Classes & Objects

Prototypes in JS

A javaScript object is an entity having state and behavior (properties and method).

JS objects have a special property called prototype.

We can set prototype using _ _ proto _ _

it's used in object similar to extends in class

```
const employee = {
   calcTax () { console.log("Tax")
     }
}

const stu1={
   sal : 1000;
 }

stu1.__ proto__ = employee;
```

Now He can access this calcTax function

***If object & prototype have same method, object's method will be used.**

Classes in JS

Class is a program-code template for creating objects.

Those objects will have some state (variables) & some behaviour (functions) inside it.

```
class MyClass {
    constructor( ) { ... }
    myMethod( ) { ... }
}

let myObj = new MyClass( ) ;
```

Classes in JS

Constructor( ) method is :

- automatically invoked by new
- initializes object

```
class MyClass {

    constructor( ) { ... }

    myMethod( ) { ... }

}
```

Inheritance in JS

inheritance is passing down properties & methods from parent class to child class.

```
class Parent {

}

class Child extends Parent {

}
```

*If Child & Parent have same method, child's method will be used. [Method Overriding]

super Keyword

The super keyword is used to call the constructor of its parent class to access the parent's properties and methods.

```
super( args )  // calls Parent‘s constructor

super.parentMethod( args )
```

Let‘s Practice

Qs. You are creating a website for your college. Create a class User with 2 properties, name & email. It also has a method called viewData( ) that allows user to view website data.

Qs. Create a new class called Admin which inherits from User. Add a new method called editData to Admin that allows it to edit website data.

Error Handling

try-catch

```
try {

    ... normal code

} catch ( err ) { //err is error object

    ... handling error

}
```