CS 551 Spring 2025 Quiz 2 KEY

February 19, 2025

Name:

Email:

Instructions: Put your name and email in the appropriate places. Answer, to the best of your ability the question(s) below. Additionally, sign your name below the academic honesty statement. If you deviate from these instructions in any way, you will receive a zero on the quiz.

Unless otherwise specified, you can assume that all necessary inputs have been made and there are no deliberate typos in function or type names.

Additional information: An `Option<T>` is an enum that has two variants: `Some(T)` and None, where `T` is a type known at compile time.

Consider the following stuct definitions and implementations:

```
1  struct Node {
2      val: i32,
3      left: Option<Box<Node>>,
4      right: Option<Box<Node>>,
5  }
6
7  struct Tree {
8      root: Option<Box<Node>>,
9  }
10
11 impl Tree {
12     fn insert(&mut self, val: i32) {
13         // assume this code properly inserts the value into the `Tree`
14     }
15 }
```

Question A:

Will the following program compile:

```
1  impl Tree {
2      fn new() -> Self {
3          Self {
4           root: None,
5          }
6      }
7
8      fn is_empty(self) -> bool {
9          // `is_none()` returns true if the Option is None and false otherwise
10         self.root.is_none()
11     }
12 }
13
14 fn main() {
15     let mut tree = Tree::new();
16
17     println!("is tree empty? {}", tree.is_empty());
18
19     tree.insert(1);
20 }
```

(circle your answer)

No

Reason: `is_empty(self)` (line 8) takes *ownership* of `self` and never gives it back. When `tree.is_empty()` is called on line 17, ownership of `tree` is given to `is_empty(self)`. Line 19 will thus be a use after move error.

Question B:

Will the following program compile:

```
1  impl Tree {
2      fn new() -> Self {
3          Self {
4           root: None,
5          }
6      }
7
8      fn is_empty(&self) -> bool {
9          // `is_none()` returns true if the Option is None and false otherwise
10         self.root.is_none()
11     }
12 }
13
14 fn main() {
15     let mut tree = Tree::new();
16
17     println!("is tree empty? {}", tree.is_empty());
18
19     tree.insert(1);
20 }
```

(circle your answer)

Yes

Reason: `is_empty(&self)` only *borrows* self, so ownership is automatically returned to `main()` after line 17.

Question C:

Will the following program compile:

```
1  impl Tree {
2      fn new(&self) -> Self {
3          Self {
4           root: None,
5          }
6      }
7
8      fn is_empty(&self) -> bool {
9          // `is_none()` returns true if the Option is None and false otherwise
10         self.root.is_none()
11     }
12 }
13
14 fn main() {
15     let mut tree = Tree::new();
16
17     println!("is tree empty? {}", tree.is_empty());
18
19     tree.insert(1);
20 }
```

No

Reason: `new(&self)` (line 2) borrows `self`, but it is called as an associated function of the `Tree` type (line 15).

Academic honesty statement: I have done this quiz completely on my own. I have not copied it from, nor have I given answers to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of 0 for the quiz involved, my grade in the class will be reduced by at least one level (e.g., from A to B) for my offense, and that I will receive a grade of "F" for the course for any additional offense of any kind.