



CMSC330 Spring 2023 Quiz 5

This entire quiz pertains to the Rust Programming Language

Problem 1: Basics

[4 pts]

Please circle **True** or **False** for the following statements:

Every value in rust has an owner

True False

With the way ownership in rust works, all values belong to a variable.

The lifetime of a value is always the same as the scope of the variable it was defined with

True False

Variables can be reassigned to other values, changing the lifetime of the value.

Rust is a memory-safe language

True False

Rust's garbage collector makes it memory safe.

There can still be Memory Leaks in Rust

True False

Memory leaks occur when users allocate space and then never use it; since this is a user issue it can still happen in Rust.

Problem 2: Lifetimes and Borrowing

[16 pts]

Version 1:

```
1 fn main(){
2     let f;
3     {
4         let x = String::from("mut");
5         f = &x;
6     }
7
8     let c;
9     {
10        let y = String::from("hey");
11        let z = "bye";
12        let d = z;
13        c = greethuh(y.as_str(),d);
14    }
15
16    {
17        let mut s = String::from("a");
18        let s1 = &s;
19        func(&mut s);
20        println!("{}",s);
21    }
22 }
23
24 fn greethuh<'a>(a:&'a str, b:&'a str) -> &'a str{
25     if a == "hello" {a} else {b}
26 }
27
28 fn func(s: &mut String){
29     s.push_str("y");
30     println!("{}",s);
31 }
```

Ownership

If there is no owner, write "NONE".

Who is the owner of "mut" immediately after line 5 is run? ☒ x

x is still the owner for "mut" because f is just borrowing an immutable reference to x

Who is the owner of "hey" immediately after line 24 is run? ☒ y

y is still the owner after line 24 is run because we pass which extracts a string slice. We are therefore not transferring ownership

Who is the owner of "bye" immediately after line 12 is run? ☐

This question was dropped

Who is the owner of "ay" immediately after line 29 is run? ☒ s

s is the owner of "ay" as we pass a mutable reference for it to func. So, the owner of "ay" is still s in main.

Who is the owner of "a" immediately after line 18 is run? ☒ s

s is the owner of "a" because s1 is borrowing an immutable reference to s.

Lifetimes

At what line is the value of "mut" dropped?

The last time "mut" is used is in line 5. So it is dropped at line 6

At what line is the value of "hey" dropped?

The value "hey" is dropped at line 14 since the last place "hey" is used is line 13.

At what line is the value of "bye" dropped?

"bye" is last used on line 13, therefore is dropped on line 14.



Version 2:

```
1 fn main(){
2     let f;
3     {
4         let x = String::from("mut");
5         f = &x;
6     }
7
8     let c;
9     {
10        let y = String::from("hey");
11        let z = "bye";
12        let d = z;
13        c = greethuh(y.as_str(),d);
14    }
15
16    {
17        let mut s = String::from("a");
18        let s1 = &mut s;
19        func(&mut s);
20        println!("{}",s);
21    }
22 }
23
24 fn greethuh<'a>(a:&'a str, b:&'a str) -> &'a str{
25     if a == "hello" {a} else {b}
26 }
27
28 fn func(s: &mut String){
29     s.push_str("y");
30     println!("{}", s,s);
31 }
```

Ownership

If there is no owner, write "NONE".

Who is the owner of "a" immediately after line 18 is run?

s

s1 is a mutable borrow of s, so s is still the owner.

Who is the owner of "mut" immediately after line 5 is run?

x

f is a borrow of x, so x is still the owner.

Who is the owner of "hey" immediately after line 24 is run?

y

.as_str() method is used, which means that the owner is still y.

Who is the owner of "bye" immediately after line 12 is run?

This question was dropped.

Who is the owner of "a" immediately after line 29 is run?

s

The s in func is a mutable borrow of s in main, so the owner is s in main.

Lifetimes

At what line is the value of "hey" dropped?

14 or 15

The last time "hey" is used is in line 13, so at line 14 it is dropped.

At what line is the value of "bye" dropped?

14 or 15

The last time "bye" is used is in line 13, so at line 14 it is dropped.

At what line is the value of "mut" dropped?

6 or 7

The last time "mut" is used is in line 5, so at line 6 it is dropped.



Version 3:

```
1 fn main(){
2     let f;
3     {
4         let x = String::from("mut");
5         f = x;
6     }
7
8     let c;
9     {
10        let y = String::from("hey");
11        let z = "bye";
12        let d = y;
13        c = greethuh(z,d.as_str());
14    }
15
16    {
17        let mut s = String::from("a");
18        let s1 = &s;
19        func(&mut s);
20        println!("{}",s);
21    }
22 }
23
24 fn greethuh<'a>(a:&'a str, b:&'a str) -> &'a str{
25     if a == "hello" {a} else {b}
26 }
27
28 fn func(s: &mut String){
29     s.push_str("y");
30     println!("{}", s,s);
31 }
```

Ownership

If there is no owner, write "NONE".

Who is the owner of "ay" immediately after line 29 is run? **s**

We are passing a mutable reference to fun.

So the owner of "ay" is still s in main.

Who is the owner of "mut" immediately after line 5 is run? **f**

The owner of "mut" is f. Strings do not have the copy trait,
so ownership is transferred to f.

Who is the owner of "bye" immediately after line 24 is run? **Dropped**

Who is the owner of "a" immediately after line 18 is run? **s**

The owner of "a" is s, since s1 is just a borrow.

Who is the owner of "hey" immediately after line 12 is run? **d**

The owner of "hey", is d since strings do not have the copy trait.

Lifetimes

At what line is the value of "bye" dropped? **14 or 15**

The last line "bye" is used is line 13, so it is dropped on line 14.

At what line is the value of "hey" dropped? **14 or 15**

The last line "hey" is used is line 13, so it is dropped on line 14.

At what line is the value of "mut" dropped? **6 or 7**

The last line "mut" is used is line 5, so it is dropped on line 6.



Version 4:

```
1 fn main(){
2     let f;
3     {
4         let x = String::from("mut");
5         f = x;
6     }
7
8     let c;
9     {
10        let y = String::from("hey");
11        let z = "bye";
12        let d = z;
13        c = greethuh(d, y.as_str());
14    }
15
16    {
17        let mut s = String::from("a");
18        let mut s1 = s;
19        func(&mut s1);
20        println!("{}", s1);
21    }
22 }
23
24 fn greethuh<'a>(a:&'a str, b:&'a str) -> &'a str{
25     if a == "hello" {a} else {b}
26 }
27
28 fn func(s: &mut String){
29     s.push_str("y");
30     println!("{}", s);
31 }
```

Ownership

If there is no owner, write "NONE".

Who is the owner of "hey" immediately after line 24 is run?

y

.as_str() method is used, which means that the owner is still y.

Who is the owner of "mut" immediately after line 5 is run?

f

Ownership is transfered to f in this line so f is the owner.

Who is the owner of "ay" immediately after line 29 is run?

s1

Ownership of "a" is transferred to s1 in line 18.

In line 29, "y" is added to s1.

Who is the owner of "a" immediately after line 18 is run?

s1

Ownership of "a" is transferred to s1 in line 18.

Who is the owner of "bye" immediately after line 12 is run?

This question was dropped

Lifetimes

At what line is the value of "bye" dropped?

14 or 15

The last time "bye" is used is in line 13, so at line 14 it is dropped.

At what line is the value of "mut" dropped?

6 or 7

The last time "bye" is used is in line 5, so at line 6 it is dropped.

At what line is the value of "hey" dropped?

14 or 15

The last time "hey" is used is in line 13, so at line 14 it is dropped.