Day 51

28/04/2025

Tuple deconstruction

```
(int a, int b) = (2, 3);
(int a, int b, string c) = (2, 3, "Hi");

int x = 10;
int y = 20;
  ↑ ↑
(x, y) = (3, 4);
```

```
int a = 10
int b = 20
int c = a
a = b;
b = c;
```

```
(a, b) = (b, a)
```

we can easily reassigns the value.

using Tuples we swap the value of a, b easily without creating extra value.

```
var value = (10, 20, "Hi", 30)
            ↓
int a = value.Item1;
        ↓
        10
```

---

```
public (int, string) Add
                     (int a, int b)

[illegible]
if (int.TryParse(a)
{
    return (a+b, "Total");
}
else
{
    return("", "error");
}
```

Object called.

```
Math emp = new math();
(int, string) value = emp.Add(1,2)
{var, dynamic} -> any allowed
int tota = value.Item1;
String value.Item2;
```

---

```
public int Add(int a, int b)
{
    return a+b;
}
```

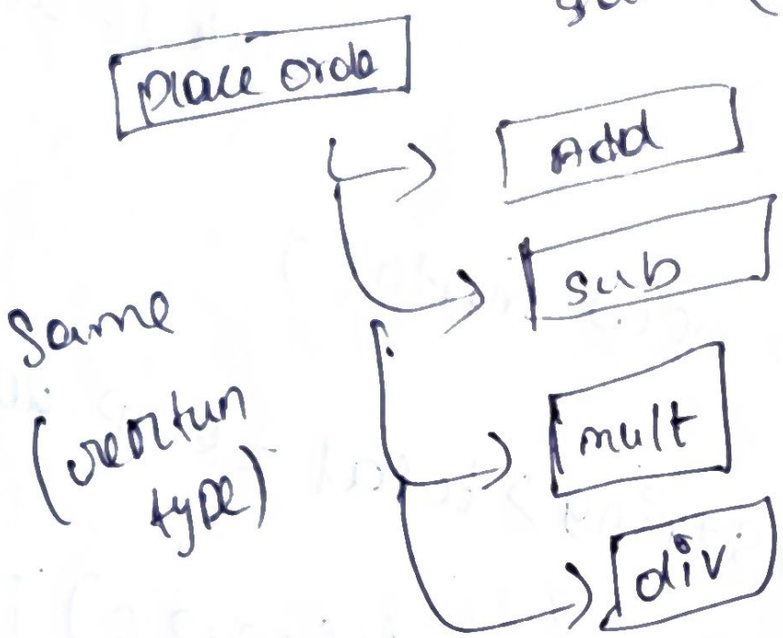
---

```
public delegate int addall(int a, int b)
math Emp = new math();
addall b = emp.add; => it will
b(1,2); call the method
```

Place orders

(multiple casting delegate)

same (parameters)




```
math emp = new math();
add all a = emp.add;
      a += emp.sub;
      a += emp.mult;
      a += emp.Div;
```

it will call all the four method in math.

// Action - void, any param

// Func - any value, any param

// predicate - bool, 1 param

↓

true / flase.

Func

Math emp = new math()

Func<int, int, string> total = emp.add;

String sum = total(10,20);

ACTION

public void todisplay(int a)

{

Console.writeline(a.tostring);

}

Action<int> act = emp.todisplay

act(10);

# Predicate

```
predicate <string> pred = emp.vale;
bool val = pred("hi");
public bool vale(string vale);
{
    return true;
}
```