CS 551 Spring 2025 Quiz 5 KEY

February 27, 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:

```rust
fn main() {}

fn func1(s: &str) {}

fn func2(s: String) {}
```

Question 1: Will the following program compile:

```rust
fn question1() {
    let s = "Question 1";
    for _ in 0..10 {
        func1(s);
        println!("{s}");
    }
}
```

(circle your answer)

Yes

Reason: `func1()` takes a string slice, which is a *borrow*. After `func1()` finishes running, all *borrowed* values are returned to the previous owner.

Question 2: Will the following program compile:

```
fn question2() {
    let s = "Question 2".to_string();
    for _ in 0..10 {
        func1(s);
    }
}
```

(circle your answer)

No

Reason: `func1()` takes a `&str`. `s` is a `String`, while we can easily *borrow* a `&str`, rust requires us to explicitly perform this borrow.

Question 3: Will the following program compile:

```
fn question3() {
    let s = "Question 3".to_string();
    for _ in 0..10 {
        func1(&s)
    }
}
```

(circle your answer)

Yes

Reason: `func1()` takes a `&str`. `s` is a `String`, but we can get a `&str` from a `String` by *borrowing* it. Note that a `&str` is different than a `&String`.

Question 4: Will the following program compile:

```
fn question4() {
    let s = "Question 4".to_string();
    for _ in 0..10 {
        func2(s)
    }
}
```

(circle your answer)

No

Reason: `func2()` *takes ownership* of its `String` parameter. The first iteration through the loop, `func2` *takes ownership* of `s`.

Question 5: Will the following program compile:

```
fn question5() {
    let s = "Question 5".to_string();
    for _ in 0..10 {
```

```
4           let s = s.clone();
5           func2(s);
6       }
7   }
```

(circle your answer)

Yes

Reason: Although `func2()` *takes ownership* of its `String` parameter, we are creating a new `String` on line 4 via the `clone()` method. Note that we are using variable shadowing so the `s` we create on line 4 is a different `s` than the one we create on line 2. The body of the for loop could have just been `func2(s.clone())`.

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.