

[[questions]]

type = "Tracing"

prompt.program = ""

#[derive(Drop)]

struct Rectangle {

width: u64,

height: u64,

}

fn perimeter(r: Rectangle) -> u64 {

2 * (r.width + r.height)

}

fn main() {

let rectangle = Rectangle { width: 20, height: 10, };

let p = perimeter(rectangle);

println!("2 * ({} + {}) = {}", rectangle.width, rectangle.height, p);

}

""

answer.doesCompile = false

context = "The area function takes ownership of its argument `rectangle`, which doesn't implement `Copy`. Calling `perimeter(rectangle)` therefore moves `rectangle`, meaning it cannot be used on the next line."

id = "3d5a7161-f117-46c6-a293-ccbabe4b4a9d"

[[questions]]

type = "Tracing"

prompt.program = ""

struct Point { x: u128, y: u128, }

fn main() {

let p = Point { x: 5, y: 10 };

let Point { z, w } = p;

println!("The values of z and w are: ({}), {}", z, w);

}

""

answer.doesCompile = false

context = "Destructuring refers to the process of extracting individual fields from a struct and binding them to separate variables. When destructuring a struct, you can either use variables with the same names as the struct fields or explicitly bind the fields to variables with different names using the syntax `field_name: variable_name`."

id = "da04e96e-b05c-489d-819a-07e53cc4fee4"