Experinment-11a:

AIM: Declare a productList interface which extends properties from two other declared interfaces like Category,Product as well as implementation to create a variable of this interface type..

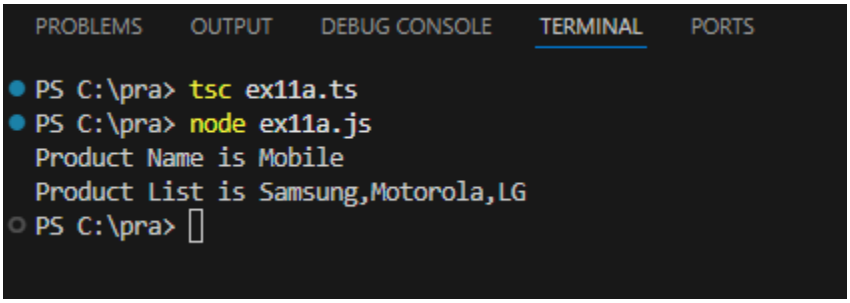
Require: npm install -g typescript

Source Code:

Ex11a.ts

```
// declaring a Category interface
interface Category {
   categoryName: string;
   }
// declaring a Product interface
interface Product {
      productName: string;
      productId: number;
   }
// declaring a ProductList interface which is extends from Category and Product interfaces
interface ProductList extends Category, Product {
      list: Array<string>;
   }
// declaring a variable which is type of ProductList interface
const productDetails: ProductList = {
      categoryName:'Gadget',
      productName:'Mobile',
      productId:1234,
      list: ['Samsung', 'Motorola', 'LG']
   };
// assigning list value of productDetails variable into another variable
const listProduct = productDetails.list;
// assigning productName value of productDetails variable into another variable
const pname: string = productDetails.productName;
// line to populate Product name
console.log('Product Name is ' + pname);
// line to populate Product list
console.log('Product List is ' + listProduct);
```

Output:



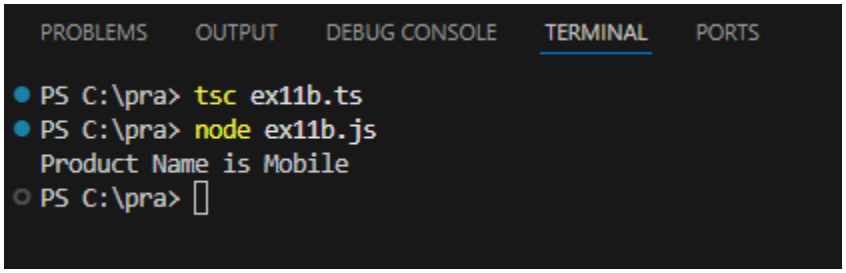
Experinment-11b:

AIM: Consider the Mobile Cart application, Create objects of the Product class and place them into the productlist array.

Source Code:

```
// Declaring a Product interface
interface Product {
    displayProductName: (productId: number) => string;
    getProductDetails(): string[];
}
// Declaring Gadget class which implements Product interface
class Gadget implements Product {
    getProductDetails(): string[] {
        return ['Samsung', 'LG', 'Moto'];
    }
    displayProductName(productId: number): string {
        if (productId === 101) {
            return 'Product Name is Mobile';
        } else if (productId === 201) {
            return 'Product Name is Tablet';
        }
        return 'Product not found'; // Added default return statement
    }
    getGadget(): string[] {
        return ['Mobile', 'Tablet', 'iPad', 'iPod'];
    }
}
// Creating instance of class of interface type
const gadget: Product = new Gadget();
// Creating variable to hold return value of displayProductName function
const productName: string = gadget.displayProductName(101);
// Line to populate Product name on console
console.log(productName);
```

Output:



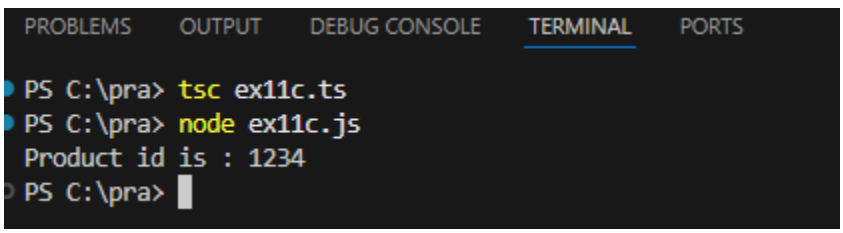
Experinment-11c:

AIM: Declare a class named - Product with the below-mentioned declarations: (i) productId as number property (ii) Constructor to initialize this value (iii) getProductId method to return the message "Product id is <<id value>>".

Source Code:

```
// declaring a Product class
class Product {
    staticproductPrice: string;
    productId: number;
// constructor declaration
    constructor(productId: number) {
        this.productId = productId;
    }
    getProductId(): string {
        return'Product id is : ' + this.productId;
    }
}
// creation of Product class object
const product: Product = new Product(1234);
// line to populate the product id details
console.log(product.getProductId());
```

Output:



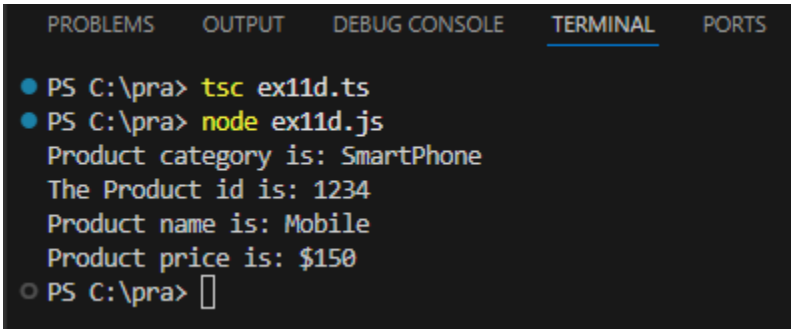
Experinment-11d:

AIM: Create a Product class with 4 properties namely productId, productName, productPrice, productCategory with private, public, static, and protected access modifiers and accessing them through Gadget class and its methods.

Source Code:

```
// Declaring a Product class with access modifiers
class Product {
    static productPrice = 150;
    private productId: number;
    public productName: string;
    protected productCategory: string;
    // Constructor with type annotations
    constructor(productId: number, productName: string, productCategory: string) {
        this.productId = productId;
        this.productName = productName;
        this.productCategory = productCategory;
    }
    // Method to display product id details
    getProductId(): void {
        console.log('The Product id is: ' + this.productId);
    }
}
// Declaring a Gadget class extending the properties from Product class
class Gadget extends Product {
    // Method to display productCategory property
    getProduct(): void {
        console.log('Product category is: ' + this.productCategory);
    }
}
// Gadget Class object creation
const g: Gadget = new Gadget(1234, 'Mobile', 'SmartPhone');
// Invoking getProduct method with the help of Gadget object
g.getProduct();
// Invoking getProductId method with the help of Gadget object
g.getProductId();
// Line to populate product name property with Gadget object
console.log('Product name is: ' + g.productName);
// Line to populate static property product price directly with Product class name
console.log('Product price is: $' + Product.productPrice);
```

Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\pra> tsc ex11d.ts
PS C:\pra> node ex11d.js
Product category is: SmartPhone
The Product id is: 1234
Product name is: Mobile
Product price is: $150
PS C:\pra>