CONTENT 13

### Typecasting in C

#### Typecasting is a way to convert one data type into another one. It is also known as data conversion or type conversion.

**Code:**

#include<stdio.h>

int main(int argc, char const \*argv[])

{

    int a=5;

    float b=54;

    printf("The value of a is %f", b);

    return 0;

}

Output:

The value of a is 54.000000

But here If I want to convert this b in integer (want to see integer equivalent) then I want to do Typecasting.

#include<stdio.h>

// Syntax of Typecasting

// (type) value

int main(int argc, char const \*argv[])

{

    int a=5;

    float b=54;

    printf("The value of a is %f", (int) b);

    return 0;

}

Output:

The value of a is 54

We got 54 because here it 54 is our typecast and it get converted to int also.

Now here I am gonna to put 54/5 as value of b now let see what we get;

#include<stdio.h>

int main(int argc, char const \*argv[])

{

    int a=5;

    float b=54/5;

    printf("The value of a is %f", b);

    return 0;

}

Output:

The value of a is 10.000000

So we get 10.000000 as output but this is not correct (because as we calculate 54/5 we get 10.800 something as right answer).

We got 10.000000 as ans because 54 is integer and 5 is also an integer. So in c programming there is something called type conversion (conversion is already done by Compiler from one data type to another data type). If you perform any operation between int & int then it will be an integer so above float b contains both int values so it get evaluates into 10.000000.

If here I do this(typecasting);

#include<stdio.h>

int main(int argc, char const \*argv[])

{

    int a=5;

    float b=(float) 54/5;

    printf("The value of a is %f", b);

    return 0;

}

Then we get correct output 10.80000 something.