**JavaScript – Oops Demo code**

**Class Example:**

<!DOCTYPE html>

<html>

<body>

<script>

//Declaring class

class Employee

  {

//Initializing an object

    constructor(m1,m2)

  {

      this.m1=m1;

      this.m2=m2;

    }

    detail()

    {

      var a,b,c;

      a = this.m1;

      b = this.m2;

      c = (a+b)/2;

  document.writeln("Mark1"+a+"<br>"+"Mark2:"+b+"<br>"+"Average:"+c+"<br>")

    }

  }

//passing object to a variable

var e1=new Employee(a,b);

//var e2=new Employee(102,"Duke William");

e1.detail(); //calling method

e2.detail();

</script>

</body>

</html>

**Class with setter and getter Method Example:**

<!DOCTYPE html>

<html>

<body>

<script>

class Stu

{

    getidnname()

    {

        return this.id+" "+this.name;

    }

    setidnname(id,name)

    {

        this.id=id;

        this.name=name;

    }

    }

      Stu.prototype.dept="IT"

      var st1 = new Stu();

      st1.setidnname(101,"Ram");

//st1.disp();

  var str = st1.getidnname();

  document.write(str+" "+st1.dept);

        //document.writeln(st1.id+" "+st1.name+" "+st1.dept+"<br/>");

</script>

</body>

</html>

**Inheritance Example:**

<!DOCTYPE html>

<html>

<body>

<script>

//Declaring class

class Employee

  {

//Initializing an object

    constructor(m1,m2)

  {

      this.m1=m1;

      this.m2=m2;

    }

    detail()

    {

      var a,b,c;

      a = this.m1;

      b = this.m2;

      c = (a+b)/2;

  document.writeln("Mark1"+a+"<br>"+"Mark2:"+b+"<br>"+"Average:"+c+"<br>")

    }

  }

//passing object to a variable

var e1=new Employee(a,b);

//var e2=new Employee(102,"Duke William");

e1.detail(); //calling method

//e2.detail();

//Re-declaring class

class Employee1 extends Employee

{

  detail1()

  {

    super();

    var d = a+b+c;

    document.writeln(d);

  }

}

var e11 = new Employee1();

e11.detail();

e11.detail1();

</script>

</body>

</html>

**Example 2: Inheritance**

<!DocType html>

<html>

<body>

<script>

class Student {

    constructor()

    {

        this.id = 101;

        this.name="Ram";

    }

    disp()

    {

        var a = 50;

        document.write("This is the base class");

    }

}

class Student1 extends Student

{

    constructor(dept)

    {

        super();

        this.dept = dept;

    }

}

Student1.prototype.grade = "A";

var s = new Student1("IT");

s.disp();

//s.details1();

document.writeln("<br/>"+s.id+" "+" "+s.name+" "+s.dept+" "+s.grade);

</script>

</body>

</html>

**Abstraction:**

Abstraction means hide the unnecessary details and showing the important or necessary Details. In Abstract class, not able to define the methods, properties. We can just declare the definition only, not able to give the definition inside the function or methods.

**Example:**

In the following program it shows the error when we define the function vechicle.

<!DOCTYPE html>

<html>

<body>

<script>

//Creating a constructor function

 function Vehicle()

{

    this.vehicleName="vehicleName";

    throw new Error("You cannot create an instance of Abstract Class");

}

Vehicle.prototype.display=function()

{

    return "Vehicle is: "+this.vehicleName;

}

//Creating a constructor function

function Bike(vehicleName)

{

    this.vehicleName=vehicleName;

}

//Creating object without using the function constructor

Bike.prototype=Object.create(Vehicle.prototype);

var bike=new Bike("Honda");

document.writeln(bike.display());

 </script>

</body>

</html>

**Polymorphism:**

It means the ability to take more than one form.

In the Following Example , we have the same method name in two classes. So it is defined as polymorphism. But it can do different functionalities.

<!DOCTYPE html>

<html>

<body>

<script>

class A

  {

     display()

    {

      document.writeln("A is invoked<br>");

    }

  }

class B extends A

  {

    display()

    {

      document.writeln("B is invoked");

    }

  }

  // vara = new A();

 // var b = new B();

 // a.display();

  //b.display();

var a=[new A(), new B()]

a.forEach(function(msg)

{

msg.display();

});

</script>

</body>

</html>

**Exception Handling:**

<html>

<head> Exception Handling</br></head>

<body>

<script>

try{

var a= ["34","32","5","31","24","44","67"]; //a is an array

document.write(a);    // displays elements of a

document.write(b); //b is undefined but still trying to fetch its value. Thus catch block will be invoked

}catch(e){

alert("There is error which shows "+e.message); //Handling error

}

</script>

</body>

</html>

**Example 2: Using Throw Keyword**

<!DocType html>

<html>

    <body>

        <script>

            const a = 30;

            try {

             if(a>50)

             {

                document.write(a);

             }

             else  {

                throw "a is below 50";

            }

            }

            catch(e)

            {

                document.write("Input is " + e);

                ///document.write(error);

            }

        </script>

    </body>

</html>

**External js class calling from the HTML file Example:**

<!DOCTYPE html>

<html>

<head>

    <script src="./c1.js" onclick="func()"></script>

</head>

<body>

    <button type="button">Click</button>

    <input type="text" id="txt1">

<script>

var user = new TryClass( 101, "John" );

user.sayHi();

var b = user.getval();

document.write(b);

var myArray = b.split(" ");

////document.write(myArray[0]+myArray[1]);

document.getElementById("txt1").innerHTML=myArray[1];

</script>

</body>

</html>

**C1.js:**

class TryClass {

    constructor(id,name) {

      this.id = id;

      this.name = name;

    }

    getval() {

      return this.id+" "+this.name;

  }

    sayHi() {

      alert( this.name );

    }

  }

Example 2:

<!DOCTYPE html>

<html>

<head>

    <script type="text/javascript"

        src="module1.js">

    </script>

</head>

<body>

    <button onclick="f()">

        Click Me To Get Student Details

    </button>

    <div>

        <p id="text" style="color:purple;

            font-weight:bold;font-size:20px;">

        </p>

    </div>

    <script type="text/javascript">

        function f() {

            var name = Student.name;

            var age = Student.age;

            var dept = Student.dept;

            var score = Student.score;

            var str = "Name:" + name + "\nAge: "

                + age + "\nDepartment:" + dept

                + "\nScore: " + score;

            document.getElementById(

                'text').innerHTML = str;

        }

    </script>

</body>

</html>

Module1.js

var Student =

{

    name : "ABC",

    age : 18,

    dept : "CSE",

    score : 90

};