Central India's Most Trusted Training Institute

**Type Conversion :**

Java provides the facility of mixing different types of variables and constants in an expression. In these types of operations data type of one operand is converted into data type of another operand. This is known as type conversion. The different types of type conversion are

**Implicit type conversions are done by the compiler while the explicit type conversions are user definecd conversions.**

*Implicit Type Conversions*

These conversions are done by the Java compiler according to some predefined rules of java language.

1. Automatic Conversions

*Automatic unary conversions :*

All operands of type byte, short and char will be converted to **int** before any operation.

*Automatic binary conversions:*

The rules for automatic binary conversions are as
(i) If one operand is double, then the other will be converted to double, and the result will be double
(ii) Otherwise if one operand is float, the other will be converted to float, and the result will be float.
(iii)Otherwise If one operand is long, then the other will be converted to long , and the result will be long
(iv) Otherwise if one operand is int, then other will be converted to int, and the result will int.

2. Type Conversion In Assignment.

If the types of the two operands in an assignment expression are different, then the type of the right hand side operand is converted to the type of left hand operand. Here if the right hand operand is of lower rank then it will be promoted to the rank of left hand operand.

## Widening Casting(Implicit type casting)

## byte->short->int->long->float->double
## char↗

**Explicit Type Conversion Or Type Casting**

There may be certain situations where implicit conversions may not solve our purpose.
For example    float z;
        int x = 20, y = 3;

z = x/y;

The value of z will be 6.0 instead of 6.66.

In these types of cases we can specify our own conversions known as explicit type casting or coercion. This is done with the help of cast operator. The cast operator is a unary operator that is used for converting in an expression to a particular data type temporarily. The expression can be any constant or variable

The syntax of cast operator is

**(datatype) expression**

Here the datatype along with the parentheses is called the cast operator.,
So if we write the above statement as
z = (float)x/y;
Now the value of z will come out be 6.66.

This happens because the cast operator (float) temporarily converted the int variable x into float type and so floating point arithmetic took place and fraction part was not lost.

Note that the cast operator changes the data type of variable x only temporarily for the evaluation , this expression, everywhere else in the program it will be an int variable only.

Program to illustrate the use of cast operator
```
class ExplicitTypeCasting
{
 public static void main (String[] args )
 {
   int x=5,y=2;
   float p,q;
   p=x/y;
 System.out. printf("p = %f\n",p);
   q=(float)x/y;
 System.out. printf ("q = %f\n",q);
 }
}
```

Output: p = 2.000000 q = 2.500000

Initially the expression x/y is evaluated, both x and yare integers so according to integer arithmetic after division, decimal value is truncated and result is integer value 2. This value will be assigned to p but p is a float variable so according to implicit type conversion in assignment the integer value 2 will be converted to float and then assigned to p. So finally the value of p is 2.0

When cast operator is used, floating point arithmetic is performed hence the value of q is 2.5

Here are some other examples of usage of cast operator

(int)20.3      constant 20.3 converted to integer type and fractional part is lost(Result 20)

(float)20/3      constant 20 converted to float type, and then divided by 3 (Result 6.66)

(float)(20/3)      First 20 divided by 3 and then result of whole expression converted to float type(Result 6.00) .

(double)( x +y -z) Result of expression x+y-z is converted to double

(double)x+y-z      First x is converted to double and then used in expression

## Programming Exercise

- Enter'the temperature in Celsius and convert that into Fahrenheit.
- Accept the radius of the circle and calculate the area and perimeter of the circle.
- Write a program to accept the number in decimal and print the number in octal and hexadecimal.
- Accept any five digit number and print the value of remainder after dividing by 3.
- Accept any two numbers, if the first number is greater than second then print the sum of these two numbers, otherwise print their difference.
- Write this program using ternary operator. Write a program to accept the principal, rate, and number of years and find out the simple interest.
- Accepts marks in five subject and calculate the total percentage marks.