CIS 120 Quiz 7A

March 4–5, 2015

Name: __________________________________________________________________________

PennKey (e.g. mkizner): ____________________________________________________________ Section: _______________________

Indicate the section you’re registered for, even if you’re attending a different section.

type 'a ref = { mutable contents: 'a }

type donut_box = {
    (* returns the number of donuts in the box *)
    get_count: unit -> int;

    (* consumes all donuts in the box *)
    eat: unit -> unit;
}
Consider the following code segment, referring to the comments in the donut_box definition on the opposite side of the page. Notice that the TAs can steal donuts from the student box, without calling eat, by modifying the student state.

```ocaml
let ta_state = { contents = 0 } in
let student_state = { contents = 0 } in
let ta_box = buy_open_box ta_state in
let student_box = buy_open_box student_state in
let ta_count = ta_box.get_count () in
let student_count = student_box.get_count () in
```

1. What are the values of `ta_count` and `student_count`?

2. Complete the definition of the `buy_open_box` function used in this code segment.

```ocaml
let buy_open_box (state: int ref) : donut_box =
  state.contents <- 25;
  { get_count =
    eat =
  }
```

3. Now write the `buy_sealed_box` function, which should prevent any pesky thieves from eating donuts without calling `eat`. (How might you go about doing this?) The `donut_box` should be initialized with 25 donuts, as before.

```ocaml
let buy_sealed_box () : donut_box =
```