Data Type Casting

Data Type Casting (자료형의 변환, 형변환)

! 데이터의 표현을 새롭게 다시표현하는 것, 표현방법을 완전히 뒤바꾸는 것

Data Type Casting - Auto(자동)

```
int main(String[] args) {
	/*
		byte, short형 상수는 존재하지 않는다.
		그러나 상황에 따라서 예외적으로 byte, short로 자동형변환을 허용한다.
		(long을 표현하려면 접미사 L을 사용해야 한다.)
	*/
	short num1 = 10;
	short num2 = 20;

	/*
		CPU는 int형 연산밖에 하지 못하므로, num1, num2는 int로 자동형변환 된다.
	*/
	short result = num1 + num2;
	....
}
```

연산 시 자료형을 일치시켜야 하는 이유?

- 1 + 2 = 3

```
00000000 00000000 00000000 00000001 1
           +
00000000 00000000 00000000 00000010 2
-----------------------------------------------
00000000 00000000 00000000 00000011 3
```

- 1 + 1.0 = ????? (자료형이 다르면 CPU는 연산 불가능)

```
00000000 00000000 00000000 00000001 1
           +
00111111 10000000 00000000 00000010 1.0
-----------------------------------------------
???????? ???????? ???????? ???????? ?
```

- CPU에게 연산을 시키려면 자료형을 일치시켜야 한다.
  서로 다른 자료형이 있으면 Java가 일치시켜 준다. (형변환은 Java가 수행)

Auto Data Type Casting Rull

byte → short →
int → long → float → double
char →

- double num2 = 3.5f + 12; // int 12 → float 12 Auto Casting

i. **정수 ↔ 정수 형변환**
(byte)00000001 → (int)00000000 00000000 00000000 00000001

ii. **실수 ↔ 정수 형변환**
- **실수 → 정수 : 3.1435 → 3 (소수부손실, 특히 주의!!!!!)**
- 정수 → 실수 : 3 → 3.0

Data Type Casting - Explicit(명시적)

(Data Type) : 형변환연산자

- **자동 형 변환의 발생지점을 표시하기 위해서**
- **자동 형 변환 규칙에 위배되지만 변환이 필요한 상황**
  (2.4(double) x 3(int) → 2(int) x 3(int)으로 표현하고 싶을 때)

```
long num1 = 2147483648L;
int num2 = (int)num1;

int num3 = 100;
long num4 = (long)num3;
```
