COSC 455

Programming Languages: Design and Implementation

Fall 2025

**Lab Assignment #4**

**Name: Blessing Abumere / 10**

**Goals:** This lab consists of 3 required tasks. The intention of this lab exercise is to set up the development environment and tools that we will be using throughout most of the semester, including Slack (<https://slack.com/>), GitHub (<https://github.com/>), Visual Studio Code (<https://code.visualstudio.com/>) with Rust development tools (<https://code.visualstudio.com/docs/languages/rust>). In doing so, you should start to develop some familiarity with an industry-like development environment and tools and write and commit some more Rust programs to your GitHub account.

**Environment:** The Visual Studio Code IDE using the Rust development tools and cargo must be used in this lab along with Slack to communicate and submit portions of the lab. No other environment, programming language or submission will be allowed.

**Submission:** All lab exercises should be submitted via Slack/GitHub as indicated in the lab description.

**Deadline:** Submitted via Blackboard/GitHub by 11:59pm on Monday, [September](https://www.youtube.com/watch?v=rdpBZ5_b48g) 22, 2025

[Repository](https://github.com/Bless-04/COSC-455---PROGRAMMING-LANGUAGES-DESIGN-IMPLEMENTATION/tree/master/Labs/Lab%204/Circle)

1. **A Bit More Defeating the Rust Learning Curve Climb.** (3 points) As mentioned in class and in prior labs, this semester will utilize the Rust programming language for a significant project implementing some aspects of a compiler/parser and will be used throughout to demonstrate various programming language aspects. While I will not directly teach Rust, we will (yes, I am suffering along with you ☺) utilize the excellent Rust Essential Training LinkedIn Learning course, found at <https://www.linkedin.com/learning/rust-essential-training>. Note that LinkedIn Learning is completely free to all TU students (you must sign in using your TU email address). This is course is 6+ hours, but much of it assumes very little prior programming experience and I have found that watching/following along at 1.25-1.5x speed works for me. We will also tackle this in smaller chunks over the next few weeks to not overwhelm.

For this lab, please complete video Chapters 8 – 10 (Modules, Input/Output, Structs); note that we skipped a few chapters, for now. While this is somewhat long, it is essentially just converting your Java/Python syntax/understanding into Rust, so much of it can be done at a faster video speed. However, in this case, please focus on the notion of Structs (i.e., Chap 10), as are they are like Java objects but not exactly the same.

While the payoff for completion of this task will mostly come when we get to implementing the simplified compiler/parser later this semester, to demonstrate your completion of this task for now, please utilize a Rust/cargo project (it may be the same as from Lab #2) with the main.rs Rust file to work through the demonstrations/examples in the videos for your understanding. That is, I should see some of the examples in your committed source code from these chapters.

**Once you have completed this task, commit/push your project to the same GitHub repository.** Without this, you will not receive credit for this portion of the lab.

[Chapter 8](https://github.com/Bless-04/COSC-455---PROGRAMMING-LANGUAGES-DESIGN-IMPLEMENTATION/tree/master/Labs/Lab%204/challenge_8)

[Chapter 10](https://github.com/Bless-04/COSC-455---PROGRAMMING-LANGUAGES-DESIGN-IMPLEMENTATION/tree/master/Labs/Lab%204/challenge_10)

1. **More Rust Practice.** (3 points) In a new Rust/cargo project, consider the following Rust starter code:

struct Rectangle {

width: f64,

height: f64,

}

impl Rectangle {

fn new(width: f64, height: f64) -> Rectangle {

Rectangle { width, height }

}

fn area(&self) -> f64 {

// TODO: implement

}

fn perimeter(&self) -> f64 {

// TODO: implement

}

fn is\_square(&self) -> bool {

// TODO: implement

}

}

fn main() {

let rect = Rectangle::new(10.0, 5.0);

println!("Area: {}", rect.area());

println!("Perimeter: {}", rect.perimeter());

println!("Is square? {}", rect.is\_square());

assert!(Rectangle::new(5.0, 5.0).is\_square());

assert!(!Rectangle::new(5.0, 6.0).is\_square());

}

Add code comments to this to demonstrate your understanding and complete the few functions noted with TODOs to ensure that this program works correctly and passes the two test cases (i.e., asserts) at the end. Note, I know that ChatGPT can solve this easily but, in the upcoming implementation of a compiler that we will get to, as well as on exams, it will not suffice and you will need to have some comfort of Rust to do well…so, get your practice now.

**Once you have completed this task, commit/push this project to your GitHub repository** (make sure that I (cosc455dehlinger) have access to it**.** Without this, you will not receive credit for this portion of the lab.

[**Rectangle**](https://github.com/Bless-04/COSC-455---PROGRAMMING-LANGUAGES-DESIGN-IMPLEMENTATION/tree/master/Labs/Lab%204/Rectangle)

1. **Even More Rust Practice.** (4 points) Building on the same Rust project as Question 2, consider the following Rust struct for a circle:

struct Circle {

radius: f64, // assume radius > 0

}

With this starter code, provide an implementation (impl) with completed functions for:

* Creating a new circle, providing its radius as a parameter
* Calculates and returns the area, using [pi](https://www.youtube.com/watch?v=VslRTJdPMIc&ab_channel=EjkopArchive) \* radius2 where, for simplicity, pi = 3.14
* Calculates and returns the circumference, using 2 \* [pi](https://www.youtube.com/watch?v=_7CClt4FQvw&ab_channel=TheSimpsonsThrowback) \* radius

Be sure to include code comments to document and demonstrate your understanding. In addition, add a few lines to the main function to declare a new circle and prints out its area and circumference. Again, I know ChatGPT can do this, but you should do this on your own.

**Once you have completed this task, commit/push this project to the Question 2 GitHub repository** (make sure that I (cosc455dehlinger) have access to it**.** Without this, you will not receive credit for this portion of the lab.

[Circle](https://github.com/Bless-04/COSC-455---PROGRAMMING-LANGUAGES-DESIGN-IMPLEMENTATION/tree/master/Labs/Lab%204/Circle)

**Finally, to indicate you have completed this lab, submit your GitHub repository link in a text/Word document to Blackboard.**

**Note:** There are at least 3 [Easter eggs](https://en.wikipedia.org/wiki/Easter_egg_(media)) (this isn’t one of them) in this lab as well as one in the Course Syllabus. If you find one, Slack me and let me know!