

# Less-than Operator Solutions

# Less-than Operator Summary

- What is the prototype of the less-than operator?

`bool operator <(const T& rhs); // Less-than operator for type T`

- How is it invoked?

- 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)`

# Standard Library and < Operator

- Why is the less-than operator important?
  - The Standard Library uses the less-than operator for sorting and ordering operations
  - All other comparison operators can be derived from less-than

# Constraints on Less-than Operator

- What factors should be considered when implementing a less-than operator?
  - The results of applying the operator should be consistent with the equality and inequality operators
  - If  $a < b$  is true, then  $a == b$  is false and  $a != b$  is true
  - If  $a == b$  is true, then  $a < b$  is false and  $b < a$  is false
  - Think carefully about what it means to compare two objects (e.g. different people can have the same name)

# Adding a Less-than Operator

- Implement a simple class with a less-than operator
- Write a program to test your class

# Sorting a vector

- Write down an expression that will sort all the elements of a vector "names" in ascending order

```
sort(names.begin(), names.end());    // Sort all the elements in the vector "names"
```

- Which include file is needed for this?

```
<algorithm>
```

- How is the resulting order of the elements determined?
  - The < operator of the element is used

# Sorting a vector

- Write a program which creates a vector of strings, sorts it and prints out the resulting vector
- What result do you expect?
  - The strings are printed in alphabetical order
- Write a similar program, but this time use the class you wrote in the previous exercise as the element type