

Assignment: Object Cloning in JavaScript

```
JS address.js > ...
1  const putri = {
2    name: "Putri",
3    birthday: "2005-02-13",
4    phone: "081234567890",
5    address: {
6      primary: "Bandung",
7      secondary: "Sukabumi"
8    }
9  };
10
11  // Shallow copy using Object.assign
12  const bayu = Object.assign({}, putri);
13  bayu.name = "Bayu";
14  bayu.address.primary = "Aceh";
15
16  console.log(putri.name);
17  console.log(putri.address.primary);
18
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Nusaputra\Second Yr\Third Sem\Platform Based on Programming\4> node address.js
Putri
Aceh
PS D:\Nusaputra\Second Yr\Third Sem\Platform Based on Programming\4> 
```

Explanation:

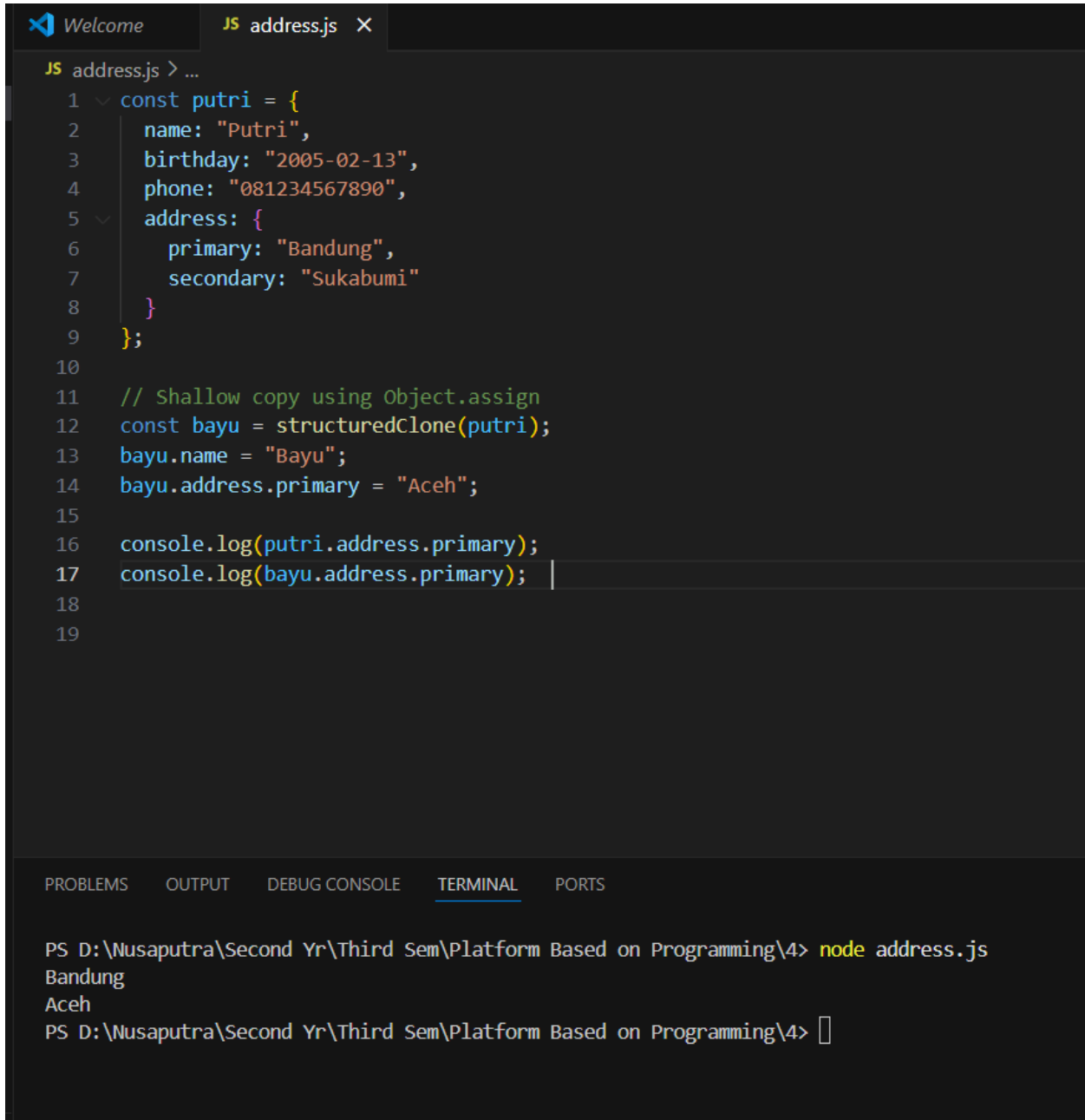
When we use `Object.assign({}, putri)`, it makes a **shallow copy**.

This means that the main object is copied, but the **nested object** (`address`) is still **shared** between `putri` and `bayu`.

So when we change `bayu.address.primary`, it also changes `putri.address.primary`.

How to Fix It (Deep Copy):

If we want **putri** to stay “Bandung” and **bayu** to be “Aceh”, we can use this:



```
JS address.js > ...
1  const putri = {
2    name: "Putri",
3    birthday: "2005-02-13",
4    phone: "081234567890",
5    address: {
6      primary: "Bandung",
7      secondary: "Sukabumi"
8    }
9  };
10
11  // Shallow copy using Object.assign
12  const bayu = structuredClone(putri);
13  bayu.name = "Bayu";
14  bayu.address.primary = "Aceh";
15
16  console.log(putri.address.primary);
17  console.log(bayu.address.primary);
18
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Nusaputra\Second Yr\Third Sem\Platform Based on Programming\4> node address.js
Bandung
Aceh
PS D:\Nusaputra\Second Yr\Third Sem\Platform Based on Programming\4> 
```

Conclusion:

- `Object.assign()` only copies the top level (shallow copy).
- `structuredClone()` copies everything (deep copy), so the objects don't affect each other.