

▣ Type casting

Introduction.

1. Allows you to change the data type of a variable from one type to another.
2. Crucial when you need to perform operations involving variables of different data type, ensuring that the data is handled correctly.

char \rightarrow int ✓

int \rightarrow char ✓

float \rightarrow int ✓

int \rightarrow float ✓

Like what is the R/P of

char + int = ???

• Implicit Type casting.

1. Aka, Automatic Type Conversion.

2. compiler automatically converts one data type to another during an operation.

3. This happens when you perform operations involving variables of different data types, and the compiler promotes one type to a larger type to maintain precision.

Snippet

// int to float

```
int num1 = 10;
```

```
float num2 = 5.5;
```

```
float result = num1 + num2;
```

```
cout << result << endl;
```

int + float

↓ ↓
10 5.5
) It changes to float
 → 10.0

o/p

15.5

// char to int.

```
char ch = 'A';
```

```
char a = ch + 1; // implicit conversion from
```

```
// a = 66 // char to int.
```

```
cout << a << endl;
```

o/p

B

// int to char.

```
int a = 97;
```

```
char ch = a; // implicit conversion from char
```

```
cout << ch << endl; // to char.
```

o/p

a

• Explicit Type casting.

1. Aka, Manual type conversion
2. Allows you to explicitly specify the desired data type during an assignment or operation.
3. You use the casting operator, which is represented by parenthesis containing the target data type.

```
int num1 = 10;
float num2 = 5.5;
float result = num1 + (int) num2;
cout << result << endl;
```

Q.1 15

↓
changed the data type explicitly.

// double to int.

```
double pi = 3.14159265;
int intpi = (int) pi; // Explicit conversion from
cout << intpi << endl; // double to int.
```

Q.2 3.

// float to char.

```
float floatingno. = 65.5;
char charvalue = (char) floatingno;
cout << charvalue << endl;
```

o/p A.

// int to float

int a = 10;

int b = 3.0;

float c = a/(float)b;

// Explicit type conversion from int to float

cout << c << endl;

o/p 3.33333.

Bonus

① int/int = int

② int/float = float

③ float/int = float

► Switch case in C++.

- Alternative to long if-else statements.

Syntax

```
Switch (exp)
```

```
{
```

```
    Case 1:
```

```
        // code
```

```
        break;
```

```
    Case 2:
```

```
        // code
```

```
        break;
```

- Expression must have integral value.

eg:- ~~if~~ int, char, long, double.

It doesn't support floating numbers.

Snippet

```
main() {
```

```
    int op;
```

```
    cout << "What operation do you want to" << endl;
```

```
    cin >> op;
```

```
    int a, b;
```

```
    cin >> a >> b;
```

```
    switch (op) {
```

```
        Case 1: cout << a + b;
```

```
        break;
```

```
        Case 2: cout << a - b;
```

```
        break;
```

```
        Case 3: cout << a * b;
```

```
        break;
```

```
    }
    return 0;
```