How to convert a class to another class type in C++?

Class conversion can be done with the help of operator overloading. This allows data of one class type to be assigned to the object of another class type.

```
#include <bits/stdc++.h>
using namespace std;
//type to which it will be converted
class Class_type_one {
   string a = "TutorialsPoint";
   public:
      string get_string(){
         return (a);
   }
   void display(){
      cout << a << endl;
   }
};
//class to be converted
class Class_type_two {
   string b;
   public:
   void operator=(Class_type_one a){
      b = a.get_string();
   }
   void display(){
      cout << b << endl;
   }
};
int main(){
   //type one
   Class_type_one a;
   //type two
   Class_type_two b;
   //type conversion
   b = a;
   a.display();
   b.display();
   return 0;
}
```

Output

TutorialsPoint
TutorialsPoint

Class Type to Basic Type:

In this type of conversion the source type is class type and the destination type is basic type. Means class data type is converted into the basic type.

- For example we have class *Time* and one object of *Time* class *'t'* and suppose we want to assign the total time of object *'t'* to any integer variable say *'duration'* then the statement below is the example of the conversion from class to basic type.

duration= t ; // where, t is object and duration is of basic data type

- Here the assignment will be done by converting *"t"* object which is of class type into the basic or primary data type. It requires special casting operator function for class type to basic type conversion. This is known as the *conversion function*. The syntax for the conversion function is as under:

```
operator typename( )
{
    ….
    ….
    ….
}
```

- For example suppose we want to assign time in *hours* and *minutes* in the form of total time in minutes into one integer variable *"duration"* then we can write the type conversion function as under:

/* Program to demonstrate Class type to Basic type conversion. */

```
#include "iostream.h"
#include "conio.h"
#include "iomanip.h"

class Time
{
        int hrs,min;
        public:
```

```
                Time(int ,int);   // constructor
                operator int();   // casting operator function
                ~Time()          // destructor
                {
                        cout<<"Destructor called..."<<endl;
                }
};

Time::Time(int a,int b)
{
        cout<<"Constructor called with two parameters..."<<endl;
        hrs=a;
        min=b;
}

Time :: operator int()
{
        cout<<"Class Type to Basic Type Conversion..."<<endl;
        return(hrs*60+min);
}

void main()
{
        clrscr();
        int h,m,duration;
        cout<<"Enter Hours ";
      cin>>h;
        cout<<"Enter Minutes ";
      cin>>m;
        Time t(h,m);       // construct object
        duration = t;      // casting conversion OR duration = (int)t
        cout<<"Total Minutes are "<<duration;
        cout<<"2nd method operator overloading "<<endl;
        duration = t.operator int();
        cout<<"Total Minutes are "<<duration;

      getch();
}
```

- Notice the statement in above program where conversion took place.

  duration = t;

- We can also specify the casting type and write the same statement by the following way to achieve the same result.

duration = (int) t; // Casting

- The conversion function should satisfy the following condition:

1. ***It must be a class member.***
2. ***It must not specify the return value even though it returns the value.***
3. ***It must not have any argument.***

Conversion to Basic Type to Class Type:

- In this type of conversion the source type is basic type and the destination type is class type. Means basic data type is converted into the 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 ;

- Here the assignment will be done by converting *"Ecode"* which is of basic or primary data type into the class type.
- The conversion from basic type to the class type can be performed by two ways:

1. ***Using constructor***
2. ***Using Operator Overloading***

Using Constructor

- We can use constructor to perform type conversion during the object creation.
- Consider the following example with class 'Time' in which we want to assign total time in minutes by integer variable *'duration'.*
- To achieve that we have implemented one constructor function which accepts one argument of type integer as follow:

/* Program to convert basic type to class type using constructor */

```
#include "iostream.h"

#include "conio.h"

class Time

{
```

```
        int hrs,min;
        public:
            Time(int);
                void display();
};

Time :: Time(int t)
{
        cout<<"Basic Type to ==> Class Type Conversion..."<<endl;
        hrs=t/60;
        min=t%60;
}

void Time::display()
{
        cout<<hrs<< ": Hours(s)" <<endl;
        cout<<min<< " Minutes" <<endl;
}

void main()
{
        clrscr();

        int duration;
```

```
        cout<<"Enter time duration in minutes";
    cin>>duration;

        Time t1=duration;

        t1.display();
        getch();
}
```

- Here, we have created an object *"t1"* of class *"Time"* and during the creation we have assigned integer variable *"duration"*. It will pass time duration to the constructor function and assign to the *"hrs"* and *"min"* members of the class *"Time"*.
- We have to note that during type conversion using the constructor we can pass only one argument and we can do type conversion at the type of initialization only.

Using Operator Overloading

- We can also achieve type conversion by operator overloading.
- We can overload assignment operator for this purpose.
- Above example of *Time* class can be rewritten for type conversion using operator overloading concept to overload the assignment operator (=) as follow:

```
/* Program to convert from basic type to class type using operator overloading */

#include "iostream.h"
#include "conio.h"
class Time
{
        int hrs,min;
        public:
```

```
                    void display();
                    void operator=(int); // overloading function
};
void Time::display()
{
        cout<<hrs<< ": Hour(s) "<<endl ;
        cout<<min<<": Minutes"<<endl ;
}

void Time::operator=(int t)
{
        cout<<"Basic Type to ==> Class Type Conversion..."<<endl;
        hrs=t/60;
        min=t%60;
}

void main()
{
        clrscr();
        Time t1;
        int duration;
        cout<<"Enter time duration in minutes";
      cin>>duration;
        cout<<"object t1 overloaded assignment..."<<endl;
```

```
    t1=duration;

    t1.display();

    cout<<"object t1 assignment operator 2nd method..."<<endl;

    t1.operator=(duration);

    t1.display();

    getch();

}
```

- By using overloaded assignment operator we can perform the type conversion at any place in program.