TYPE CONVERSION

Content

- Type conversion
- Basic to Basic type
- Basic to Class type
- Class to Basic type
- Class to Class type

Let's understand

- **Have you bought any Pulse or Rice?**
- **It may possible that you will find some pebbles inside it?**
- **That pebbles are also counted in weight.**
- **When someone is giving speech in English if he/she may use some words of other language?**
- **The other language words used are also considered in English speech.**

Type Conversion

- Similarly, when constants and variables of different types are mixed up in an expression

```
float x=6.2;

int n= x*2+6-1.1;
```

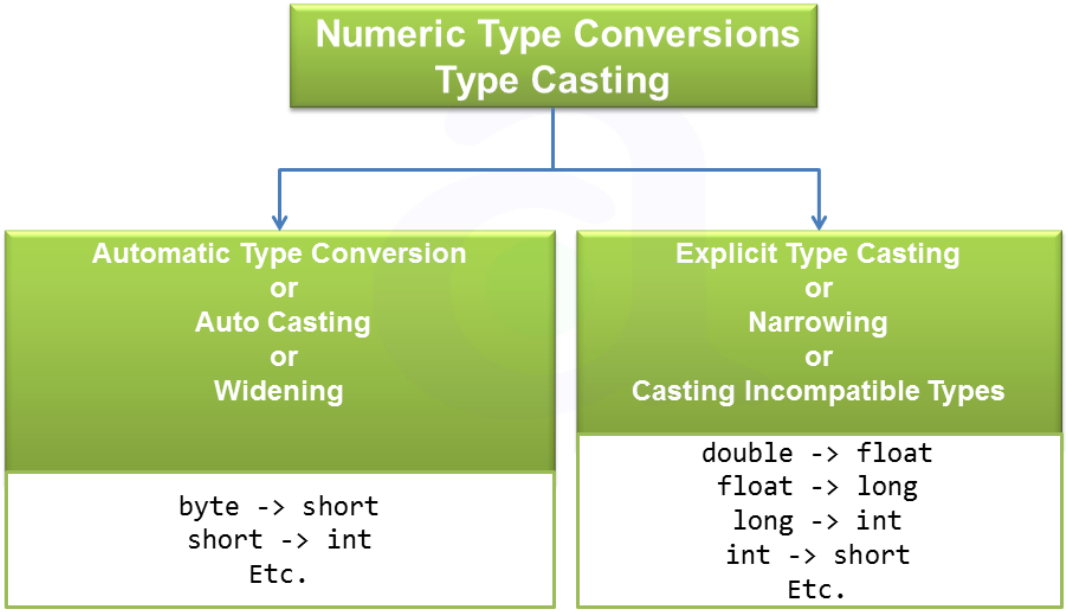
Output

17

- Then they are converted into the same type.
- This is called type conversion

Basic to Basic type conversion

- When a primitive data type is converted in to another primitive type it is known as basic to basic type conversion.
- It can be further divided into two categories:



- **Implicit Type Conversion Rule:**
  - If two variables are there, 1 will small size and other with large size; the result of the expression will be in large size.
  - E.g.
    - • cout<<(1.3*2);
    - Result:
      - 2.6 not=> 2

- **Explicit Type Conversion:**
  - Forces an expression to be of specific type.
  - It is also called type casting.
  - Syntax of type casting/explicit conversion:
    - type(expression);
  - E.g.
    - • int(4*6.22);
  - Result:
    - • 24 not=> 24.88

- **Explicit Type Conversion:**
  - There is a possibility of loss of data in case of type casting.
  - E.g.
    - • cout<<int(4*6.22);
  - Result:
    - • 24 not=> 24.88
      - There is a loss of .88.
  - So type casting should be done carefully.

Basic to Class type conversion

- In this type of conversion the source type is basic type and the destination type is class type.
- For example we have class *employee* and one object of employee *‘emp’* and suppose we want to assign the employee code of employee *‘emp’* by any integer variable say *‘Ecode’* then the statement below is the example of the conversion from basic to class type.
- emp = Ecode ;

Class to Basic type

- In this type of conversion the source type is class type and the destination type is basic type.
- For example we have class *employee* and one object of employee *‘emp’* and suppose we want to assign the employee code of employee object *‘emp’* to any integer variable say *‘Ecode’* then the statement below is the example of the conversion from class to basic type.
- Ecode = emp ;

Example

```
#include <iostream>

using namespace std;
class classtobasic
{
    int marks;
public:
    classtobasic(int x)
    {
        marks=x;
    }
    operator int()//casting operator or conversion function
    {

        return(marks+100);
    }

};
```

```
int main()
{
    int marks1;
    marks1=233;
    classtobasic object(marks1);
    marks1=object; //calling conversion method
    cout<<"Marks are"<<marks1;
    return 0;
}
```

Class to Class type

- In this type of conversion both the type that is source type and the destination type are of class type.
- In other words, one class data type is converted into the another class type.
- For example we have two classes one for "*computer*" and another for "*mobile".* Suppose if we wish to assign *"price"* of *computer* to *mobile* then it can be achieved by the statement below which is the example of the conversion from one class to another class type.
- mob = comp ;