2.1 TypeCasting

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 types, ensuring that the data is handled correctly.

char → int
int → char
int → float
float → int

Implicit Type Casting

- → Aka Automatic type conversion
- → Compiler automatically converts one data type to another during an operation
- → This happens when you perform operations involving variables of different data types & the compiler promotes one type to a larger type to maintain precision

```
int num1 = 10;
float num2 = 5.5;
cout << num1 + num2;
```

o/p = 15.5

```
char ch = 'A';
int num = 1;
cout << ch + num;
```

o/p = 66

Explicit Type Conversion

- → Aka manual conversion
- → Allow you to explicitly specify the desired data type during an assignment or operation
- → 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;
```

10 + 5

o/p = 15.0

```
double pi = 3.14;
int intPi = (int) pi;
cout << intPi;
```

```
float num = 65.35;
Char ch = (char) num;
cout << ch;
```

```
int a = 10;
int b = 3.0;
float c = a/b;
cout << c;
```

o/p = 3

```
float c = a/(float(b));
cout << =c;
```

o/p = 3.33

Bonus

1) int / int → int
2) float / int → float
3) int / float → float