```
    // * => multiplication
    // / => division
    // % => modulo operator

// LOGICAL OPERATORS
    // && => logical and
    // || => logical or
    // ! => logical not

// COMPARISON OPERATORS
    // == => complete equality check for value and type
    // != => complete inequality check for value and type
    // > < >= <= are also comparison operators
```

Control structures

```
// ---------- CONTROL STRUCTURE ----------
    // Go does not have brackets to segregate its predicate conditional and loop statements

// CONDITIONALS

// IF ELSE IF ELSE

x := 10
if x > 10 {
    fmt.Print("x is greater than 10")
} else if x < 10 {
    fmt.Print("x is smaller than 10")
} else {
    fmt.Print("x is equals to 10")
}

// SWITCH CASE DEFAULT
    // note there is no break statement required for each case statement here unlike other languages
    // default => specifies the default case if logic falls through every other case statement within the switch clause

day := "Monday"
switch day {
```