Ministry of Higher Education
Jami University
Scientific assistantship
Faculty (Computer Science)
Department of Information Systems

Type conversion in java

1. **Implicit (Automatic) Type Conversion** (Widening Conversion):
   - Occurs when a smaller data type is converted into a larger data type automatically by Java.
   - No data loss occurs in widening conversions.
   - For example: `byte` to `int`, `int` to `double`.

2. **Explicit Type Conversion** (Casting or Narrowing Conversion):
   - Required when a larger data type is converted into a smaller data type.
   - Potential loss of data or precision, so you need to specify the type conversion explicitly.
   - For example: `double` to `int`, `long` to `short`.

Syntax for Explicit Type Conversion:

```java
dataType variableName = (dataType) value;
```

10 Examples of Type Conversion in Java:

1. **Implicit Conversion: `int` to `double`**

   ```java
   int num = 10;
   double result = num;  // Implicit conversion
   System.out.println(result);  // Output: 10.0
   ```

2. **Implicit Conversion: `char` to `int`**

   ```java
   ```

Saifullah Haidari

```
char letter = 'A';
int ascii = letter;  // Implicit conversion
System.out.println(ascii);  // Output: 65
```

3. **Implicit Conversion: `float` to `double`**

```java
float pi = 3.14f;
double largePi = pi;  // Implicit conversion
System.out.println(largePi);  // Output: 3.14
```

4. **Explicit Conversion: `double` to `int`**

```java
double temperature = 36.6;
int temp = (int) temperature;  // Explicit casting
System.out.println(temp);  // Output: 36
```

5. **Explicit Conversion: `long` to `int`**

```java
long largeNumber = 100000L;
int number = (int) largeNumber;  // Explicit casting
System.out.println(number);  // Output: 100000
```

6. **Explicit Conversion: `double` to `float`**

```java
double bigValue = 123.456;
float smallValue = (float) bigValue;  // Explicit casting
System.out.println(smallValue);  // Output: 123.456
```

7. **Implicit Conversion: `byte` to `int`**

   ```java
   byte smallNum = 20;
   int bigNum = smallNum;  // Implicit conversion
   System.out.println(bigNum);  // Output: 20
   ```

8. **Explicit Conversion: `int` to `byte`**

   ```java
   int number = 150;
   byte smallNumber = (byte) number;  // Explicit casting
   System.out.println(smallNumber);  // Output: -106 (Data loss due to overflow)
   ```

9. **Implicit Conversion: `short` to `int`**

   ```java
   short shortValue = 1000;
   int intValue = shortValue;  // Implicit conversion
   System.out.println(intValue);  // Output: 1000
   ```

10. **Explicit Conversion: `float` to `int`**

    ```java
    float height = 5.9f;
    int intHeight = (int) height;  // Explicit casting
    System.out.println(intHeight);  // Output: 5 (Fractional part is lost)
    ```

Key Points:

- **Widening conversions** (e.g., `int` to `double`) are safe because they involve no loss of precision.
- **Narrowing conversions** (e.g., `double` to `int`) may result in loss of data or precision, and thus require explicit casting.
- Implicit conversions happen automatically when Java finds it safe to do so.

Saifullah Haidari

Type conversion is crucial when working with different data types in Java, as it ensures flexibility while minimizing errors related to data manipulation.

Saifullah Haidari