Type Casting in Java

In Java, **type casting** is a method or process that converts a data type into another data type in both ways manually and automatically. The automatic conversion is done by the compiler and manual conversion performed by the programmer. In this section, we will discuss **type casting** and **its types** with proper examples.

Type casting

Convert a value from one data type to another data type is known as **type casting**.

Types of Type Casting

There are two types of type casting:

- Widening Type Casting
- Narrowing Type Casting

Widening Type Casting

Converting a lower data type into a higher one is called **widening** type casting. It is also known as **implicit**

conversion or **casting down**. It is done automatically. It is safe because there is no chance to lose data. It takes place when:

- Both data types must be compatible with each other.
- The target type must be larger than the source type.

1. **byte** -> **short** -> **char** -> **int** -> **long** -> **float** -> **double**

For example, the conversion between numeric data type to char or Boolean is not done automatically. Also, the char and Boolean data types are not compatible with each other. Let's see an example.

WideningTypeCastingExample.java

```
1.  public class WideningTypeCastingExample
2.  {
3.  public static void main(String[] args)
4.  {
5.  int x = 7;
6.  //automatically converts the integer type into long type
7.  long y = x;
8.  //automatically converts the long type into float type
9.  float z = y;
10. System.out.println("Before conversion, int value "+x);
11. System.out.println("After conversion, long value "+y);
12. System.out.println("After conversion, float value "+z);
```

13. }
14. }

Output

```
Before conversion, the value is: 7
After conversion, the long value is: 7
After conversion, the float value is: 7.0
```

In the above example, we have taken a variable x and converted it into a long type. After that, the long type is converted into the float type.

Narrowing Type Casting

Converting a higher data type into a lower one is called **narrowing** type casting. It is also known as **explicit conversion** or **casting up**. It is done manually by the programmer. If we do not perform casting then the compiler reports a compile-time error.

1. **double** -> **float** -> **long** -> **int** -> **char** -> **short** -> **byte**

Let's see an example of narrowing type casting.

In the following example, we have performed the narrowing type casting two times. First, we have converted the double type into long data type after that long data type is converted into int type.

NarrowingTypeCastingExample.java

```
1.  public class NarrowingTypeCastingExample
2.  {
3.  public static void main(String args[])
4.  {
5.  double d = 166.66;
6.  //converting double data type into long data type
7.  long l = (long)d;
8.  //converting long data type into int data type
9.  int i = (int)l;
10. System.out.println("Before conversion: "+d);
11. //fractional part lost
12. System.out.println("After conversion into long type: "+l);
13. //fractional part lost
14. System.out.println("After conversion into int type: "+i);
15. }
16. }
```

Output

```
Before conversion: 166.66
After conversion into long type: 166
After conversion into int type: 166
```

Example 1

```
1.  public class IntegerFloatValueExample1 {
2.     public static void main(String[] args) {
3.        //Input Number with Integer DataType and Double Da
    taType
4.           Integer a = new Integer(2454);
```

```
5.          Double b = new Double(86549876);
6.         //Print the Integer values as float
7.          System.out.println("Value : "+a.floatValue());
8.          System.out.println("Value : "+b.floatValue());
9.      }
10.  }
```

Output:

```
Value : 2454.0
Value : 8.6549872E7
```

Example 2

```
1.  import java.util.Scanner;
2.  public class IntegerFloatValueExample2 {
3.     public static void main(String[] args) {
4.        Scanner readInput = new Scanner(System.in);
5.        System.out.println("Enter The Input Values: ");
6.        //Input Number with Integer DataType and Double Da
    taType
7.          Integer a = readInput.nextInt();
8.          Double b = readInput.nextDouble();
9.          readInput.close();
10.         //Print the Integer values as float
11.         System.out.println("Value : "+a.floatValue());
12.         System.out.println("Value : "+b.floatValue());
13.       }
```

```
14.  }
```

Output:

```
Enter The Input Values:
2343543
34546658233654
Value : 2343543.0
Value : 3.45466575E13
```

Example 3

```
1.  public class IntegerFloatValueExample3 {
2.     public static void main(String[] args) {
3.        int x = 3;
4.         int y = 2;
5.         Float f = new Float(x);
6.         float result = f.floatValue()/y;
7.         System.out.println("Value is = "+result);
8.        }
9.   }
```

Output:

```
Value is = 1.5
```

Example 4

```
1.  import java.util.Scanner;
```

```
2.  public class IntegerFloatValueExample4 {
3.     public static void main(String[] args) {
4.        Scanner readInput = new Scanner(System.in);
5.        System.out.print("Enter The Desired Input Value: ");

6.        //Input Number with Integer DataType`
7.           Integer a = readInput.nextInt();
8.           readInput.close();
9.           float f = a.floatValue();
10.          //Print the Integer values as float
11.          System.out.println("The Float Value is: "+f);

12.       }
13.  }
```

Output:

```
Enter The Desired Input Value: 46567
The Float Value is: 46567.0
```

Java Convert String to int

We can convert **String to an int in java** using *Integer.parseInt()* method. To convert String into Integer, we can use *Integer.valueOf()* method which returns instance of Integer class.

Scenario

It is generally used if we have to perform mathematical operations on the string which contains a number. Whenever we receive data from TextField or TextArea, entered data is received as a string. If entered data is in number format, we need to convert the string to an int. To do so, we use Integer.parseInt() method.

Signature

The parseInt() is the static method of Integer class. The **signature** of parseInt() method is given below:

```
1. public static int parseInt(String s)
```

Java String to int Example: Integer.parseInt()

Let's see the simple code to convert a string to an int in java.

```
1. int i=Integer.parseInt("200");
```

Let's see the simple example of converting String to int in Java.

```java
//Java Program to demonstrate the conversion of String int o int
//using Integer.parseInt() method
public class StringToIntExample1{
public static void main(String args[]){
//Declaring String variable
String s="200";
//Converting String into int using Integer.parseInt()
int i=Integer.parseInt(s);
//Printing value of i
System.out.println(i);
}}
```

Output:

```
200
```

Understanding String Concatenation Operator

```java
//Java Program to understand the working of string concat enation operator
public class StringToIntExample{
public static void main(String args[]){
//Declaring String variable
String s="200";
//Converting String into int using Integer.parseInt()
int i=Integer.parseInt(s);
System.out.println(s+100);//200100, because "200"+100, here + is a string concatenation operator
```

```
9. System.out.println(i+100);//300, because 200+100, here +
   is a binary plus operator
10. }}
```

Output:

```
200100
300
```

Java String to Integer Example: Integer.valueOf()

The Integer.valueOf() method converts String into Integer object. Let's see the simple code to convert String to Integer in Java.

```
1. //Java Program to demonstrate the conversion of String int
   o Integer
2. //using Integer.valueOf() method
3. public class StringToIntegerExample2{
4. public static void main(String args[]){
5. //Declaring a string
6. String s="200";
7. //converting String into Integer using Integer.valueOf() met
   hod
8. Integer i=Integer.valueOf(s);
9. System.out.println(i);
10. }}
```

Output:

```
300
```

NumberFormatException Case

If you don't have numbers in string literal, calling Integer.parseInt() or Integer.valueOf() methods throw NumberFormatException.

```
1. //Java Program to demonstrate the case of NumberFormat Exception
2. public class StringToIntegerExample3{
3. public static void main(String args[]){
4. String s="hello";
5. int i=Integer.parseInt(s);
6. System.out.println(i);
7. }}
```

Output:

```
Exception in thread "main" java.lang.NumberFormatException: For input string: "hello"
        at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
```

```
        at
java.base/java.lang.Integer.parseInt(Integer.java:6
52)
        at
java.base/java.lang.Integer.parseInt(Integer.java:7
70)
        at
StringToIntegerExample3.main(StringToIntegerExample
3.java:4)
```

References

1. Integer.parseInt() JavaDoc
2. Integer.valueOf() JavaDoc
3. NumberFormatException JavaDoc

Java Convert int to String

We can convert **int to String in java** using *String.valueOf()* and *Integer.to*
can use *String.format()* method, string concatenation operator etc.

Scenario

It is generally used if we have to display number in textfield because ev
form.

---

1) String.valueOf()

The String.valueOf() method converts int to String. The valueOf() is
The **signature** of valueOf() method is given below:

```
public static String valueOf(int i)
```

Java int to String Example using String.valueOf()

Let's see the simple code to convert int to String in java.

```
int i=10;
String s=String.valueOf(i);//Now it will return "10"
```

Let's see the simple example of converting String to int in java.

```
public class IntToStringExample1{
public static void main(String args[]){
int i=200;
String s=String.valueOf(i);
System.out.println(i+100);//300 because + is binary plus operator
System.out.println(s+100);//200100 because + is string concatenation op
}}
```

Output:

```
300
200100
```

2) Integer.toString()

The Integer.toString() method converts int to String. The toString() is
The **signature** of toString() method is given below:

```
public static String toString(int i)
```

Java int to String Example using Integer.toString()

Let's see the simple code to convert int to String in java using Integer.toS

```
int i=10;
String s=Integer.toString(i);//Now it will return "10"
```

Let's see the simple example of converting String to int in java.

```
public class IntToStringExample2{
public static void main(String args[]){
int i=200;
String s=Integer.toString(i);
System.out.println(i+100);//300 because + is binary plus operator
System.out.println(s+100);//200100 because + is string concatenation op
}}
```

Output:

```
300
200100
```

3) String.format()

The String.format() method is used to format given arguments into String

```
public static String format(String format, Object... args)
```

Java int to String Example using String.format()

Let's see the simple code to convert int to String in java using String.form

```
public class IntToStringExample3{
public static void main(String args[]){
int i=200;
String s=String.format("%d",i);
System.out.println(s);
}}
```

Output:

```
200
```