CS 551 Spring 2025 Quiz 7 KEY

April 9, 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 imports have been made and there are no deliberate typos in function or type names.

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

```
#[derive(Debug, Clone)]
struct MyStruct {
    num: i32,
    string: String,
}

fn function1(my_struct: MyStruct) { }

fn function2(my_struct: &MyStruct) { }

fn function3(my_string: &str) { }

fn function4(my_string: String) { }

fn function5(num: i32) { }

fn main() { }
```

Question 1: Will the following program compile:

```
fn question1() {
    function1(MyStruct {
        num: 1,
        string: "question1".to_string(),
    });
    println!("{my_struct:?}");
}
```

(circle your answer)

No

Reason: the variable `my_struct` never exists.

Question 2: Will the following program compile:

```
fn question2() {
    let num = 2;
    let string = "question2".to_string();
    let my_struct = MyStruct { num, string };
    function2(&my_struct);
    println!("{my_struct:?}");
}
```

(circle your answer)

Yes

Reason: `function2()` only *borrows* `my_struct`.

Question 3: Will the following program compile:

```
fn question3() {
    let num = 3;
    let string = "question3".to_string();
    let my_struct = MyStruct { num, string };
    function3(&my_struct.string);
    println!("{my_struct:?}");
}
```

(circle your answer)

Yes

Reason: `question3()` takes a `&str`. You can *borrow* a `String` to get a `&str`.

Question 4: Will the following program compile:

```
fn question4() {
    let num = 4;
    let string = "question4".to_string();
    let my_struct = MyStruct { num, string };
    function4(my_struct.string);
    println!("{my_struct:?}");
}
```

(circle your answer)

No

Reason: `function4()` takes ownership of it's `String` parameter. I.e., `my_struct`'s `string` field is *moved* into `function()`. When we then try to print out `my_struct` (which only needs to borrow `my_struct`) we get a borrow of a partially moved value error.

Question 5: Will the following program compile:

```rust
fn question5() {
    let num = 5;
    let string = "question5".to_string();
    let my_struct = MyStruct { num, string };
    function5(my_struct.num);
    println!("{my_struct:?}");
}
```

(circle your answer)

Yes

Reason: `function5()` takes an `i32`, which is a primitive. All primitives implement the `Copy` trait, which means their values are *copied* and the copy is *moved*. `Copy` is for types that can be handled via a simple bit copy. See `https://doc.rust-lang.org/std/marker/trait.Copy.html`.

Question 6: Will the following program compile:

```rust
fn question6() {
    let num = 6;
    let string = "question6".to_string();
    let my_struct = MyStruct { num, string };
    function1(my_struct.clone());
    println!("{my_struct:?}");
}
```

(circle your answer)

Yes

Reason: `function1()` takes ownership of its parameter, but instead of moving `my_struct` we are `clone()`ing it and moving that clone into `function1()`.

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.