CS 180 Problem Solving and OO Programming

Fall 2011

Recitation Week 13: November 14-18, 2011
Threads and GUI

In this exercise you will write a program that consists of two classes, namely, `ConcExercise` and `MyThread`. Here is what these classes do.

`ConcExercise`: This class has a `main()` method which creates a `ConcExercise` object named `ce`. The constructor in `ConcExercise` creates a simple GUI that contains one button labeled START and an empty text field textfield.

`ConcExercise` also contains a method named `update()` that when called with an integer parameter writes the integer into the textfield (after converting the integer to a string).

The `actionPerformed()` method in `ConcExercise` is called when the `START` button is clicked by the user. When called, this method sets a boolean flag named `start` to true.

After creating the `ce` object the `main()` method waits until `start` is true. It then creates an object named `mt` of type `MyThread` and starts the thread. It then waits for `mt` to complete execution. Note that the while loop in `main()` that waits for the user to press `Start` must have a sleep command inside to sleep for a small amount of time, say 10 milliseconds (guess why). The `main()` terminates when `mt` joins. mt() is created as follows:

`MyThread` mt=new `MyThread` (ce, n);

where `ce` is the object mentioned above and n> 0 is an integer. =

`MyThread`: This class extends the `Thread` class. Its constructor copies the input parameters into local variables. Upon the thread start command the `run()` method begins execution and repeats the following steps `n` times.

(a) Display a random integer in the range 0-100 (exclusive) in the textfield.
(b) Sleep for 1 second.

Note that `mt` can call update() using the `ce` object.

<End of Problems for Week 13>