

Hack postgres Source Code: Vol I

Chapter 1 : C & Rust

1.14 Constructor & Methods in Structs in Rust. Making C and postgres Nap for Sometime.

Rust

```
/*
 * Create structure type Square.
 * As in OOPS, struct in rust is like a class somewhat.
 */

struct Square {
    width: u32,
    height: u32,
} // No need of semi colon

/*
 * Implementation of a constructor and a method.
 * we can associate constructor and methods to structure
 * using `impl` key word.
 */

impl Square {

    /*
     * Constructor has return type same as structure to which it
     * it is associated with.
     * It doesn't have `self: &Self` as first parameter.
     * `Self` inside `impl` implementation represents
     * structure type ( In this case `Square` ).
     * This is also called as associate function.
     * This constructor creates instance of Square.
     */

    fn square_constructor(size: u32) -> Square {
        Square {
            width: size,
            height: size,
        }
    }

    /*
     * Ownership system rules are also applied to constructors and
```

```
    * methods in `impl` implementation.
    * We can also use `&self` as shorthand for self: &Self.
    * In this method, we are receiving reference of
    * Square (borrowing Square from `main`) instance
    * and returning area.
    */

    fn square_area(self: &Self) -> u32 {

        self.width * self.height
    }
}

fn main() {

    /*
    * We use `::` to access constructor of struct type.
    */
    let square_instance: Square = Square::square_constructor(3);
    let square_instance_area: u32 = square_instance.square_area();

    println!("Square are is {}", square_instance_area);

}
```