = 0;
    while (!stream1.eof())
    {
        stream1.get();
        count++;
    }
    stream1.close();
    ofstream stream2;
    stream2.open (file, ios :: out | ios :: app);
    if (stream2.fail ())
    {
        cout << "File cannot be open. Function terminates.";
        assert (false);
    }
    stream2.put ( ' ');
    stream2 << count;
    stream2.close ();
}

```

10 문자 배열의 내용을 파일에 출력하는 함수를 작성하세요.

```

void arrayToFile (const char* file, char arr [ ], int size)
{
    ofstream stream;
    stream.open (file, ios :: out);
    if (stream.fail ())
    {
        cout << "File (s) cannot be open. Program terminates.";
        assert (false);
    }
    for (int i = 0; i < size; i++)
    {
        stream.put (arr [i]);
    }
}

```

```
stream.close();  
}
```

11 id(int 자료형), name(string 자료형), gpa(double 자료형)이라는 학생 정보로 구성된 학생 데이터 레코드를 한 줄 읽어 들이고, 이를 변수에 저장하는 코드를 작성하세요. 사용자로부터 '1 윤인성 4.2' 등의 문자열 한 줄을 입력받고, 이를 id, name, gpa 변수에 저장하는 형태로 구현하는 방식입니다.

```
int id;  
string name;  
double gpa;  
string record ;  
cout << "Enter a complete record of a student: ";  
getline (cin, record);  
istringstream stream (record);  
stream >> id;  
stream >> name;  
stream >> gpa;
```

Solutions

Problems

01 모든 형태의 벡터를 매개변수로 받아서, 벡터의 요소를 출력하는 템플릿 함수를 작성하세요.

```
template <typename T>
void print (const vector <T>& vec)
{
    for (int i = 0; i < vec.size (); i++)
    {
        cout << vec.at(i) << endl;
    }
}
```

02 모든 형태의 리스트를 매개변수로 받아서, 리스트의 요소를 출력하는 템플릿 함수를 작성하세요.

```
template <typename T>
void print (list<T>& lst)
{
    typename list <T> :: const_iterator iter;
    for (iter = lst.begin (); iter != lst.end(); iter++)
    {
        cout << *iter << endl;
    }
}
```

03 10 개의 요소를 갖는 벡터를 만들고, at 함수를 활용해 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 로 요소를 초기화하는 코드를 작성하세요.

04 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 를 요소로 갖는 리스트를 작성하는 코드를 작성하세요.

```
vector<int> vec (10);  
for (int i = 0; i < 10; i++)  
{  
    vec.at (i) = i;  
}
```

05 int 자료형을 요소로 갖는 벡터의 2 번째 인덱스와 3 번째 인덱스의 값을 스왑하는 코드를 작성하세요.

```
int temp = vec[2];  
vec[2] = vec[3];  
vec[3] = temp;
```

06 int 자료형을 요소로 갖는 벡터의 2 번째 값과 3 번째 값을 스왑하는 코드를 작성하세요. 임의 접근 반복자가 아닌, 일반 반복자와 ++ 연산자를 사용해서 스왑하는 코드를 작성하세요.

```
list<int> :: iterator iter1, iter2;  
iter1 = iter2 = lst.begin();  
// Move iter1 to position 2  
++iter1;  
++iter1;  
// Move iter2 to position 3
```

```

++iter2;
++iter2;
++iter2;
// Swap the values of two positions
int temp = *iter1;
*iter1 = *iter2;
*iter2 = temp;

```

07 벡터에는 `push_front` 연산이 정의되어 있지 않습니다. `int` 자료형을 요소로 갖는 벡터의 가장 앞에 요소를 추가하는 코드를 직접 작성하세요.

The following shows the code. Note that we need to use the `insert` member function and an iterator that points to the beginning of the vector. The rest of the items are pushed one element toward the end.

```

vec.insert (vec.begin(), element);

```

08 벡터에는 `pop_front` 연산이 정의되어 있지 않습니다. `int` 자료형을 요소로 갖는 벡터의 첫 번째 요소를 제거하는 코드를 직접 작성하세요.

The following shows the code. Note that we need an iterator to point to the first element. The rest of the items are pushed one element toward the front.

```

vec.erase (vec.begin());

```

09 벡터에는 요소를 제거하는 함수가 정의되어 있지 않습니다. 벡터의 요소를 제거하는 함수를 직접 만드세요.

```

vector<int> :: iterator iter;
iter = find (vec.begin(), vec.end(), value);
vec.erase (iter);

```

10 우선 순위 큐에 14, 24, 76, 18, 20, 54 를 삽입한 후, 2 개의 정수를 팝(pop)했다고 합시다. 이어서

정수를 팝하면, 어떤 정수가 팝 될지 예측하세요.

The next integer to be popped is 24.

11 세트에 20, 17, 20, 14, 15, 19, 17, 10 을 삽입한 후에 size 멤버 함수를 출력하면, 어떤 값이 출력

될지 예측하세요.

After the insertion, the set contains: 10 14 15 17 19 20. The size is 6.

12 맵에 (20, 10), (14, 40), (17, 3)이라는 페어가 들어있다고 합시다. 맵 내부에서 이러한 페어가 어떤 순서로 저장될지 예측하세요.

The order is: (14, 40) (17, 3) (20, 10).

13 정수 20, 14, 18, 22, 76 이 저장된 st 라는 이름의 세트가 있다고 합시다. 이때 18 과 22 를 출력할 수 있는 코드를 작성하세요.

```
set <int> :: iterator iter1 = st.begin();  
set <int> :: iterator iter2 = st.end();  
cout << *(++iter1) << endl;  
cout << *(--iter2) << endl;
```

14 (15, 10), (16, 10), (6, 16), (12, 8)이라는 페어가 저장된 mp 라는 이름의 맵이 있다고 합시다. 이때 다음 코드는 무엇을 출력할지 예측하세요.

```
cout << mp [6]  << endl;  
cout << mp [16] << endl;  
cout << mp [10] << endl;
```

The following integers are printed. Note that the last statement prints 0 because the pair with key = 10 is not in the map.

```
16  
10  
0
```

15 맵을 만들고 (3, 10), (5, 12), (7, 8)이라는 페어를 insert 멤버 함수를 활용해서 삽입하는 코드를 작성하세요.

```
map <int, int> mp;  
mp.insert (pair <int, int> (3, 10));  
mp.insert (pair <int, int> (5, 12));  
mp.insert (pair <int, int> (7, 18));
```