Day 12

char a = 'a';

M T W T F S S
Page No.:
Date: 21/01/25
YOUVA

Narrowing: (Explicitaly)

- It is process of converting / storing larger type data into smaller type.
- Narrowing cannot be implicitaly by compiler because there could be possibility of loss of data
- Narrowing must be done explicitaly by the programmer using type-cast operator.

Typecast operator:

Typecast operator is a unary operator

Syntax:

(datatype to be converted) value;

Eg:→

```
class Narrowing
{
    public static void main(String args[])
    {
        System.out.println((int)20.31);  // 20
        System.out.println((char)120);   // x
        int i = (int) 20.34;
        System.out.println(i);           // 20

        char ch = (char) 56.46 64F
        System.out.println(ch);          // 8
        byte b = (byte) 2001;
        System.out.println(b)            // -56
        byte bt = -100;
        char c = (char) bt;
        System.out.println(c);           // ?
    }
}
```

eyeshi

M T W T F S S
Page No.: YOUVA
Date:

Assignment operator continued:

If the type of value is not same as type of variable there are two possibilities.

① If the type of value is smaller than the type of variable. ⇒
It performs widening

② If the type of value is bigger, larger than the type of variable. then,
~~It performs narrowing expli~~
the compiler cannot perform narrowing implicitly so it throws error

Relational operator:-

- Relational operators are binary operators
- It is used to perform comparion ~~two values~~ between two values or operands., for decision making
- Purpose.
- They are also called as comparison operators.

< → Lesser than
> → greater than
<= → Less than equal
=> → greater than equal
== → equality operator
!= → Not equals.

The result of relational operator is boolean type.

Eg:

```
class Relation2
{
    public static void main(String args[])
    {
        System.out.println(10>20);      //false
        System.out.println(10<20);      //true.
        System.out.println(0<0.0);      //false
        System.out.println(59.999==60); //false
        System.out.println(97!=100);    //true
        System.out.println('a'<'A');    //false
        System.out.println('z'<=122);   //true
        System.out.println(false<true); //CTE
        System.out.println(false==true); //false
        System.out.println(true!=true); //False.
        System.out.println(22f>=22.0/7.0); //False.
        System.out.println(22.0/7.0==22.0f/7.0f);
                                        //true.
    }
}
```

conditional operator's :→

conditional operator is a ternary operator which is used for decision making purpose.

Syntax:

```
condition ? valueIexp : valueIexp
   ↓          ↓          ↓
operand1   operand2   operand3.
```

Eg:

```
int num=-5;
String name=num>0?"positive":"Negative";
System.out.println(num+"is"+name);
```