

# Design Patterns

## WA1

**Topic:** - Implement Adapter Design Pattern.

**Description:** - An Adapter Pattern says that just "converts the interface of a class into another interface that a client wants".

**Code :** -

```
interface iPhone {
    useLightning() : void
}

interface Android {
    useTypeC() : void
}

class iPhone14 implements iPhone {
    useLightning() : void {
        console.log(`Using Lightning Port.`)
    }
}

class SamsungA32 implements Android {
    useTypeC(): void {
        console.log(`Using Type C.`)
    }
}

class TypeCToLightning implements Android {
    iPhoneDevice : iPhone

    constructor(device : iPhone) {
```

```
        this.iPhoneDevice = device
    }

    useTypeC(): void {
        console.log(`Adapting Lightning from Type C.`)
        this.iPhoneDevice.useLightning()
    }
}

const iPhone14Max = new iPhone14()
const adapter = new TypeCToLightning(iPhone14Max)

adapter.useTypeC()
```

**Output : -**

```
Adapting Lightning from Type C.
Using Lightning Port.
```