17/02/2023

Camlin Page
Date / /

Day 9

Programming paradigms :-
There are two commonly used paradigms in programming : 1) Functional programming
2) Object-oriented programming.

1) Functional programming :-
- abbreviated as FP
- There is clear distinction between data and functions in functional programming as data can exist outside of function

```
var course = 100;          ——→ data
var GST    = 1.18;
 function totalprice (price, tax){   ——→ function
       return price * tax;
 }
 var topay = totalprice (course, GST);
 console.log(topay);
```

2)

* Scope - It is all about code accessibility. It determines which parts of the code are accessible and which parts are inaccessible

```
var num1 = 10 ;            ——→ global scope

function score () {
     var num2 = 20;        ——— local scope
     console.log (num2);
}
```

2] Object-oriented programming
- often referred to as OOP.
- OOP revolves around the idea of organi
our program using objects to group related data
and functionability.

```
var purchase1 = {
        course : 100,
        GST : 1.18,
        totalprice : function () {
                var calculation = purshase1.course *
                                  purchase1.GST;
                console.log ('Total price :', calculation
        }
}

purchase1.course ;        // 100
purchase1.GST ;           // 1.18
```

⇓

```
var purchase2 = {
        course : 200,
        GST : 1.18,
        totalprice : function () {
                var calculation = this.course * this.GST;
                console.log ('Total price :', calculation
        }
}

purchase2.totalprice();        // 218
```

Instead of - With the object oriented approach we can code more efficiently by reusing existing code

Camlin Page
Date / /

* Classes :-

In javascript any class is built using the class keyword, followed by the ~~ca~~ name of the starting with capital letter and pair of curly braces. Inside of the curly braces we have the constructor function which accepts as many parameters as needed

```
class Car {
    constructor (color, speed){
        this.color = color;
        this.speed = speed;
    }
    turboOn(){
        console.log ("turbo is on!")
    }
    const car1 = new Car ("red", 120)
```

Class is a blueprint that we can repeatedly use to build new object of certain kind, as many time as we like.

* Inheritance -

```
var bird = {
    hasWings : true,
    canFly : true,
    hasFeathers : true
}
var eagle = object.create(bird);
console.log("eagle:", eagle);
console.log("eagle can fly :", eagle.canFly); //true
console.log("eagle has wings :", eagle.hasWings); //true
```