

Programming Languages G22.2110 Summer 2007 hw10

Assigned Th 7/26/2007, no due date.

Do not Submit this Homework Assignment

This assignment is optional. It will not be graded, and it earns you no points towards the course grade.

1. Concurrency (0 points)

Consider the following (buggy) Java shared list class:

```
class SharedList {
    private static class Cell {
        int data = 0;
        Cell next = null;
    }
    private Cell head = null;
    public void put(int newData) {
        Cell oldHead = head;
        head = new Cell();
        head.data = newData;
        head.next = oldHead;
    }
    public int get() {
        while (null == head) { }
        int result = head.data;
        head = head.next;
        return result;
    }
}
```

- 1a. Assume that the list starts out empty, and a producer thread *p* calls `put(123)`, getting as far as shown in the following table before a consumer thread *c* gets scheduled instead.

| head | head.data | head.next | oldHead | result | thread: code |
|------|-----------|-----------|---------|--------|------------------------|
| null | (n/a) | (n/a) | (n/a) | (n/a) | p: Cell oldHead = head |
| | | | null | | p: head = new Cell() |
| new | 0 | null | | | p: head.data = newData |

Show an interleaving that eventually causes a `NullPointerException`.

- 1b. Fix the code to prevent the race condition.
1b. Fix the code to prevent deadlock.

2. How to learn a language (0 points)

Reflect on your experience in learning how to learn a language. As a reminder, here are the steps you were asked to repeat for each of the languages you learned this semester:

- (i) Find peers and gurus.
- (ii) Get compiler, and figure out how to use it.
- (iii) Read tutorial.
- (iv) Bookmark language and library reference.
- (v) Read example code.
- (vi) Write little programs exercising I/O, types, control flow, and library usage.
- (vii) Learn how to understand common error messages.
- (viii) Write slightly larger programs than in Step (vi).

2a. Which step helped you the most, and why?

2b. Which step took the most time? How long?

2c. Think of another helpful finger exercise to add to Step (vi).

2d. What were the exercises for Step (viii) that you did as part of homeworks for this class?

3. SML (0 points)

3a. Write a function `listMinInt` that finds the minimum element of an integer list. Your function should have the type `int list -> int`. Your function does not need to yield a sensible result for empty lists.

3b. Write a function `removeFirst` that takes a pair of an item and a list, and returns the list with the first occurrence of the item removed, if any. Your function should have the type `'a * 'a list -> 'a list`.

3c. Write a function `listSortInt` that sorts an integer list in non-descending order. You can reuse the functions from Questions 3a and 3b for this step. Your function should have the type `int list -> int list`.

4. SML (0 points)

4a. Write a function `listMin` that finds the minimum element of a list, according to a comparison function it receives as a parameter. Your function should be curried: given the comparison function, it should return another function that then takes a list parameter and returns the minimum element. For example, `listMin (op <)` should return a function that works the same as `listMinInt` from Question 3a. The type of your function should be `('a * 'a -> bool) -> 'a list -> 'a`.

4b. Write a function `listSort` that sorts a list in the order specified by a comparison function it receives as a parameter. Your function should be curried: given the comparison function, it should return another function that then takes a list parameter and returns the sorted list. For example, `listSort (op <);` should return a function that works the same as `listSortInt` from Question 3c. The type of your function `listSort` should be `('a * 'a -> bool) -> 'a list -> 'a list`.

<http://www.cs.nyu.edu/courses/summer07/G22.2110-001/hw10.pdf>

Total points: 0.