

File Edit Selection View Go Run Terminal Help ← → Q Task 5

EXPLORER OPEN EDITORS JS Hierarchy.js JS Prototypejs JS Hierarchy.js > user > showInfo

```
1  class user { //Parent class
2      #name //private field with getter
3      #password //private no getter
4      #pin //private no getter
5      constructor(name, email, password, pin) { //constructor used to initialize object properties
6          this.name = name;
7          this.email = email; //public property name
8          this.#password = password; //private property
9          this.#pin = pin;
10     }
11     showInfo() { //method show only safe and public info
12         console.log("user name:", this.name);
13         console.log("user email:", this.email);
14     }
15     get password() {
16         return this.#password; //controlled access
17     }
18     set password(newPassword) {
19         this.#password = newPassword //update password
20     }
21     showPinInsideClass() { //method to acces pin only same class
22         console.log("user pin:", this.#pin);
23     }
24     checkPassword(inputPassword) { //method to safety check password
25         return this.#password === inputPassword; //we don't show or print password directly
26     }
27 }
28
29
30 class admin extends user { //child class inherits from user
31     constructor(name, email, password, pin, role) { //admin has name,email, and extra own property is role
32     }
33 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\User\Desktop\Tasks Training\Task 5> node .\Hierarchy.js  
user name: priya  
user email: priya@0001  
access password using getter: p155  
updated password: pc125  
user pin: 123  
user name: mrun  
user email: mrun@000  
user pin: 1234  
admin role: super admin  
false

SUGGESTED ACTIONS Build Workspace Show Config

Activate Windows and your code Go to settings > Activate Windows

Type here to search 627 PM 19°C Partly cloudy 12/30/2025

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar showing files like 'JS Hierarchy.js' and 'JS Prototype.js'. The main area displays two code files side-by-side:

```
JS Hierarchy.js | JS Prototype.js
```

**JS Hierarchy.js:**

```
29 class admin extends user { //child class inherits from user
30   constructor(name,email,password,pin,role) { //admin has name,email, and extra own property is role
31     super(name, email,password,pin) //calls constructor of parent class user
32     this.role = role;
33   }
34   showAdminInfo() {
35     this.showInfo();
36     this.showInInsideClass();
37     console.log("admin role:", this.role);
38   }
39 }
40 }

41 const user1 = new user("priya","priya@2001","p155",123); //create object for user class
42 user1.showInfo(); //only show name and email
43 console.log("access password using getter:", user1.password);
44 user1.password = "pc125";
45 console.log("updated password:", user1.password);
46 user1.showInInsideClass(); //only access inside same class
47

48 const admin1 = new admin("mrun","mrung@2000","admin1",1234,"super admin"); //create object for admin class
49 admin1.showAdminInfo(); //only show name,email and role
50 console.log(user1.checkPassword("1255"));
51

52 // console.log(user1.#password);//cannot access private fields give error
53
```

**JS Prototype.js:**

```
JS Hierarchy.js > ...
```

The terminal below shows the execution of the code:

```
PS C:\Users\User\Desktop\Tasks Training\Task 5> node .\Hierarchy.js
user name: priya
user email: priya@2001
access password using getter: p155
updated password: pc125
user pin: 123
user name: mrun
user email: mrung@2000
user pin: 1234
admin role: super admin
false
PS C:\Users\User\Desktop\Tasks Training\Task 5>
```

A sidebar on the right provides AI integration options and a suggestion for activating Windows 10.

The screenshot shows a code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Task 5.
- Sidebar:** Explorer, Open Editors (JS Hierarchy.js, JS Prototype.js), Task 5.
- Code Editor:** JS Prototype.js (selected). The code defines a User constructor function and a prototype for greet, isAdult, and updateEmail methods.
- Terminal:** PS C:\Users\User\Desktop\Tasks Training\Task 5> node .\Hierarchy.js. The output shows the creation of user objects and their method executions.
- Suggestions:** Ask about your code, Build Workspace, Show Config.
- System Tray:** Windows logo, search bar, taskbar icons (File Explorer, Edge, File Manager, etc.), system status (19°C, partly cloudy).

```
function User(name, email, age) { //constructor functions are used to create multiple objects with similar properties
    this.name = name;
    this.email = email;
    this.age = age;
}

User.prototype.greet = function() { //prototype-based utility function:greet 1st function
    console.log(`Hello ${this.name}`);
}

User.prototype.isAdult = function() { //check if user is adult 2nd function prototype
    if(this.age >= 18) {
        console.log(`${this.name} is an adult`);
    } else{
        console.log(`${this.name} is not an adult`);
    }
}

User.prototype.updateEmail = function(newEmail) { //3rd function prototype
    this.email = newEmail;
    console.log(`${this.name} is new mail is ${this.email}`);
}

const user1 = new User("priya", "priya@gmail.com", 24) //create user object
const user2 = new User("neha", "neha@gmail.com", 15);

user1.greet();
user1.isAdult();
user1.updateEmail("priya01@gmail.com");
```

```
PS C:\Users\User\Desktop\Tasks Training\Task 5> node .\Hierarchy.js
user name: priya
user email: priya@gmail.com
access password using getter: p155
updated password: pc125
user pin: 123
user name: mrun
user email: mrun@gmail.com
user pin: 1234
admin role: super admin
false
```

The screenshot shows the Visual Studio Code (VS Code) interface. The top bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar labeled "Task 5". The left sidebar has sections for Explorer, Open Editors, Task 5, and Outline/Timeline. The main editor area displays two files: "JS Hierarchy.js" and "JS Prototype.js". The "JS Prototype.js" file is currently active and contains the following code:

```
1  function User(name, email, age) { //constructor functions are used to create multiple objects with similar properties
2
3
4
5
6
7
8
9
10    User.prototype.isAdult = function() { //check if user is adult 2nd functin prototype
11      if(this.age >= 18) {
12        console.log(`${this.name} is an adult`);
13      }
14      else{
15        console.log(`${this.name} is not an adult`);
16      }
17    }
18
19
20    User.prototype.updateEmail = function(newEmail) { //3rdfunction prototype
21      this.email = newEmail;
22      console.log(`${this.name} is new mail is ${this.email}`);
23    }
24
25
26    const user1 = new User("priya", "priya@gmail.com", 24) //create user object
27    const user2 = new User("neha", "neha@gmail.com",15);
28
29    user1.greet();
30    user1.isAdult();
31    user1.updateEmail("priye01@gmail.com");
32
33    user2.greet();
34    user2.isAdult();
35    user2.updateEmail("priye01@gmail.com");

```

The bottom right corner features an AI integration panel with a "Ask about your code" button and a "Generate Agent Instructions" link. The bottom status bar shows the terminal output, system information like CPU and RAM usage, and a weather forecast for 19°C Partly cloudy.