**Solution 1:** Fix using wrapper, bind, call, apply.

function askPassword(ok, fail) {

let password = prompt("Password?", "");

if (password == "rockstar") ok();

else fail();

}

let user = {

name: "John",

loginOk() {

alert(`${this.name} logged in`);

},

loginFail() {

alert(`${this.name} failed to login in`);

},

};

//askPassword(user.loginOk,user.loginFail);

askPassword(

() => user.loginOk(),

() => user.loginFail()

);

**Comment**: since we are calling function askPassword from global context, the this, will refer to calling context that is global. This global context does not have the methods loginOk and loginFail. Due to this the error occurred while running the code.

To fix error, we need ensure that the this context is preserved. For, this we can defined the wrapper by using arrow function () => **user.loginOk()** and () => **user.loginFail()** respectively

**Solution 2:**

let group={

title:"Our Group",

students:["John","Pete","Alice"],

showList:function(){

// this.students.forEach(function(student){

// console.log(this.title+ ':'+student);

// });

this.students.forEach((student)=>{

console.log(this.title+ ':'+student);

});

}

}

group.showList();

**Comment**: Here this is in the inner function and will refer to global context and the global context does not have the title property. Hence, the result like undefined John is seen. To find the correct result we can create a wrapper inside loop as (student)=>{console.log(this.title+’:’+student}. Doing this the, this will refers to current context that is group