Coder Army

Lecture - 004

By - Rohit Negi Bhaiya

Typed By – Nishit Mehta

Typecasting, Conditions and For Loop

Difference between Assignment Operator (=) and Comparison Operator (==):

Assignment Operator assigns a value to a variable. Ex: a **=** 10. Here 'a' is assigned with value '10'.

Comparison Operator compares the two values, if equal it returns '1' or 'true' else it returns '0' or 'false'.

Typecasting:

It refers to the process of converting a value of one data type to another data type.
Ex:
int a = 66;
char c = 'b';
a = c; Here we are updating value 'a' to the value of 'c'. As we know 'c' is of 'char' datatype, it's value cannot be stored inside 'a'. Therefore, 'a' stores the ASCII value of 'c' i.e., **98**. Hence, the updated value of 'a' becomes **98**.

c = a; Here we are updating value 'c' to the value of 'a'. As we know 'a' is of 'int' datatype, it's value cannot be stored inside 'c'. Therefore, 'c' stores that character whose ASCII value is 'a' i.e., **B**. Hence, the updated value of 'c' becomes **B**.

Typecasting is of two types:

1. **Implicit Typecasting:**
   Ex char ch = 97; cout << ch; ---→ **a**
2. **Explicit Typecasting:**
   Ex char ch = (char) 97; cout << ch; ---→ **a**

Data Lose Concept:

In Typecasting, we need to remember one thing that, there are chances of data lose to occur. While converting from bigger datatypes to smaller datatypes, there is chance that some data might get lost. But this will not occur while converting from smaller datatypes to bigger datatypes.

Ex char ch = 9999;
cout << ch;
9999 ----------→ 1001110000111
Binary Conversion
stores only last 8 bits

Therefore, 'ch' stores **00001111** in memory, resulting in data lose.

Conditions:

if:

if(*condition*) {
execute
}

if-else:

if(*condition*) {
execute 1
}
else {
execute 2
}

if-else if:

if(*condition 1*)
execute 1
else if (*condition 2*)
execute 2

if-else if-else:

if(*condition 1*)
execute 1
else if (*condition 2*)
execute 2
else if (....)

else
execute n

nested if:

if(*condition 1*)
{
if (*condition*)
{
}
else
{
}
}

For Loop:

initialization → condition → updation

```
for(int i = 0; i < 5; i = i + 1) {
        cout << "Love";  --------→ executing body
}
```

flow –