

# RUST PROGRAMMING COURSE NOTES

DAVID YANG

## 1. WEEK 1: RUST BASICS

---

```
// fields go in structs
struct Dog {
    breed: String,
    age: u32, // unsigned
}

// methods/functions go in "impl" block
impl Dog {
    fn bark(&self) {
        println!("bark!");
    }
}

// add and max functions
fn add(a: i32, b: i32) -> i32{
    a + b // or "return a + b;"
}

fn max(a: i32, b: i32) -> i32{
    if a > b {
        a
    }

    else {
        b
    }
}

fn order(a: i32, b: i32) -> (i32, i32) {
    if a > b { (b, a) } else { (a, b) }
}

fn my_name() -> String {
    "David".to_string()
}

fn main() {
    // print statement
```

```

println!("Hello, world!");

// for loop
for i in 0..10 {
    println!("i is: {}", i);
}

// defining immutable int
let x: i32 = 0;
println!("{}", x);

// array
let arr = [1,2,3,4];

// using classes
let sparky = Dog {
    breed: "Chihuahua".to_string(),
    age: 4,
};

sparky.bark();

// testing functions
println!("{}", add(4, 5));

let x: i32 = -5;

// no parentheses around conditionals
let mut abs: i32 = if x > 0 {
    x
} else {
    -x
};

abs += 1;

}

```

---