C++ 스터디

1. 데이터 타입, 변수, 연산자, 표준 입출력

D.Com

C++의 가장 큰 특징

- C++의 Python과 가장 대조되는 큰 특징은 변수를 선언할 때 미리 변수의 타입을 선언해야 한다는 것이다! 한번 C++에서 다루는 데이터 타입에 대해서 알아보자!
- 컴파일 언어와 스크립트 언어에 대해서는 저번 장 참고

어떤 타입을 넣던 알아서
처리해주는게 편하지 않아?

나는 컴파일 언어기 때문에
어떤 타입의 데이터가 오는지
알아야지 처리하기 편하고
빨라 ——

D.Com

변수

6번째 줄 int x;

- 선언문이다. C++의 모든 변수들은 선언한 후 사용 되어야한다.
- 선언문을 통해 변수의 타입을 정한다. (int, double, char 등)
- Int는 integer의 약자로 변수가 정수임을 나타낸다.
- 그 정수 변수의 이름은 x이다.

7번째 줄 x = 10;

- 할당문이다. 부여문은 변수에 값(value)를 할당 시킨다.
- 할당문의 핵심은 '=' 으로 assignment operator이다.
- 여기서는 변수 x에 value 10(int 타입)을 할당하였다.
- 이것은 value 10이 변수 x를 위한 컴파일러의 메모리 공간에 저장된다는 것이다.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int x;
7      x = 10;
8  }
```

D.Com

C++ 데이터 타입 (정수)

정수형 타입	할당되는 메모리의 크기	데이터의 표현 범위
(signed) short	2 바이트(16비트)	-2^{15}~$(2^{15}-1)$
unsigned short	2 바이트	0~$(2^{16}-1)$
(signed) int	4 바이트	-2^{31}~$(2^{31}-1)$
unsigned int	4 바이트	0~$(2^{31}-1)$

만약 입력한 수가 데이터의 표현 범위를 넘어 간다면, overflow(수가 표현 범위보다 큰 것, 왼쪽) 혹은 underflow(수가 표현 범위보다 작은 것, 오른쪽)이 발생한다.

N비트당 2의 n제곱 만큼을 표현할 수 있다.

```
수를 입력해 주세요 : 999999999999
입력한 수는 -858993460 입니다!
```

```
수를 입력해 주세요 : -999999999999
입력한 수는 -858993460 입니다!
```

D.Com

C++ 데이터 타입 (실수)

실수형 타입	할당되는 메모리의 크기	데이터의 표현 범위
float	4 바이트	$(3.4 \times 10^{-38}) \sim (3.4 \times 10^{38})$
double	8 바이트	$(1.7 \times 10^{-308}) \sim (1.7 \times 10^{308})$
long double	double형과 동일함.	double형과 동일함.

실수형 타입	지수의 길이	가수의 길이	유효 자릿수
float	8 비트	23 비트	소수 부분 6자리까지 오차없이 표현할 수 있음.
double	11 비트	52 비트	소수 부분 15자리까지 오차없이 표현할 수 있음.

D.Com

C++ 데이터 타입 (문자)

- char 타입은 1 문자 를 표현하기 위한 데이터 타입으로 character(문자)의 약자이다.
- 알파벳과, 특수문자, 10진수(0~9), 개행문자('₩n' ,'₩0' 등)을 포함하고 있다.
- 대부분은 American Standard Code for Information Interchange(ASCII) - 아스키 코드를 지원한다. (걍 아스키 코드를 베이스로 두고있다는 뜻)
- 아스키 코드는 256개의 문자표로 숫자(0~255)가 들어오면 해당하는 문자로 바꿔 보여준다 생각하면 된다.

```
1   #include <iostream>
2
3   using namespace std;
4
5   int main() {
6       char x;
7       x = 65; // 65가 'A'임
8       char y;
9       y = 'A'; // char타입은 작은따옴표!
10      cout << x << ',' << y << endl;
11  }
12
```

Microsoft Visual Studio 디버그 콘솔

```
A,A

C:₩Users₩bluej₩Desktop₩방학프로그래밍₩
 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

D.Com

아스키 코드 표 (128~255 생략)

0~9 숫자와
알파벳 A
알파벳 a
기억해두면
쓸모가 많음

0	*null*	16		32	*space*	48	0	64	@	80	P	96	`	112	p
1		17		33	!	49	1	65	A	81	Q	97	a	113	q
2		18		34	"	50	2	66	B	82	R	98	b	114	r
3		19		35	#	51	3	67	C	83	S	99	c	115	s
4		20		36	$	52	4	68	D	84	T	100	d	116	t
5		21		37	%	53	5	69	E	85	U	101	e	117	u
6		22		38	&	54	6	70	F	86	V	102	f	118	v
7	*bell*	23		39	'	55	7	71	G	87	W	103	g	119	w
8	*backspace*	24		40	(	56	8	72	H	88	X	104	h	120	x
9	*tab*	25		41	)	57	9	73	I	89	Y	105	i	121	y
10	*newline*	26		42	*	58	:	74	J	90	Z	106	j	122	z
11		27		43	+	59	;	75	K	91	[	107	k	123	{
12	*form feed*	28		44	,	60	<	76	L	92	\	108	l	124	|
13	*return*	29		45	-	61	=	77	M	93	]	109	m	125	}
14		30		46	.	62	>	78	N	94	^	110	n	126	~
15		31		47	/	63	?	79	O	95	_	111	o	127	

D.Com

string vs char

타입명	메모리 크기	데이터 유형	예시
char	1 바이트(8 비트)	문자 단 하나	‘a’, ‘1’, ‘.’, ‘y’
string	안정해짐	여러 문자	“안녕하세용~”

- “A”와 ‘A’는 타입도 다를 뿐더러 “A”는 사실 문자열의 종료를 알리는 ‘₩0’가 감춰져 있어 2바이트의 데이터가 된다.
- string(문자열)은 여러 개의 char로 이루어져 있어 한 개의 char 변수에 담을 수 없다.
- char은 작은 따옴표로 표시하고 string(문자열)은 큰 따옴표로 표시한다.
- char 타입의 특성을 응용하여 C++의 대소문자 변환, 문자열에 int 넣기 등이 가능하다.

D.Com

C++ 데이터 타입 (논리)

- bool 타입도 지원한다!

타입명	메모리 크기	데이터 유형	예시
bool	1 비트	논리형	true, false

D.Com

상수

- 상수는 변수와 다르게 변하지 않는 수를 의미한다.
- 편의성을 위해서 특정 상수에 이름을 붙여서 선언할 수 있다.

  const double PI = 3.14159;
- const 라는 키워드는 변경이 불가능하게 선언하겠다는 것이다.
- 한번 선언되면 수정이 불가능 해진다.

  PI = 2.5; (compile error)
- 상수는 선언문 밖에서 수정하는 것이 불가능하기에, 모든 상수는 선언문에서 초기화되어야 한다.
- 보통 상수는 모두 대문자로 작성한다.

```
1   #include <iostream>
2
3   using namespace std;
4
5   int main() {
6       const double PI = 3.14159;
7       double temp = 0;
8
9       temp = PI;
10  }
11
```

D.Com

타입 변환

- Python에서 했던 것처럼 데이터 변환을 할 수는 있다.

  (하지만, C에서 int, char, float 등은 객체가 아니므로, 생성자를 사용하는 것이 아니다.)

- 묵시적 타입 변환
  - 값의 대입이 일어날 때 : 데이터 손실이 있지만 변환 가능

  (char형 → short형 → int형 → long형 → float형 → double형 → long doubl형)

```
#include <iostream>

using namespace std;

int main() {
    int x = 'a'; //아스키 코드 'a'는 97
    cout << x << endl;
}
```

```
Microsoft Visual Studio 디버그 콘솔
97

C:₩Users₩bluej₩Desktop₩방학프로그래밍₩
 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

```
#include <iostream>

using namespace std;

int main() {
    char x = 'D';
    double y = x + 0.5;
    int z = y;
    cout << y << ',' << z << endl;
}
```

```
Microsoft Visual Studio 디버그 콘솔
68.5,68

C:₩Users₩bluej₩Desktop₩방학프로그래밍
 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

D.Com

타입 변환

- 명시적 타입 변환
  - 타입 캐스트(cast) 연산자를 이용, 강제적으로 수행하는 타입 변환

1. (변환 할 타입) 변환 할 데이터 - C언어와 C++ 둘 다 사용 가능함.
2. 변환 할 타입 (변환 할 데이터) - C++에서만 사용 가능함.

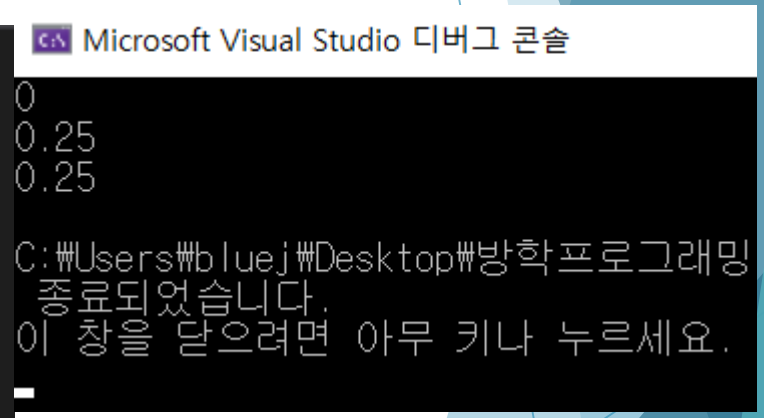
```cpp
#include <iostream>

using namespace std;

int main() {
    int num1 = 1;
    int num2 = 4;

    double result1 = num1 / num2; //int int 연산
    double result2 = (double)num1 / num2; // double int 연산
    double result3 = double(num1) / num2; // double int 연산

    cout << result1 << endl;
    cout << result2 << endl;
    cout << result3 << endl;
}
```



D.Com

연산자

- 연산자가 기본적으로 무엇인지는 알 테니 기호만 설명함. (연산자)= 은 복합 대입 연산자이다. (예 : int i = 1 / i += 3 / i == 4(true))

연산자	연산 내용	예시
+	덧셈	3 + 2 = 5
-	뺄셈	3 - 2 = 1
/	나눗셈	3 / 2 = 1
*	곱셈	3 * 2 = 6
%	나머지 연산	3 % 2 = 1

expression	equivalent to…
y += x;	y = y + x;
x -= 5;	x = x - 5;
x /= y;	x = x / y;
price *= units + 1;	price = price * (units + 1);

연산자

연산자	연산 내용	예시
==	같다	int i = 1 / ++i == 2 // true
!=	다르다	int i =1 / i++ != 1 // false
〉	크다	3 〉 2 == true
〉=	크거나 같다	3 〉= 3 == true
〈	작다	2 〈 3 == true
〈=	작거나 같다	3 〈= 3 == true
!	부정	!true == false

연산자	연산 내용	예시
++i	전위 연산자	int i = 1 / ++i == 2 // true
i++	후위 연산자	int i = 1 / i++ == 1 // true (그 다음 행에서 2)
--i	전위 연산자	int i = 2 / --i == 1 // true
i--	후위 연산자	int i = 2 / i-- == 2 // true(그 다음 행에서 1)

D.Com

연산자

연산자	연산 내용	예시
&&	AND	true && true == true false && true == false false && false == false
||	OR	true || true == true false || true == true false || false == false

연산자 (예제)

- examples

```
// assignment operator
#include <iostream>
using namespace std;
int main()
{
    int a, b; // a:? , b:?
    a = 10; // a:10, b:?
    b = 4; // a:10, b:4
    a = b; // a:4, b:4
    b = 7; // a:4, b:7
    cout << "a:";
    cout << a;
    cout << " b:";
    cout << b;
}
```

```
// compound assignment operators
#include <iostream>
using namespace std;
int main()
{
    int a, b = 3;
    a = b;
    a += 2; // equivalent to a=a+2
    cout << a;
}
```

```
// conditional operator
#include <iostream>
using namespace std;
int main()
{
    int a, b, c;
    a = 2;
    b = 7;
    c = (a > b) ? a : b;
    cout << c << '\n';
}
```

D.Com

표준 입출력 (출력)

- cout ⟨⟨ “Please enter two integer values : ”;

  전 장에서 봤던 이러한 문장이 보통의 출력 문장이다.

  ⟨⟨의 기능을 파이썬의 쉼표와 비슷하게 이해해도 된다.

D.Com

표준 입출력 (입력)

- cin 〉〉 value1;

  마찬가지로 이런 문장을 통해 엔터를 누르기까지의 키보드 입력을 받을 수 있다.

  엔터를 누르게 되면 값이 변수(value1)에 할당된다.

- cin 〉〉 value1 〉〉 value2;

  이 문장은 2개의 숫자를 입력 받는다.

  (숫자 1개 → 엔터 → 숫자 1개 → 엔터 or 숫자 1개 → 공백 → 숫자 1개 → 엔터)

  첫번째 숫자는 value1에 할당되고 두번째 숫자는 value2에 할당된다.

D.Com

표준 입출력 (예시)

```
#include <iostream>
using namespace std;
int main() {
    int value1, value2, sum;
    cout << "Please enter two integer values: ";
    cin >> value1 >> value2;
    sum = value1 + value2;
    cout << value1 << " + " << value2 << " = " << sum << '₩n';
}
```

```
#include <iostream>
int main() {
    int value1, value2, sum;
    std::cout << "Please enter two integer values: ";
    std::cin >> value1 >> value2;
    sum = value1 + value2;
    std::cout << value1 << " + " << value2 << " = " << sum << '₩n';
}
```

D.Com

과제 1번

- 랩 #1번을 풀어서 제출하세요.
- 제출시에는 cpp 파일만!

D.Com