

```
}
```

# Employee Inheritance:

```
'use strict';
```

```
const fs = require('fs');
```

```
process.stdin.resume();
```

```
process.stdin.setEncoding("ascii");
```

```
let inputString = "";
```

```
let currentLine = 0;
```

```
process.stdin.on("data", function (chunk) {
```

```
    inputString += chunk;
```

```
});
```

```
process.stdin.on("end", function () {
```

```
    inputString = inputString.split('\n');
```

```
    main();
```

```
});
```

```
function readLine() {
```

```
    return inputString[currentLine++];
```

```
}
```

```
function Employee(title) {
```

```
    this.title = title;
```

```
    this.setTitle = function(tE){
```

```
        this.title = tE;
```

```
}

this.getTitle = function() {

    return this.title;

}

}

Employee.prototype.setTitle = function(tE){

    this.title = tE;

}
```

```
Employee.prototype.getTitle = function(tE){

    return this.title;

}
```

```
function Engineer(title, isManager) {

    Employee.call(this, title);

    this.isManager = isManager;

    this.setIsManager = function(isM){

        this.isManager = isM;

    }

    this.getIsManager = function() {

        return this.isManager;

    }

}
```

```
Engineer.prototype.setIsManager = function(tE){

    this.isManager = tE;

}
```

```
Engineer.prototype.getIsManager = function() {  
  
    return this.isManager;  
  
}
```

```
function main() {  
  
    const ws = fs.createWriteStream(process.env.OUTPUT_PATH);  
  
    var inputs = readLine().split(' ');  
  
    var engineerObject = new Engineer(inputs[0], inputs[1].toLowerCase() === 'true');  
  
    ws.write(`Initial Employee Profile - Title is ${engineerObject.getTitle()}. ${engineerObject.getIsManager() ? 'Is' : 'Is not'} a Manager\n`)  
  
    engineerObject.setTitle(readLine());  
  
    engineerObject.setIsManager(readLine().toLowerCase() === 'true');  
  
    ws.write(`Final Employee Profile - Title is ${engineerObject.getTitle()}. ${engineerObject.getIsManager() ? 'Is' : 'Is not'} a Manager\n`)  
  
    ws.write(`Engineer.prototype has property setTitle: ${Engineer.prototype.hasOwnProperty('setTitle')}\n`);  
  
    ws.write(`Engineer.prototype has property getTitle: ${Engineer.prototype.hasOwnProperty('getTitle')}\n`);  
  
    ws.write(`Engineer.prototype has property setIsManager: ${Engineer.prototype.hasOwnProperty('setIsManager')}\n`);  
  
    ws.write(`Engineer.prototype has property getIsManager: ${Engineer.prototype.hasOwnProperty('getIsManager')}\n`);  
  
}
```