C++ Class Implementation and Operator Overloading

**Implement a `Book` class in C++ with the following specifications:**

**1. Data Members:**

- title (a string representing the title of the book)

- author (a string representing the author of the book)

- year (an integer representing the year the book was published)

**2. Constructors:**

A constructor that takes three parameters (`const std::string& t`, `const std::string& a`, `int y`) to initialize the `title`, `author`, and `year`.

**3. Member Functions:**

std::string toString() const`: Returns a string representation of the book in the format "Title by Author (Year)".  
  
**Hint (Will be used in the << overloading operator)**

**4. Operator Overloads:**

Overload the == operator to compare two Book objects. Two books are considered equal if they have the same title, author, and year.

Overload the << operator to print a Book object using the toString() method.

**Example Usage:  
Write a `main` function to demonstrate the functionality of your `Book` class, including the comparison and printing of `Book` objects.**

**int main() {**

Book book1("1984", "George Orwell", 1949);

Book book2("Animal Farm", "George Orwell", 1945);

Book book3("1984", "George Orwell", 1949);

// Comparison using overloaded ==

if (book1 == book3) {

std::cout << "Books are equal" << std::endl;

} else {

std::cout << "Books are not equal" << std::endl;

}

// Printing using overloaded <<

std::cout << book1 << std::endl;

return 0;

**}**

**Expected Output:**

Books are equal  
1984 by George Orwell (1949)

**Hints:**

To overload the `==` operator, you should define it as a member function or a friend function.  
To overload the `<<` operator, you should define it as a friend function.