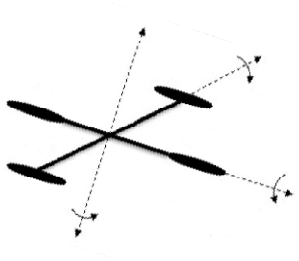
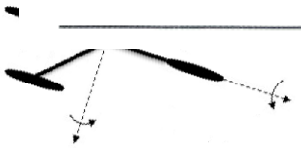
원시 데이터 유형과 연산



- 숫자 데이터 유형

이름	같은 표현	범위	저장 공간 크기
short	short int	$-2^{15}(-32,768)$~$2^{15}-1(32,767)$	16-bit signed
unsigned short	unsigned short int	0~$2^{16}-1(65535)$	16-bit unsigned
int		$-2^{31}(-2147483648)$~$2^{31}-1(2147483647)$	32-bit signed
unsigned int	unsigned	0~$2^{32}-1(4294967295)$	32-bit unsigned
long	long int	$-2^{31}(-2147483648)$~$2^{31}-1(2147483647)$	32-bit signed
unsigned long	unsigned long int	0~$2^{32}-1(4294967295)$	32-bit unsigned
float		음수범위: −3.4028235E+38~−1.4E−45	32-bit IEEE 754
		양수범위: 1.4E−45~3.4028235E+38	
double		음수범위: −1.7976931348623157E+308~	64-bit IEEE 754
		−4.9E−324	
		양수범위:	
		4.9E−324~1.7976931348623157E+308	
long double		음수범위: −1.18E+4932~3.37E−4932	80-bit
		양수범위: 3.37E−4932~1.18E+4932	
		유효숫자 길이: 19	



- printf 문

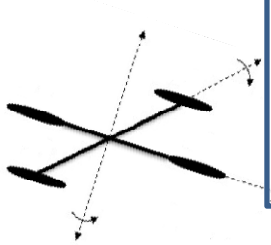
가장 기본적인 출력 함수. (stdio.h)

문법)

```
printf("Test printf. a = %d  ₩n", a);
printf("%d, %f, %c ₩n", a, b, c);
```

```
%d : 정수형
%ld : long int 로 선언된것
%f : float 나 double 같은 실수형
%c : char 형 문자열이 아닌 한 '문자'
%s : char 형 문자열(문자배열 이라고도부른다.)
```

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int a;
    float b;
    char c;

    a = 40;
    b = 40;
    c = 40;

    printf("Test printf. a = %d  ₩n", a);
    printf("%d, %f, %c ₩n", a, b, c);
    system("PAUSE");
    return EXIT_SUCCESS;
```



```
C:₩Documents and Settings₩Administrator₩바탕 화면
Test printf. a = 40
40, 40.000000, (
계속하려면 아무 키나 누르십시오 . . .
```

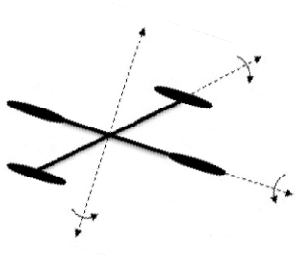
- scanf 문

가장 기본적인 입력 함수. (stdio.h)

문법)

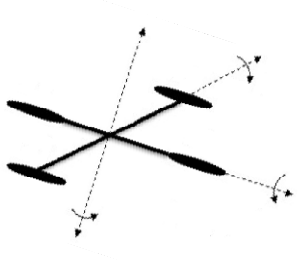
```
scanf("%d", &a);
```

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int a, b, c;

    printf ("정수 3개를 입력하시오.₩n");
    scanf ("%d%d%d", &a, &b, &c);
    printf ("당신이 입력한 정수는 %d, %d, %d입니다.₩n", a, b, c);

    system("PAUSE");
    return EXIT_SUCCESS;
}
}
```



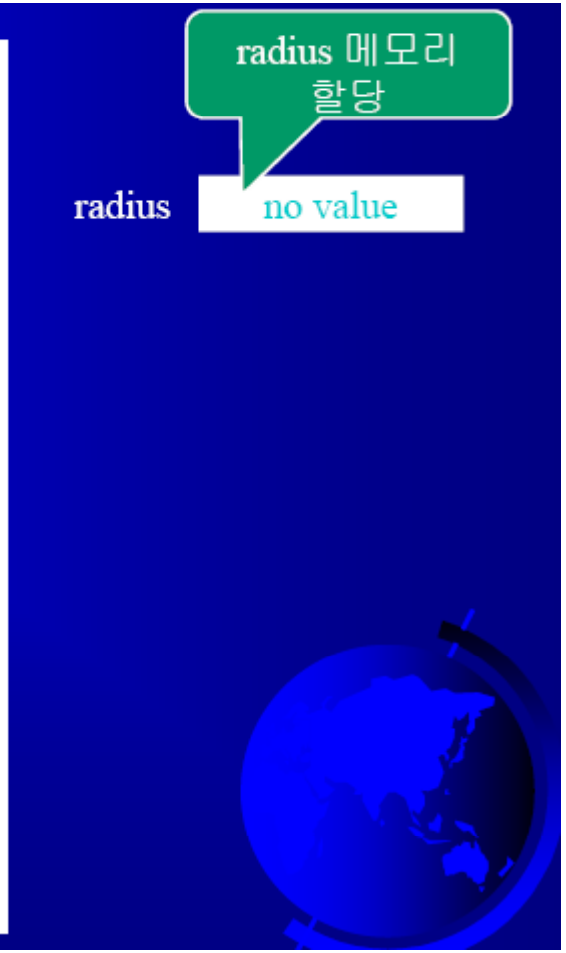
- C++ 표준 입출력 함수 cout, cin

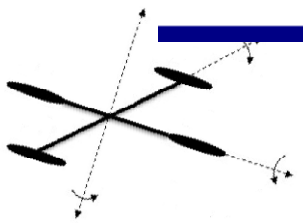
```
cout<<" 내용 ";
cout<<변수;
cout<<" 내용 "<<변수;
cout<<" 내용 "<<endl;
cout<<" 내용 /n";

cin>>변수;
```



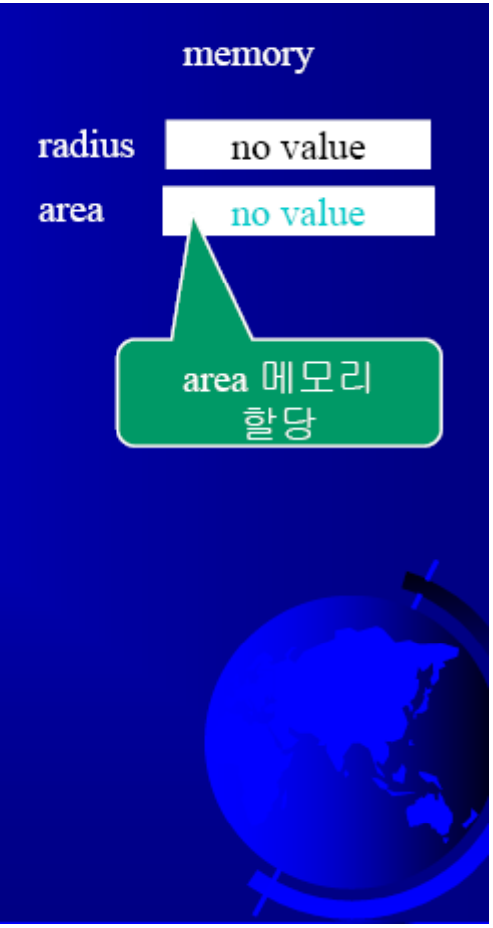
- 간단한 프로그램 작성 : 원의 면적 구하기

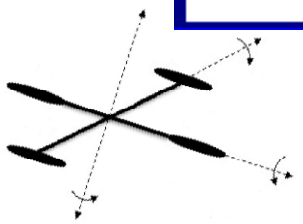
```
#include <iostream>

int main() {
  double radius;
  double area;

  // Step 1: Read in radius
  radius = 20;

  // Step 2: Compute area
  area = radius * radius * 3.14159;

  // Step 3: Display the area
  std::cout << "The area is ";
  std::cout << area << std::endl;
}
```



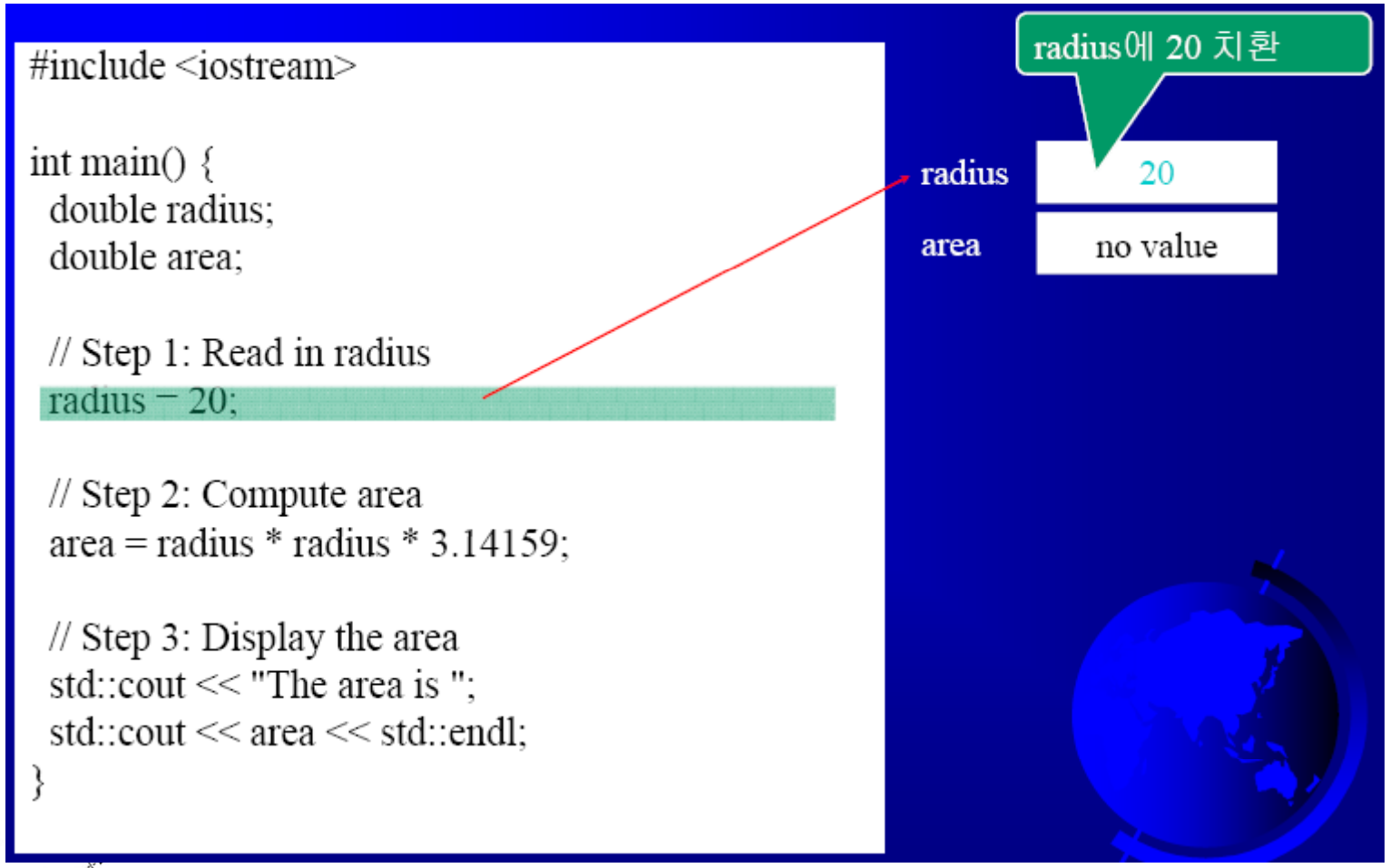


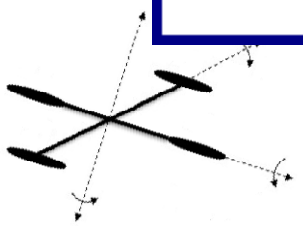
```
#include <iostream>

int main() {
  double radius;
  double area;

  // Step 1: Read in radius
  radius = 20;

  // Step 2: Compute area
  area = radius * radius * 3.14159;

  // Step 3: Display the area
  std::cout << "The area is ";
  std::cout << area << std::endl;
}
```



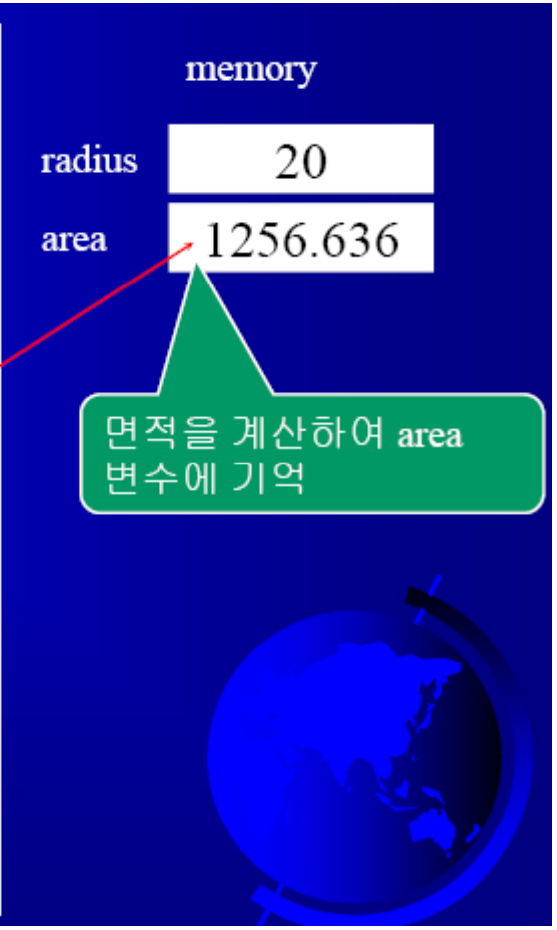


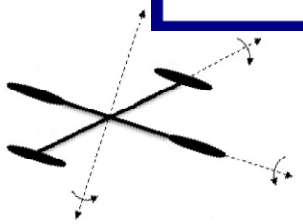
```
#include <iostream>

int main() {
  double radius;
  double area;

  // Step 1: Read in radius
  radius = 20;

  // Step 2: Compute area
  area = radius * radius * 3.14159;

  // Step 3: Display the area
  std::cout << "The area is ";
  std::cout << area << std::endl;
}
```



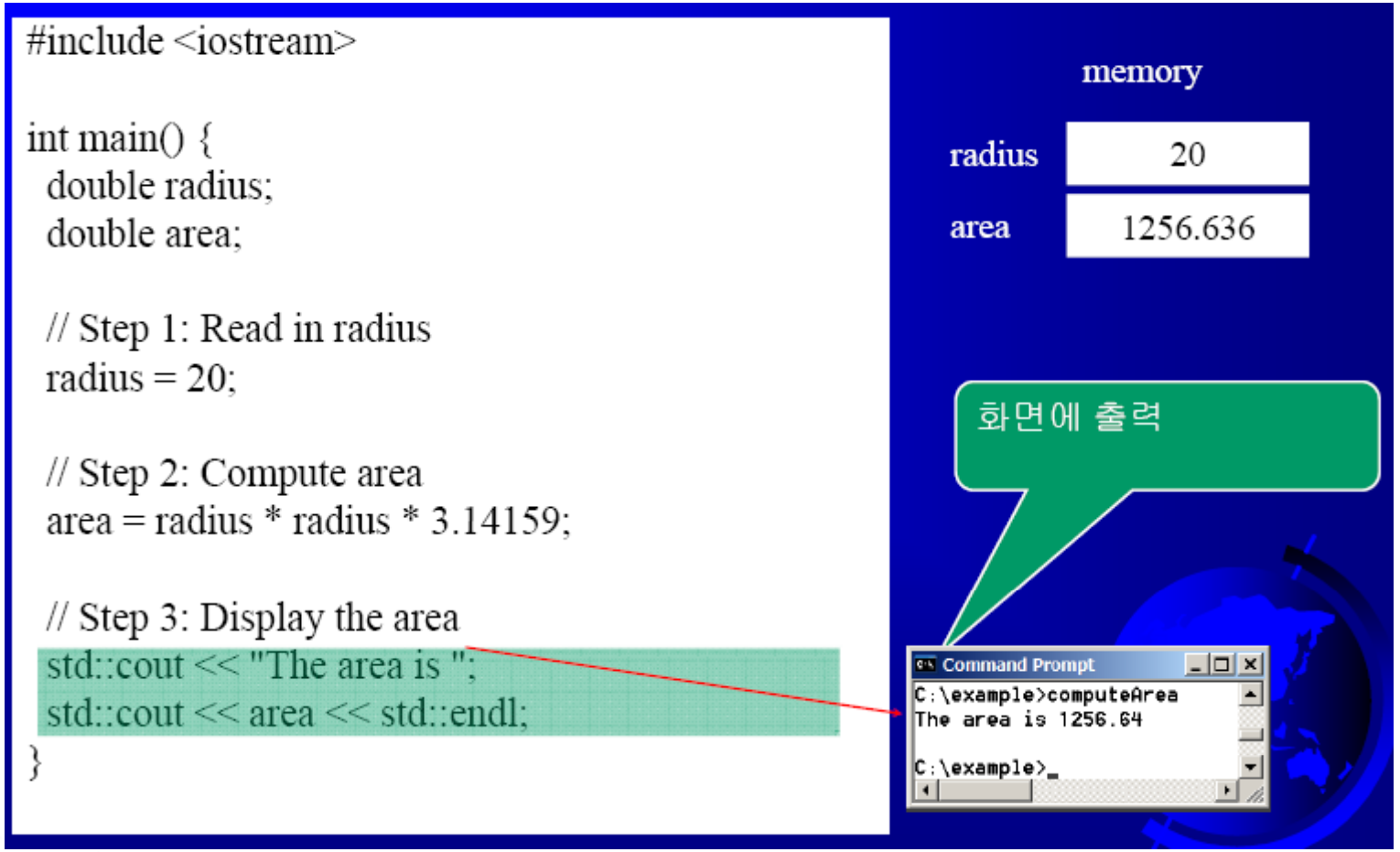


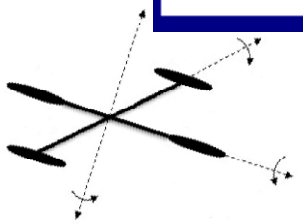
```
#include <iostream>

int main() {
  double radius;
  double area;

  // Step 1: Read in radius
  radius = 20;

  // Step 2: Compute area
  area = radius * radius * 3.14159;

  // Step 3: Display the area
  std::cout << "The area is ";
  std::cout << area << std::endl;
}
```





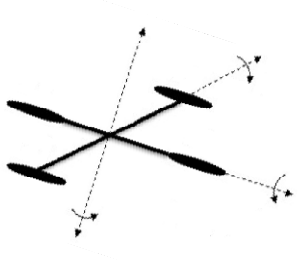
```
#include <iostream>

int main() {
  double radius;
  double area;

  // Step 1: Read in radius
  radius = 20;

  // Step 2: Compute area
  area = radius * radius * 3.14159;

  // Step 3: Display the area
  std::cout << "The area is ";
  std::cout << area << std::endl;
}
```





- 간단한 프로그램 작성 : 원의 면적 구하기 (키보드로 입력받기)

```
#include <iostream>

int main() {
  // Step 1: Read in radius
  double radius;
  std::cout << "Enter a radius: ";
  std::cin >> radius;

  // Step 2: Compute area
  double area = radius * radius * 3.14159;

  // Step 3: Display the area
  std::cout << "The area is " << area << std::endl;
}
```

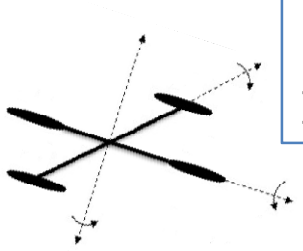


- std:: 접두어(prefix) 생략

std:: => standard name space. 잠재적인 명명(naming)문제의 방어차원에서 모든 이름이 namespace를 가짐.

using namespace std;

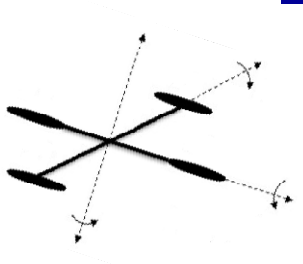
```
#include <iostream>
using namespace std;

int main() {
  // Step 1: Read in radius
  double radius;
  cout << "Enter a radius: ";
  cin >> radius;

  // Step 2: Compute area
  double area = radius * radius * 3.14159;

  // Step 3: Display the area
  cout << "The area is " << area << endl;
}
```



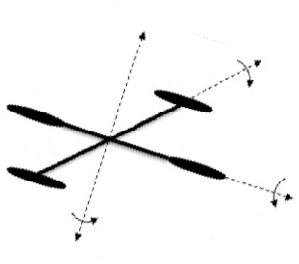
-변수 : 특정 유형의 데이터를 저장

-변수의 선언 : **datatype variableName;**

```
int x;                    // 정수 변수 x를 선언

double radius;            // double(배정도실수) 변수
                          // radius를 선언

double interestRate;  // double 변수 interestRate를
                          // 선언

char a;                   // 문자 변수 a를 선언
```



```
int x = 1;                  // 변수 x에 1을 할당

double radius = 1.0;  // 변수 radius에 1.0을 할당

x = 5 * (3 / 2) + 3 * 2;  // 수식의 값의 변수 x에 할당

x = y + 1;              // 변수 y와 1을 더해서 변수 x에 할당

area = radius * radius * 3.14159; // 면적(area) 계산
```



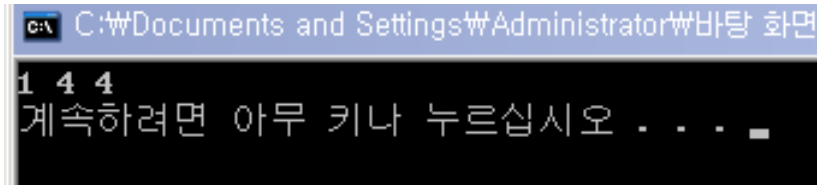
-상수(constant) :

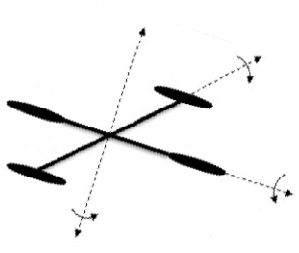
변수는 저장값을 변경할 수 있지만,
상수는 선언된 이후 변경될 수 없다.

```
const datatype CONSTANTNAME = VALUE;

const double PI = 3.14159;
const int SIZE = 3;
```

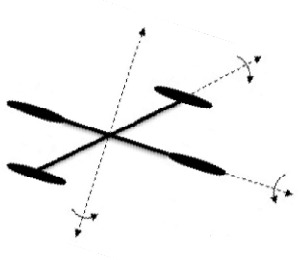
-sizeof 함수 : 데이터 유형의 크기를 구함.

```
cout<<sizeof(char)<<" "<<sizeof(int)<<" "<<sizeof(float)<<endl;
```





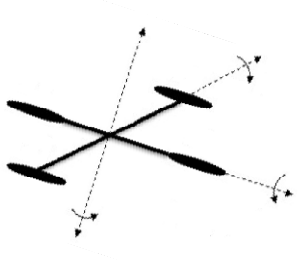
```
#include <iostream>
using namespace std;

int main() {
  const double PI = 3.14159;

  // Step 1: Read in radius
  double radius = 20;

  // Step 2: Compute area
  double area = radius * radius * PI;

  // Step 3: Display the area
  cout << "The area is ";
  cout << area << std::endl;
}
```



- 리터럴(literal) : 프로그램에서 직접 사용되는 상수 값.

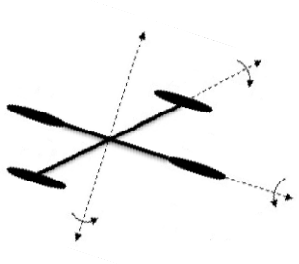
```
int ii = 34;
long k = 1000000;
double d = 5.0;
```

8진수	->	0으로 시작	->010 (10진수 8)
16진수	->	0x로 시작	->0x000F (10진수 15)



- 수 관련 연산자

Name	Meaning	Example	Result
+	Addition	34 + 1	35
-	Subtraction	34.0 - 0.1	33.9
*	Multiplication	300 * 30	9000
/	Division	1.0 / 2.0	0.5
%	Remainder	20 % 3	2



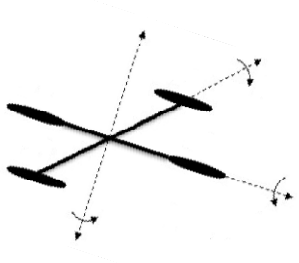
- 정수 나눗셈

+, -, *, /, %

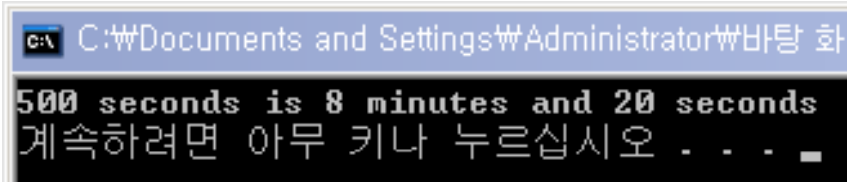
5 / 2는 정수 2가 된다.

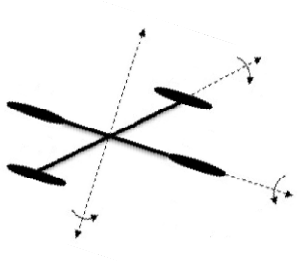
5.0 / 2는 double 값 2.5가 된다.

5 % 2는 1이 된다. (나머지 값)



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

int main()
{
  int seconds = 500;
  int minutes = seconds / 60;
  int remainingSeconds = seconds % 60;
  cout << seconds << " seconds is " << minutes <<
    " minutes and " << remainingSeconds << " seconds " << endl;
  return 0;
}
```

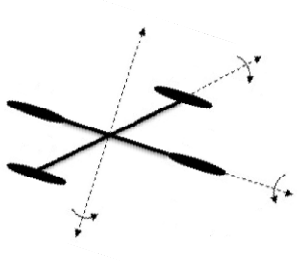




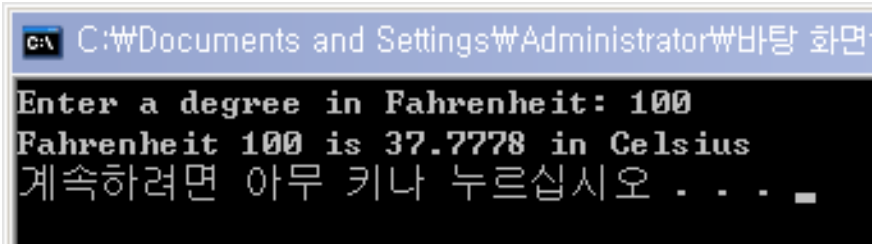
- 산술식

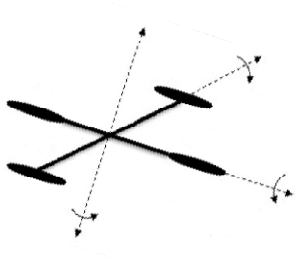
$$\frac{3+4x}{5} - \frac{10(y-5)(a+b+c)}{x} + 9\left(\frac{4}{x} + \frac{9+x}{y}\right)$$


```
(3+4*x)/5 - 10*(y-5)*(a+b+c)/x + 9*(4/x + (9+x)/y)
```



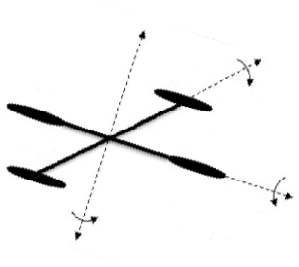
```
#include <iostream>
using namespace std;

int main()
{
  // Enter a degree in Fahrenheit
  double fahrenheit;
  cout << "Enter a degree in Fahrenheit: ";
  cin >> fahrenheit;

  // Obtain a celsius degree
  double celsius = (5.0 / 9) * (fahrenheit - 32);

  // Display result
  cout << "Fahrenheit " << fahrenheit << " is " <<
      celsius << " in Celsius" << endl;
  return 0;
}
```





- 단축연산자

연산자	사용예	같은표현
+=	i += 8	i = i + 8
-=	f -= 8.0	f = f - 8.0
*=	i *= 8	i = i * 8
/=	i /= 8	i = i / 8
%=	i %= 8	i = i % 8



- 증가/감소 연산자

Operator	Name	Description
++var	전치증가	변수의 값을 우선 1만큼 증가 시킨 후 사용된다.
var++	후치증가	변수의 값을 사용 후 1만큼 증가 시킨다.
--var	전치감소	변수의 값을 우선 1만큼 감소 시킨 후 사용된다.
var--	후치감소	변수의 값을 사용 후 1만큼 감소 시킨다.

```
int i = 10;
int newNum = 10 * i++;
```

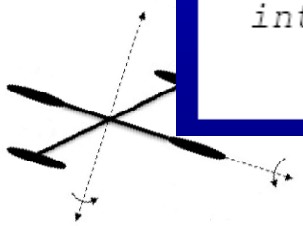
Same effect as

```
int newNum = 10 * i
i = i + 1;
```

```
int i = 10;
int newNum = 10 * (++i);
```

Same effect as

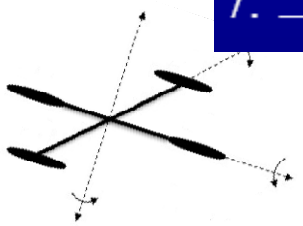
```
i = i + 1;
int newNum = 10 * i;
```



Conversion Rules

서로 다른 유형 값에 대해서 이항 연산을 할 때 C++는 자동으로 피연산자를 다음의 규칙에 따라 변환한다.

1. 피연산자에 하나의 long double이 있으면 나머지도 long double로 변환된다.
2. 그렇지 않고, double이 있으면 나머지도 double로 변환된다.
3. 그렇지 않고, float가 있으면 나머지도 float로 변환된다.
4. 그렇지 않고, unsigned long이 있으면 나머지도 unsigned long으로 변환된다.
5. 그렇지 않고, long이 있으면 나머지도 long으로 변환된다.
6. 그렇지 않고, unsigned int가 있으면 나머지도 unsigned int로 변환된다.
7. 그렇지 않으면 모든 피연산자가 int로 변환된다.



암시적인 캐스팅

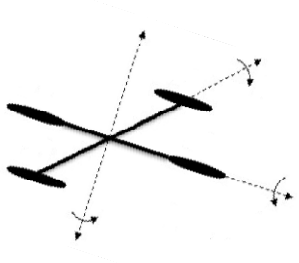
double d = 3; (type widening)

명시적인 캐스팅

int i = static_cast<int>(3.0); (타입 축소)

int i = (int)3.9; (소수점 이하는 버림)

```
double d = 4.5;

int i = static_cast<int>(d); // d의 유형은 변경되지 않음
```

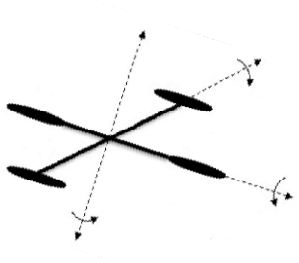


- char 유형

```
char letter = 'A'; (ASCII)
char numChar = '4'; (ASCII)
```

```
char ch = 'a';
cout << ++ch;
```

```
cout << "Enter a character: ";
char ch;
cin >> ch;
```



- 이스케이프 문자

Character Escape Sequence	Name	ASCII Code
\b	Backspace	8
\t	Tab	9
\n	Linefeed	10
\f	Formfeed	12
\r	Carriage Return	13
\\	Backslash	92
\'	Single Quote	39
\"	Double Quote	34

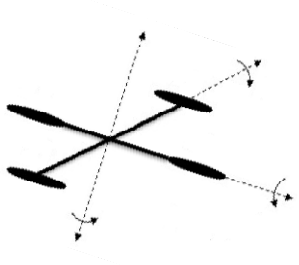
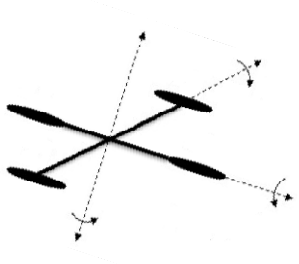


TABLE B.1 ASCII Character Set in the Decimal Index

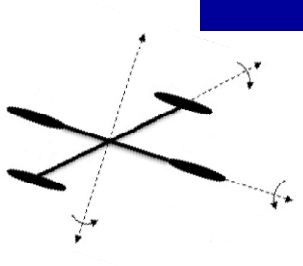
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht
1	nl	vt	ff	cr	so	si	dle	dc1	dc2	dc3
2	dc4	nak	syn	etb	can	em	sub	esc	fs	gs
3	rs	us	sp	!	"	#	$	%	&	'
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	`	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	del		



예제: 돈을 소단위 화폐로 변환하기

일정한 금액의 돈을 소단위 화폐로 변환하는 프로그램을 살펴본다. 사용자가 일정 금액에 대해 달러와 센트 값을 double 형태로 입력하면 금액을 작은 화폐 단위(달러(dollar), 쿼터(quarter), 다임(dime), 니켈(nickel), 페니(penny))로 변환하여 출력한다.

출력 조건은 달러를 시작으로 큰 단위의 화폐를 최대로 하여 출력하는 것이다.

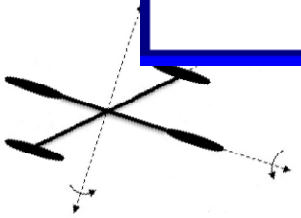


amount는 11.56

```
int remainingAmount = (int)(amount * 100);

// Find the number of one dollars
int numberOfOneDollars = remainingAmount / 100;
remainingAmount = remainingAmount % 100;

// Find the number of quarters in the remaining amount
int numberOfQuarters = remainingAmount / 25;
remainingAmount = remainingAmount % 25;

// Find the number of dimes in the remaining amount
int numberOfDimes = remainingAmount / 10;
remainingAmount = remainingAmount % 10;

// Find the number of nickels in the remaining amount
int numberOfNickels = remainingAmount / 5;
remainingAmount = remainingAmount % 5;

// Find the number of pennies in the remaining amount
int numberOfPennies = remainingAmount;
```

remainingAmount 1156

remainingAmount 초기화

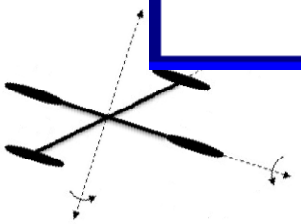


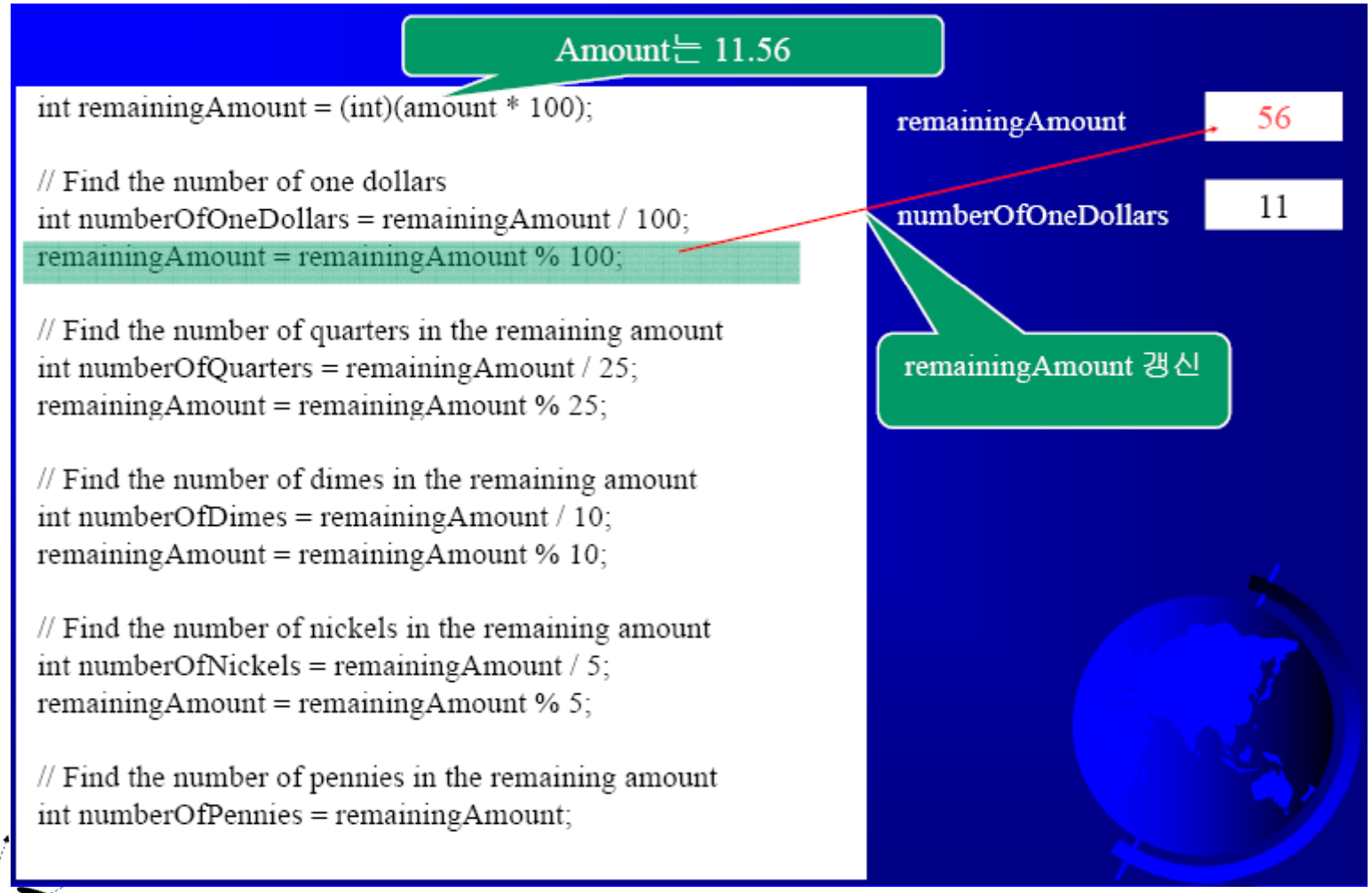
Amount는 11.56

```
int remainingAmount = (int)(amount * 100);

// Find the number of one dollars
int numberOfOneDollars = remainingAmount / 100;
remainingAmount = remainingAmount % 100;

// Find the number of quarters in the remaining amount
int numberOfQuarters = remainingAmount / 25;
remainingAmount = remainingAmount % 25;

// Find the number of dimes in the remaining amount
int numberOfDimes = remainingAmount / 10;
remainingAmount = remainingAmount % 10;

// Find the number of nickels in the remaining amount
int numberOfNickels = remainingAmount / 5;
remainingAmount = remainingAmount % 5;

// Find the number of pennies in the remaining amount
int numberOfPennies = remainingAmount;
```

remainingAmount	1156
numberOfOneDollars	11

numberOfOneDollars 저장



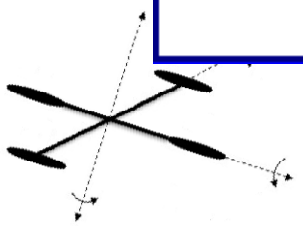

Amount는 11.56
int remainingAmount = (int)(amount * 100);
// Find the number of one dollars
int numberOfOneDollars = remainingAmount / 100;
remainingAmount = remainingAmount % 100;
// Find the number of quarters in the remaining amount
int numberOfQuarters = remainingAmount / 25;
remainingAmount = remainingAmount % 25;
// Find the number of dimes in the remaining amount
int numberOfDimes = remainingAmount / 10;
remainingAmount = remainingAmount % 10;
// Find the number of nickels in the remaining amount
int numberOfNickels = remainingAmount / 5;
remainingAmount = remainingAmount % 5;
// Find the number of pennies in the remaining amount
int numberOfPennies = remainingAmount;
remainingAmount
56
numberOfOneDollars
11
remainingAmount 갱신

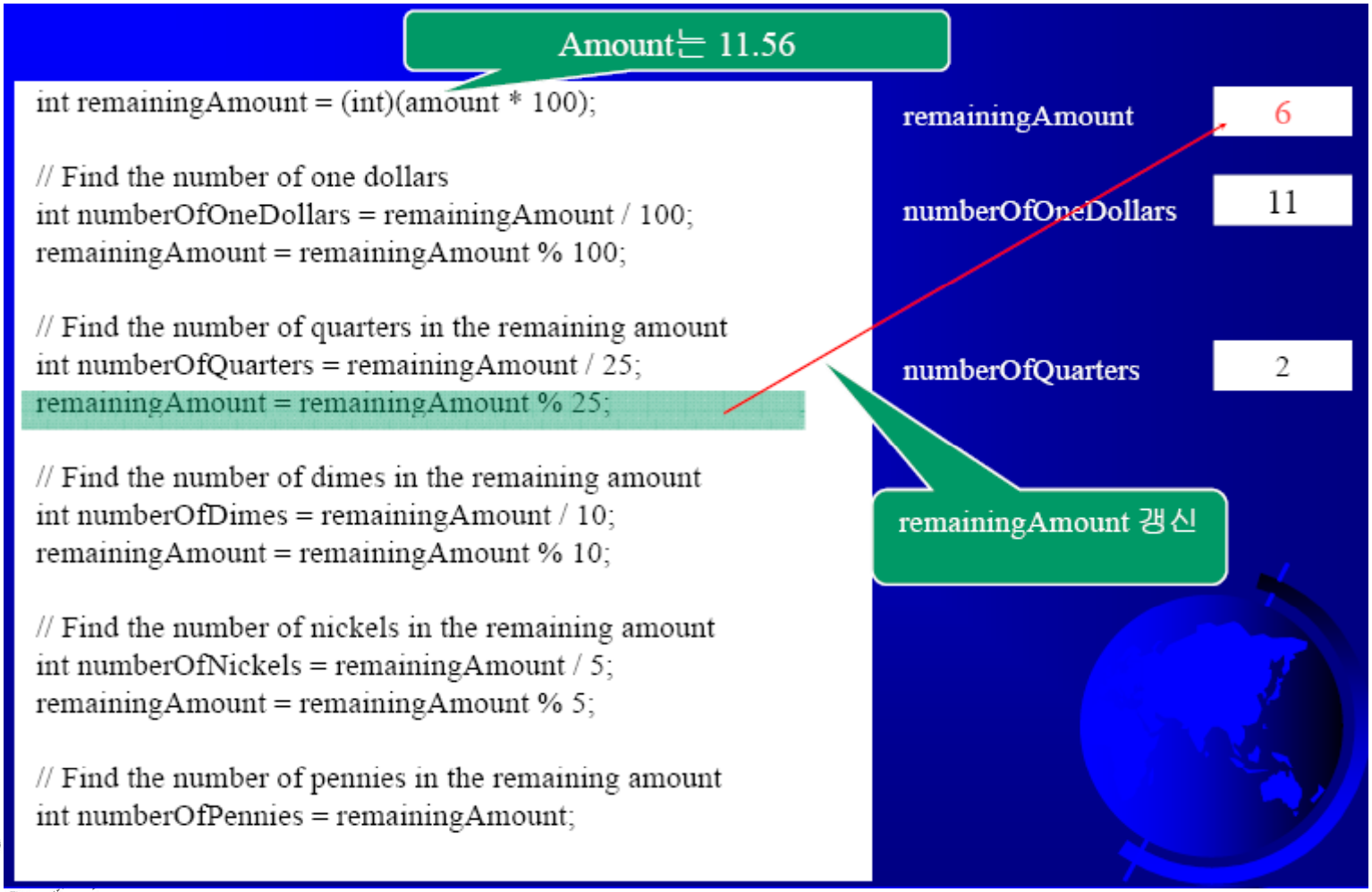
Amount는 11.56

```
int remainingAmount = (int)(amount * 100);

// Find the number of one dollars
int numberOfOneDollars = remainingAmount / 100;
remainingAmount = remainingAmount % 100;

// Find the number of quarters in the remaining amount
int numberOfQuarters = remainingAmount / 25;
remainingAmount = remainingAmount % 25;

// Find the number of dimes in the remaining amount
int numberOfDimes = remainingAmount / 10;
remainingAmount = remainingAmount % 10;

// Find the number of nickels in the remaining amount
int numberOfNickels = remainingAmount / 5;
remainingAmount = remainingAmount % 5;

// Find the number of pennies in the remaining amount
int numberOfPennies = remainingAmount;
```

remainingAmount	56
numberOfOneDollars	11
numberOfOneQuarters	2

numberOfOneQuarters 저장

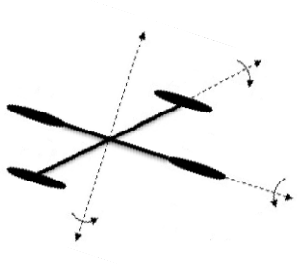



Amount는 11.56
int remainingAmount = (int)(amount * 100);
// Find the number of one dollars
int numberOfOneDollars = remainingAmount / 100;
remainingAmount = remainingAmount % 100;
// Find the number of quarters in the remaining amount
int numberOfQuarters = remainingAmount / 25;
remainingAmount = remainingAmount % 25;
// Find the number of dimes in the remaining amount
int numberOfDimes = remainingAmount / 10;
remainingAmount = remainingAmount % 10;
// Find the number of nickels in the remaining amount
int numberOfNickels = remainingAmount / 5;
remainingAmount = remainingAmount % 5;
// Find the number of pennies in the remaining amount
int numberOfPennies = remainingAmount;
remainingAmount
6
numberOfOneDollars
11
numberOfQuarters
2
remainingAmount 갱신

예제: 현재 시각 표시하기

이 절에서는 현재 시간을 GMT(Greenwich Mean Time) 기준으로 하여 13:19:8과 같이 시(hour):분(minute):초(second) 형식으로 출력하는 프로그램을 살펴본다.

time(0) 함수는 ctime 헤더 파일에 있는 함수이며, 1970년 1월 1일(GMT)을 00:00:00으로 하여 현재 시각까지 지난 시간을 초(second)로 반환한다(그림 2.1). 이 시간은 UNIX 기준(epoch) 시간이라고도 하는데 이때 UNIX 시스템이 공식적으로 출시되었기 때문이다.



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

int main()
{
  // Obtain the total seconds since the midnight, Jan 1, 1970
  int totalSeconds = time(0);

  // Compute the current second in the minute in the hour
  int currentSecond = totalSeconds % 60;

  // Obtain the total minutes
  int totalMinutes = totalSeconds / 60;

  // Compute the current minute in the hour
  int currentMinute = totalMinutes % 60;

  // Obtain the total hours
  long totalHours = totalMinutes / 60;

  // Compute the current hour
  int currentHour = (int)(totalHours % 24);

  // Display results
  cout << "Current time is " << currentHour << ":"
    << currentMinute << ":" << currentSecond << " GMT" << endl;

  return 0;
}
```