

Homework for Ownership and Borrowing

Odd-Harald Lillestø Myhren

July 17, 2023

Understanding Rust's Ownership System and Memory Management

1. Describe ownership moves in Rust. Provide one example to illustrate this concept.
2. Explain the concept of borrowing and mutable borrowing in Rust. What are the differences between these two?
3. Demonstrate how to create references and borrow data in Rust. Provide an example code snippet.
4. Explain the process of dereferencing references in Rust. How can it be done?
5. Discuss the borrowing rules and restrictions in Rust. For instance, how many immutable and mutable references can you have at once?

Code Assignment

Here's some Rust code that tries to change the name of Frank.

```
fn main() {
    let person = String::from("Frank");
    update_name(person);
    assert_eq!(person, String::from("Ralph"));
}

fn update_name(input: String) {
    *input = String::from("Ralph");
}
```

This code fails to compile. Look at the error below, and fix the code by using a mutable reference.

```
error[E0384]: cannot assign to immutable argument 'input'
--> src/main.rs:8:5
|
7 | fn update_name(input: String) {
|               ----- help: consider making this binding mutable: 'mut input'
8 |     *input = String::from("Ralph");
|     ~~~~~ cannot assign to immutable argument
```