

Equality and Inequality Operators Solutions

- What is the prototype of the equality operator?

`bool operator ==(const T& rhs); // Equality operator for type T`

- Whenever we write a statement such as

`a == b;`

- The compiler will generate code which calls the operator with the appropriate argument
- The operator is a member function, so it will be called as

`a.operator==(b);`

- What is the prototype of the inequality operator?

`bool operator !=(const T& rhs); // Inequality operator for type T`

- Whenever we write a statement such as

`a != b;`

- The compiler will generate code which calls the operator with the appropriate argument
- The operator is a member function, so it will be called as
`a.operator!=(b);`

- Why are these operators implemented as member functions?
 - Because they are unary operators
- Why do these operators take their argument by reference to const?
 - The most efficient way to inspect data
 - Avoids copying large objects
 - Avoids accidentally modifying the data

- Explain what this code does

```
return !(*this == other);
```

- *this gives the current object value
- *this == other compares the current object to “other”
- ! inverts the result of the equality operator
- The returned value will be true if the objects are unequal, false if they are equal