### **Lab 2:** **Trait in Rust**

**Exercise: Shape Area Calculator**

In this exercise, we will define a trait called Shape that represents shapes and requires them to implement a method to calculate their area. Then, you will create two structs that implement the Shape trait: Rectangle and Circle. Finally, we will calculate and display the areas of these shapes.

1. Create a new Rust project using cargo:

Open your terminal/command prompt and run the following command:

cargo new shape\_area\_calculator

cd shape\_area\_calculator

1. Open the main.rs file in the src directory of your project. You can use any code editor for this.
2. Define a trait called Shape with a method to calculate the area:

trait Shape {

fn area(&self) -> f64;

}

1. Implement the Shape trait for two different structs, Rectangle and Circle:

struct Rectangle {

width: f64,

height: f64,

}

impl Shape for Rectangle {

fn area(&self) -> f64 {

self.width \* self.height

}

}

struct Circle {

radius: f64,

}

impl Shape for Circle {

fn area(&self) -> f64 {

std::f64::consts::PI \* self.radius \* self.radius

}

}

In the main function, create instances of Rectangle and Circle, calculate their areas, and display the results:

fn main() {

let rectangle = Rectangle {

width: 5.0,

height: 10.0,

};

let circle = Circle {

radius: 7.0,

};

println!("Area of the rectangle: {}", rectangle.area());

println!("Area of the circle: {}", circle.area());

}

1. Save the file and return to your terminal/command prompt.
2. Build and run your program using cargo run:

cargo run

The program will display the areas of the rectangle and the circle.

Example Output:

Area of the rectangle: 50

Area of the circle: 153.93804002589985

We have now successfully completed the lab exercise on traits in Rust, creating a program to calculate and display the areas of different shapes using a trait!

**Happy coding!**