Solution Review 2: Make a Calculator

This lesson gives a detailed solution review to the challenge in the previous lesson.

We'll cover the following

- Solution :
  - Explanation

Solution :

```
fn test(a: i32, operator: char ,b: i32) {
    match operator {
            '+' => {
                println!("{}", a + b);
            },
            '-' => {
                println!("{}", a - b);
            },
            '*' => {
                println!("{}", a * b);
            },
            '/' => {
                if b == 0{
                    println!("Division by 0 is undefined");
                }
                else {
                    println!("{}", a / b);
                }
            },
            '%' => {
                println!("{}", a % b);
            },
            _ => println!("{}", "invalid operator"),
        }
}
fn main(){
    print!("3 + 2: ");
    test(3,'+',2);
    print!("3 - 2: ");
    test(3,'-',2);
    print!("3 * 2: ");
    test(3,'*',2);
    print!("3 / 2: ");
    test(3,'/',2);
    print!("3 % 2: ");
    test(3,'%',2);
    print!("3 ( 2: ");
    test(3,'(',2);
    print!("3 ( 0: ");
```

```
    test(3, '/', 0)
}
```

Explanation

- **`match` construct**

  A `match` construct is defined from **line 2 to line 19**.

  - On **line 2**, the `match` statement takes an `operator` variable.
    - On **line 3**, checks if the operator variable is equal to `+` then it displays the result of addition on **line 4**.
    - On **line 6**, checks if the operator variable is equal to `-` then it displays the result of subtraction on **line** 7.
    - On **line 9**, checks if the operator variable is equal to `*` then it displays the result of multiplication on **line 10**.
    - On **line 12**, checks if the operator variable is equal to `/`, and the dividend is equal to 0 then it displays that it is not possible to divide the number by 0 on **line 14**, else it displays the result of division on **line 17**.
    - On **line 20**, checks if the operator variable is equal to `%` then it displays the result of modulus on **line 21**.
    - On **line 23**, checks if the operator variable does not belong to the above then it prints “invalid” on **line 23**.

  The following illustration explains with $a = 3$ and $b = 2$.

Subtraction

Multiplication

Division

4 of 5

Modulus

5 of 5

Now that you have learned about conditional statements, what if you want to execute a sequence of statements multiple times? Let's learn about this in the next chapter, "Loops".