# RELATIONAL OPERATORS OVERLOADING IN C++

There are various relational operators supported by C++ language like $<$, $>$, $<=$, $>=$, $==$, *etc.* which can be used to compare C++ built-in data types.

You can overload any of these operators, which can be used to compare the objects of a class.

Following example explains how a < operator can be overloaded and similar way you can overload other relational operators.

```
#include <iostream>
using namespace std;

class Distance
{
   private:
      int feet;             // 0 to infinite
      int inches;           // 0 to 12
   public:
      // required constructors
      Distance(){
         feet = 0;
         inches = 0;
      }
      Distance(int f, int i){
         feet = f;
         inches = i;
      }
      // method to display distance
      void displayDistance()
      {
         cout << "F: " << feet << " I:" << inches <<endl;
      }
      // overloaded minus (-) operator
      Distance operator- ()
      {
         feet = -feet;
         inches = -inches;
         return Distance(feet, inches);
      }
      // overloaded < operator
      bool operator <(const Distance& d)
      {
         if(feet < d.feet)
         {
            return true;
         }
         if(feet == d.feet && inches < d.inches)
         {
            return true;
         }
         return false;
      }
};
int main()
{
   Distance D1(11, 10), D2(5, 11);

   if( D1 < D2 )
   {
      cout << "D1 is less than D2 " << endl;
   }
   else
   {
      cout << "D2 is less than D1 " << endl;
```

```
    }
    return 0;
}
```

When the above code is compiled and executed, it produces the following result:

```
D2 is less than D1
```