**Coding challenge #1**

1. Examine the following *faulty* Rust code:

**fn** square\_it**(**num**:** **i32)** **->** **i32{**

num**.**pow**(**2**)**

**}**

**fn** main**()** **{**

**let** num**:i32** **=** 5**;**

**let** result**:i32** **=** square\_it**(**num**);**

println!**(**"{} squared is {}"**,** num**,** result**);**

**}**

When you attempt to compile this code, you will discover that it accidentally breaks Rust’s ownership rules(!)

1. Modify this code so that Rust’s ownership rules are met.
2. Compiling and running the code successfully should produce the following on-screen output:

**5 squared is 25**

**Coding challenge #2**

1. Examine the following *faulty* Rust code:

**fn** is\_qa\_email**(**email**:**String**)** **->** **i32** **{**

**let** domain **=** "q.com"**;**

email**.**starts\_with**(**domain**);**

**}**

**fn** main**()** **{**

**let** email **=** String**::**from**(**"helpdesk@qa.com"**);**

**if** as\_qa\_email**(**email**) == false** **{**

println!**(**"{} is a QA account"**,** email**);**

**}**

**else** **{**

println!**(**"{} is not a QA account"**,** email**);**

**}**

**}**

When you attempt to compile this code, you will discover that it accidentally breaks Rust’s ownership rules(!)

1. Modify this code so that Rust’s ownership rules are met *and* the various bugs, typos, and semantic errors are fixed.
2. Compiling and running the code successfully should produce the following on-screen output:

**helpdesk@qa.com is a QA account**

1. Examine the following *faulty* Rust code:

**fn** modify\_ext**(**old\_file**:**String**)** **->** usize **{**

old\_file**.**push\_str**(**".json"**);**

**return** old\_file**.**len**()**

**}**

**fn** main**()** **{**

**let** tempfile**:**String **=** String**::**from**(**"myfile"**);**

**let** namelen**:** usize **=** tempfile**.**len**();**

println!**(**"filename is {} chars long"**,** namelen**);**

namelen **=** modify\_ext**(**tempfile**);**

println!**(**"filename is now: {}"**,** tempfile**);**

println!**(**"filename is now {} chars long"**,** namelen**);**

**}**

When you attempt to compile this code, you will discover that it accidentally breaks Rust’s ownership rules(!)

1. Modify this code so that Rust’s ownership rules are met.
2. Compiling and running the code successfully should produce the following on-screen output:

**filename is 6 chars long**

**filename is now: myfile.json**

**filename is now 11 chars long**