COMP 333
Summer 2025

Structs and Methods in Rust (Answers)

1.a.) Declare a struct named `MyStruct` that holds two fields: `first_string` (which holds a `String`), and `second_string` (which holds another String).

```
struct MyStruct {
    first_string: String,
    second_string: String
}
```

1.b.) Implement a method for `MyStruct`, which takes a reference to a `MyStruct` instance, and returns a reference to the `first_string` field. The method should be named `get_first_string_ref`.

```
impl MyStruct {
    fn get_first_string_ref(&self) -> &String {
        &self.first_string
    }
}
```

1.b.) Implement another method for `MyStruct`, called `with_second_string`. This should take ownership over a `MyStruct` instance, and additionally take ownership over another `String` parameter. It should return a new `MyStruct` instance, holding the same `first_string` as the original `MyStruct` instance, along with the second `String` parameter. Example usage is below:

```
let original = MyStruct {
  first_string: "foo".to_string(),
  second_string: "bar".to_string()
};
let new_version =
  original.with_second_string("apple".to_string());
println!("{}", new_version.first_string);
println!("{}", new_version.second_string);


---Output---
foo
apple

impl MyStruct {
    // unchanged from before
    fn get_first_string_ref(&self) -> &String {
        &self.first_string
    }

    // new method
    fn with_second_string(self, other: String) -> MyStruct {
        MyStruct {
            first_string: self.first_string,
            second_string: other
        }
    }
}
```