

CS 551 Spring 2025 Quiz 10 KEY

April 17, 2025

Name:

Email:

Instructions: Put your name and email in the appropriate places. Answer, to the best of your ability the **FIVE QUESTIONS** 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 imports have been made and there are no deliberate typos in function or type names.

Consider the following structs, traits, enums, and functions:

```
1  #[derive(Debug, Clone)]
2  struct MyStruct {
3      string: String,
4  }
5
6  impl MyStruct {
7      fn new(string: String) -> Self {
8          Self { string }
9      }
10 }
11
12 impl Deref for MyStruct {
13     type Target = String;
14
15     /// You can call 'deref()' on a struct or it will be implicitly called
16     /// when you get a reference to a struct. E.g., when you borrow it.
17     ///
18     /// This is how you can pass a '&String' to a function that takes an '&str'.
19     /// It's because 'String' has a 'Deref' implementation that 'Target's 'str'.
20     fn deref(&self) -> &Self::Target {
21         &self.string
22     }
23 }
24
25 impl AsRef<String> for MyStruct {
```

```

26     fn as_ref(&self) -> &String {
27         &self.string
28     }
29 }
30
31 fn function1(my_string: &str) {}
32
33 fn function2(my_string: &String) {}
34
35 fn function3(my_struct: MyStruct) {}
36
37 fn function4(my_struct: &MyStruct) {}
38
39 fn main() {}

```

Question 1: Will the following program compile:

```

1 fn question1() {
2     let my_string = String::from("question1");
3     let my_struct = MyStruct::new(my_string);
4
5     function1(&my_struct.string);
6
7     println!("{my_struct:?}");
8 }

```

(circle your answer)

Yes

Reason: `function1()` only *borrow*s `my_struct.string`.

Question 2: Will the following program compile:

```

1 fn question2() {
2     let my_string = String::from("question2");
3     let my_struct = MyStruct::new(my_string);
4
5     function1(function1(my_struct.as_ref()));
6
7     println!("{my_struct:?}");
8 }

```

(circle your answer)

Yes

Reason: The call to `as_ref()` uses the `AsRef<String>` implementation of `MyStruct` to turn a `&my_struct` into a `&String`, which is then `Deref`'d into a `&str` implicitly. This is functionally equivalent to question 1, except there is an extra level of indirection (although the compiler should optimize that out in this case).

Question 3: Will the following program compile:

```

1 fn question3() {
2     let my_string = String::from("question3");
3     let my_struct = MyStruct::new(my_string);
4
5     function2(&my_struct);
6
7     println!("{my_struct:?}");
8 }

```

(circle your answer)

Yes

Reason: The borrow in `function2(&my_struct)` uses the `Deref` implementation of `MyStruct` to turn a `&my_struct` into a `&String`.

Question 4: Will the following program compile:

```

1 fn question4() {
2     let my_string = String::from("question4");
3     let my_struct = MyStruct::new(my_string);
4
5     function3(my_struct);
6
7     println!("{my_struct:?}");
8 }

```

(circle your answer)

No

Reason: `function3()` takes ownership of `my_struct`.

Question 5: Will the following program compile:

```

1 fn question5() {
2     let my_string = String::from("question5");
3     let my_struct = MyStruct::new(my_string);
4
5     function4(&my_struct);
6
7     println!("{my_struct:?}");
8 }

```

(circle your answer)

Yes

Reason: `function4()` only *borrow*s `my_struct`.

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.