Thread-local Variables Solutions

Thread-local variables

- How do thread-local variables differ from static variables?
  - A static variable is a single object
  - This object is shared by all the threads which can see its declaration
  - The object is not destroyed until the end of the program
  - With thread-local variables, each thread has its own object
  - This object cannot be accessed by other threads
  - It is destroyed when the thread completes its execution

Thread-local variables

- Where can we declare thread-local variables?
  - Globally or at namespace scope
  - As data members of a class
  - As local variables in a function or within a scope
- When are thread-local variables initialized?
  - For globally, or at namespace scope, or as class member
    - At or before first use
  - For local variables in a function or within a scope
    - When the thread first passes through the declaration (same as for static local variables)

Thread-local Variable Example

- Write a program which uses thread-local variables
- Verify that each thread uses a different object