

Experiment-9a:

AIM: On the page, display the price of the mobile-based in three different colors. Instead of using the number in our code, represent them by string values like GoldPlatinum, PinkGold, SilverTitanium.

Require: npm install -g typescript

Source Code:

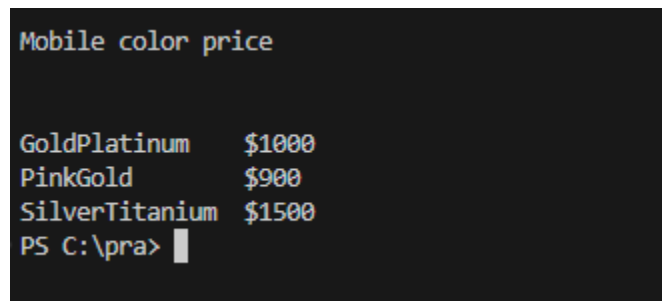
Ex9a.ts

```
const obj: { GoldPlatinum: string } = { GoldPlatinum: "$1000" };
const ob1: { PinkGold: string } = { PinkGold: "$900" };
const ob2: { SilverTitanium: string } = { SilverTitanium: "$1500" };
console.log("\nMobile color price\n");
console.log("\nGoldPlatinum\t" + obj.GoldPlatinum);
console.log("PinkGold\t" + ob1.PinkGold);
console.log("SilverTitanium\t" + ob2.SilverTitanium);
```

Output:

C:\pra> tsc ex9a.ts

C:\pra>node ex9a.js



```
Mobile color price

GoldPlatinum    $1000
PinkGold        $900
SilverTitanium  $1500
PS C:\pra>
```

Experiment-9b:

AIM: Define an arrow function inside the event handler to filter the product array with the selected product object using the productId received by the function. Pass the selected product object to the next screen.

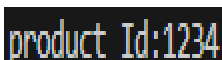
Source Code:

```
var getProductDetails=(productId : number):string=>{return "product Id:"+productId};
console.log(getProductDetails(1234));
```

Output:

C:\pra> tsc ex9b.ts

C:\pra>node ex9b.js



```
product Id:1234
```

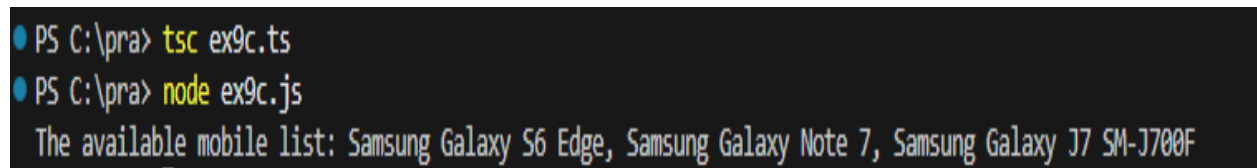
Experiment-9c:

AIM: Consider that developer needs to declare a function - getMobileByVendor which accepts string as input parameter and returns the list of mobiles.

Source Code:

```
function getMobileByManufacturer(manufacturer: string): string[] {  
    let mobileList: string[];  
    if (manufacturer === 'Samsung') {  
        mobileList = ['Samsung Galaxy S6 Edge', 'Samsung Galaxy Note 7', 'Samsung Galaxy J7 SM-J700F'];  
    } else if (manufacturer === 'Apple') {  
        mobileList = ['Apple iPhone 5s', 'Apple iPhone 6s', 'Apple iPhone 7'];  
    } else {  
        mobileList = ['Nokia 105', 'Nokia 230 Dual Sim'];  
    }  
    return mobileList;  
}  
  
console.log('The available mobile list: ' + getMobileByManufacturer('Samsung').join(', '));
```

Output:



```
PS C:\pra> tsc ex9c.ts  
PS C:\pra> node ex9c.js  
The available mobile list: Samsung Galaxy S6 Edge, Samsung Galaxy Note 7, Samsung Galaxy J7 SM-J700F
```

Experiment-9d:

AIM: Consider that developer needs to declare a manufacturer's array holding 4 objects with id and price as a parameter and needs to implement an arrow function - myfunction to populate the id parameter of manufacturers array whose price is greater than or equaling

Source Code:

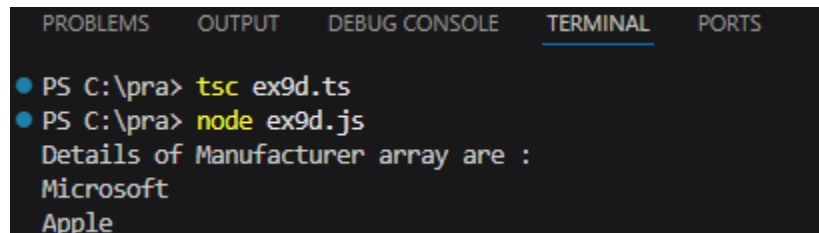
```
const manufacturers = [{ id:'Samsung', price:150 },  
    { id:'Microsoft', price:200 },  
    { id:'Apple', price:400 },  
    { id:'Micromax', price:100 }  
];  
  
let test;  
  
// Arrow function to populate the details of Array whose price is greater than 200  
function myFunction() {  
    test = manufacturers.filter((manufacturer) =>manufacturer.price>= 200);
```

```

}
// self-invoking an arrow function
myFunction();
console.log('Details of Manufacturer array are : ');
// logic to populate the manufacturer array details based on id value
for (const item of test) {
    console.log(item.id);
}

```

Output:



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\pra> tsc ex9d.ts
PS C:\pra> node ex9d.js
Details of Manufacturer array are :
Microsoft
Apple

```

Experiment-9e:

AIM: Declare a function - getMobileByManufacturer with two parameters namely manufacturer and id, where manufacturer value should be passed as Samsung and id parameter should be optional while invoking the function, if id is passed as 101 then this function should .

Source Code:

```

function getMobileByManufacturer(manufacturer: string = 'Samsung', id?: number): string[] {
    let mobileList: string[];

    // Logic to be evaluated if id parameter is provided while invoking the function
    if (id) {
        if (id === 101) {
            return ['Moto G Play, 4th Gen', 'Moto Z Play with Style Mod'];
        }
    }

    // Logic to return mobileList based on manufacturer category
    if (manufacturer === 'Samsung') {
        mobileList = ['Samsung Galaxy S6 Edge', 'Samsung Galaxy Note 7', 'Samsung Galaxy J7 SM-J700F'];
    } else if (manufacturer === 'Apple') {
        mobileList = ['Apple iPhone 5s', 'Apple iPhone 6s', 'Apple iPhone 7'];
    } else {

```

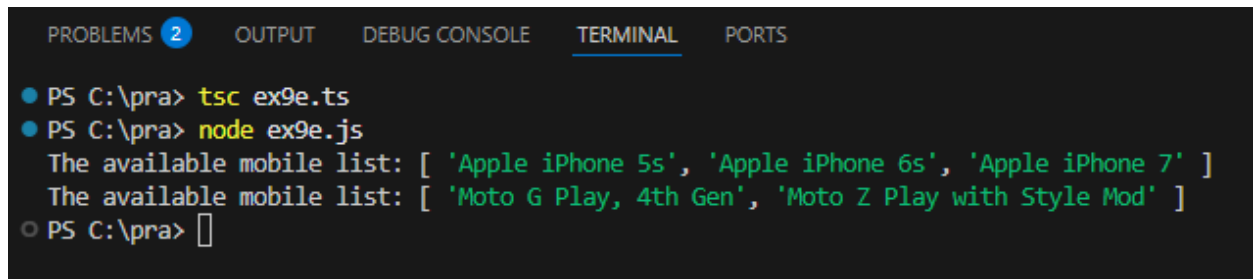
```
        mobileList = ['Nokia 105', 'Nokia 230 Dual Sim'];
    }

    return mobileList;
}

// Statement to invoke function with optional parameter
console.log('The available mobile list:', getMobileByManufacturer('Apple'));

// Statement to invoke function with default parameter
console.log('The available mobile list:', getMobileByManufacturer(undefined, 101));
```

Output:



The screenshot shows a VS Code terminal window with the 'TERMINAL' tab selected. The terminal output is as follows:

```
PS C:\pra> tsc ex9e.ts
PS C:\pra> node ex9e.js
The available mobile list: [ 'Apple iPhone 5s', 'Apple iPhone 6s', 'Apple iPhone 7' ]
The available mobile list: [ 'Moto G Play, 4th Gen', 'Moto Z Play with Style Mod' ]
PS C:\pra> 
```