# Number (Typecasting)

Sungchul Lee

# Learning Object

- Typecasting
  1. **Implicit** casting
  2. Explicit casting
- Casting Conversions
  - Explicit casing

# Typecasting

- You can always assign a value to a numeric variable whose type supports a larger range of values. You **cannot**, however, assign a value to a variable of a type with smaller range unless you use **type casting**.
  - Variable value can be changed
  - Data type also can be changed

# Typecasting

- Typecasting: a process that **converts** a value of one data type to another data type
  1. **Implicit** casting
  2. Explicit casting

# Implicit Casting

- Number is converted from a lower to a higher precision naturally
  - Higher precision: a data type with a larger range of values
    - *double has a higher precision than float*
    - *long has a higher precision than int*
    - *Floating point number vs. integer number?*

# Implicit Casting

- In class exercise on implicit casting
  - float x1 = 3.2F;
  - double x2 = x1;
  - int y1 =5;
  - double y2 = y1;
- **Implicit casting is done automatically**
  - That’s why it is also called promotion
  - You may not care about the implicit casting case

# Explicit Casting

- Explicit casting can help convert a high precision number into a lower precision number
  - Also called “forced casting”
  - Grammar: (<data type>) <expression>
  - Example:

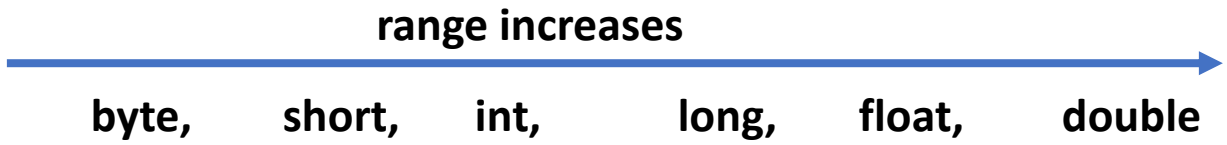
```
double result = 3.6;
int iResult =0;
iResult = result; //Error result bigger
iResult = (int)result;
```

  - The iResult value will be 3

# Casting Conversions

**Casting** is an operation that converts a value of one data type into a value of another data type.

- **Casting** a variable of a type with a **small range** to a variable of a type with a **larger range** is known as **widening** a type.
- **Casting** a variable of a type with a **large range** to a variable of a type with a **smaller range** is known as **narrowing** a type.




**Widening** a type can be preformed automatically without explicit casting.
**Narrowing** a type must be preformed explicit.

# Type Conversion for Explicit Casting

- A variable of type A can be assigned a value of type B even if it isn't a widening conversion if it is compatible and you use a type case.
  - Example:
  - int x = (int)3.14;  //assigns 3 into x
  - double y = 256;  //assign 256.0 into y
- When a floating-point value is cast to an integer type, the fractional component will be lost due to truncation.

# Practice

1. Make a new project (Reference: Create Project and Class File)
   - ❑Project name: Typecasting
2. Create a new Class File
   - ❑Class name: Typecasting
3. Coding:

```
public class Typecasting {
	public static void main(String[] args) {
	// TODO Auto-generated method stub
		double result = 3.6;
		int iResult =0;
		//iResult = result; //Error result bigger
		iResult = (int)result;
		System.out.println(result); //3.6
		System.out.println(iResult); //3

		int x = (int)3.14;//Explicit
		double y = x; //Implicit
		System.out.println(x); //3
		System.out.println(y); //3.0
		}
}
```

# Practice – Code and Result

Typecasting.java 1

G: > 내 드라이브 > 00. Class > UWW > CS172-java > Code 2019 > Data Type >

```java
public class Typecasting {

    Run | Debug
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        double result = 3.6;
        int iResult =0;
        //iResult = result; //Error result bigger
        iResult = (int)result;
        System.out.println(result); //3.6
        System.out.println(iResult); //3

        int x = (int)3.14;  //Explicit casting
        double y = x;   // Implicit casting
        System.out.println(x); //3
        System.out.println(y); //3.0
    }
}
```

Result

```
3.6
3
3
3.0
```

# Summary

- Typecasting
  1. **Implicit** casting
  2. Explicit casting
- Casting Conversions
  - Explicit casing

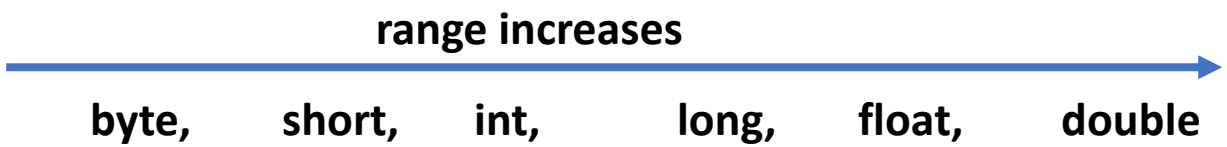
Typecasting.java 1

G: > 내 드라이브 > 00. Class > UWW > CS172-java > Code 2019 > Data Type >

```java
public class Typecasting {

    Run | Debug
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        double result = 3.6;
        int iResult =0;
        //iResult = result; //Error result bigger
        iResult = (int)result;
        System.out.println(result); //3.6
        System.out.println(iResult); //3

        int x = (int)3.14;  //Explicit casting
        double y = x;   // Implicit casting
        System.out.println(x); //3
        System.out.println(y); //3.0
    }
}
```




**Widening** a type can be preformed automatically without explicit casting.
**Narrowing** a type must be preformed explicit.