0/6 Questions Answered

6 questions with unsaved changes

	STUDENT NAME					
Search students by name or email						
Q1 Scoping & Borrowing 4 Points						
<b>21.1</b> Ov 2 Points	wnership					
pub f }	<pre>n add_bye(a: &amp;mut String) {     a.push_str("bye");</pre>					
pub f }	<pre>n add_period(mut a: String)-&gt;String {     a.push_str(".");     return a;</pre>					

a
 Data is out of scope
 Save Answer \*Unsaved Changes

## Q1.2 Lifetimes

2 Points

The following code takes two strings and returns the shorter of the two strings.

```
fn shortest (x:&str, y:&str) -> &str {
    if x.len() < y.len() { x } else { y }
}</pre>
```

It works fine until we have code like this:

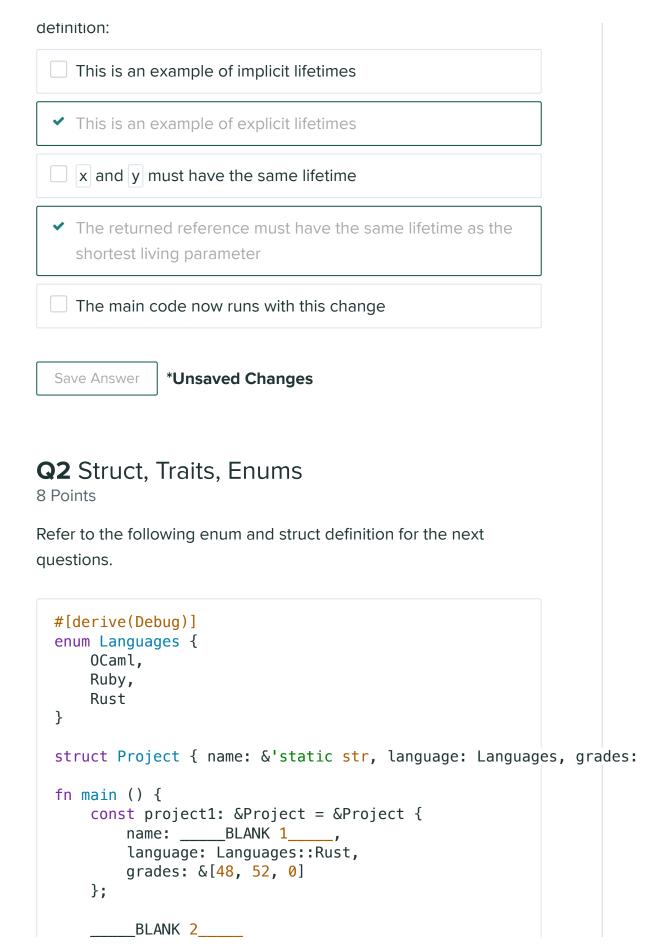
```
fn main () {
    let x = String::from("there");
    let z;
    {
        let y = String::from("hi");
        z = shortest(&x,&y); //will be &y
    } //drop y, and thereby z
    println!("z = {}",z); //yikes!
}
```

We can help mitigate this by having the function definition to be:

fn shortest<'a>(x:&'a str, y:&'a str) -> &'a str {...}

Select the statements that are true about the updated function

. . ...



} Q2.1 1 Point What would go in place for BLANK 1 so that the project name is Stark Suit Repair? String::from("Stark Suit Repair") "Stark Suit Repair"  $\checkmark$ &String::from("Stark Suit Repair") ✓ &"Stark Suit Repair" Save Answer \*Unsaved Changes Q2.2 1 Point Which of the following code options would print Rust at BLANK 2? Select all that apply. println!("{:?}", project1.language); ✓ println!("{}", project1.language); println!("{:#}", project1.language); println!("{:#?}", project1.language);  $\checkmark$ nrintln!("{·?}" language)·

princense (is) , canguage/, \*Unsaved Changes Save Answer Q2.3 Traits 6 Points Let's say we wanted to implement the Assignment trait to the struct Project. It is defined as the following: trait Assignment { fn maxPoints(&self) -> i32; fn quiz() -> Project { // returns a project Project { name:"Quiz", language: Languages::Rust, grades: &[4,8,8] } } }

Complete the following code so that Project is an Assignment by filling in the blanks,

```
___Blank 3 ____{
fn maxPoints(&self) -> i32 { // sums up points
    let mut total = 0;
    for &i in self.grades {
        total += i;
      }
      return total;
    }
}
fn main () {
    let q = __Blank 4___;
    println!("The total possible points for {} is {}.", q.name, ___!
    //Should print out the sum of q's grades
}
```

Blank 3 (extending project to implement assignment) impl Assignment for Project Blank 4 (getting the quiz assignment) Project::quiz() Blank 5 (calling the method, you will not receive points for hardcoding this) q.maxPoints() \*Unsaved Changes Save Answer **Q**3 8 Points fn shortest(x: &str, y: &str) -> &str { if x.bytes().len() < y.bytes().len() {</pre> Х } else { y.bytes() } } fn main() { let hi = String::from("HelloCMSC330"); let hello = "WinterBreakLoading"; let l = shortest(hi, hello);

```
println!("{}", hi);
```

}

There exists three errors/bugs to the code above. State 2 of the bugs and provide a fix to them.

## Bug 1

shortest expects two borrowed strs, but we send them str which if done is a type error but also means the data "HelloCMSC330" would be dropped. Fix can be changing `let I = shortest(&hi, &hello);` or changing variable hi to "HelloCMSC330"

## Bug 2

shortest expects explicit lifetime annotations, can be fixed with fnshortest<'a>(x: &'a str, y:'a str) -> &'a str`

Sav	ve	An	SW	e
				_

## \*Unsaved Changes

Save All Answers

Submit & View Submission >