**OOJS JavaScript Assignment (12-03-19)**

**Q1. Create a hierarchy of person, employee and developers.**

**Ans:-** With the help of prototype, one object can refer to the other object’s prototype which usually contains the references for the existing data.

function Person(name,age){

this.name=name;

this.age=age;

}

function Employee(salary){

this.salary=salary;

}

function Developers(comp){

this.comp=comp;

}

Employee.prototype=new Person("Alpha",21);

Developers.prototype=new Employee("5000")

var dev=new Developers("Feen");

console.log(dev.name); //shows only “Alpha”

console.log(dev); //show all the content in its prototype which also refers to Employee and Person.

**Q2. Given an array, say [1,2,3,4,5]. Print each element of an array after 3 secs.**

Ans: Here is the program for printing the array.

var a=[1,2,3,4,5];

var len=a.length-1;

var i=0;

function print() {

if(i<=len){

console.log(a[i]);

i++;

}

}

setInterval(print, 3000);

**Q3. Explain difference between Bind and Call (example).**

Ans:- **Call()** is an alternate means to **invoke** the function. call() takes a context parameter as the first argument: and accessible using this inside the function body.

var sayHello = function (str) {

str ='Hello';

console.log(str, this.name);

};

var al = {name: ‘Alpha’};

sayHello.call(al); // Hello Alpha.

- **Bind()** is use to fixor can say refer the value of **this** to a specific object(context).

var person = {

name: 'Abhishek',

print: function () {

console.log('Name is: ', this.name); // this.name means person.name in this context

}

};

var p = person.print;

p(); // won't print the name

p = p.bind(person);

p(); // will print 'Abhishek'

**Q4. Explain 3 properties of argument object**.

Ans:- Argument object actually an default array object exist in function parameter section. Here are the # properties of argument object.

-arguments.callee

Reference to the currently executing function that the arguments belong to.

-arguments.caller

Reference to the function that invoked the currently executing function.

-arguments[iterator]

Returns a new [Array iterator](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/@@iterator) object that contains the values for each index in the arguments.

**Q5. Create a function which returns number of invocations and number of instances of a function.**

Ans:- The function which counts the function invocations and its object.

function fun1(){

if(this==window)

inv++;

else

ins++;

}

**Q6. Create a counter using closures.**

**Ans:** Simple closure program that just return the value.

function outer()

{

var counter = 0;

function inner()

{

counter += 1;

}

inner();

return counter;

}