Problem 1: What will be printed?

Code 1

```
let str = "Greetings, ";
let user = {
    firstName: "John",
    lastname: "Smith",
    display: function () {
        console.log(str, this.firstName);
        show("hi");
    },
};
user.display();
function show(msg) {
    console.log(msg + " " + this.lastname);
}
show.call(user, "hello");
```

Output Code1

```
Greetings, John
hi undefined
hello Smith
```

Lexical Environment Diagram For Code 1

```
1. Global EC creation:
Outer: null, this: Window, LE: [{show: fn}], TDZ{str, user}
2. Global EC execution:
Outer: null, this: Window, LE: [{show: fn, str: "Greetings, ", user: obj}],
TDZ{}
3. display FEC creation:
Outer: global, this: user, LE: [{arguments: obj}], TDZ{}
4. display FEC execution:
Outer: global, this: user, LE: [{arguments: obj}], TDZ{}
5. show FEC creation:
Outer: global, this: Window, LE: [{arguments: obj, msg: "hi"}], TDZ{}
6. show FEC execution:
Outer: global, this: Window, LE: [{arguments: obj, msg: "hi"}], TDZ{}
7. show FEC creation:
Outer: global, this: user, LE: [{arguments: obj, msg: "hello"}], TDZ{}
8. show FEC execution:
Outer: global, this: user, LE: [{arguments: obj, msg: "hello"}], TDZ{}
```

Problem 2: Fix using wrapper, bind, call, or apply.

Original Code

```
"use strict";
```

```
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 log in`);
    },
};

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

Fixed Code

```
"use strict";

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 log in`);
    },
};

askPassword(user.loginOk.bind(user), user.loginFail.bind(user));
```

Problem 3: Fix using wrapper, bind, call, or apply.

Original Code

```
"use strict";
let group = {
    title: "Our Group",
    students: ["John", "Pete", "Alice"],
    showList: function () {
        this.students.forEach(function (student) {
            console.log(this.title + ": " + student);
        });
    },
};
```

```
group.showList();
```

Fixed Code

```
"use strict";
let group = {
    title: "Our Group",
    students: ["John", "Pete", "Alice"],
    showList: function () {
        this.students.forEach(function (student) {
            console.log(this.title + ": " + student);
        }.bind(this));
    },
};
group.showList();
```