1)

interface Producto {

    Nombre:string;

    Precio:number;

    Oferta?:boolean;

}

const productos : Array <Producto> = [ {"Nombre":"Camisa", "Precio":60,},

                    {"Nombre":"Pantalones", "Precio":45, "Oferta":true},

                    {"Nombre":"Polo", "Precio":35,},

                    {"Nombre":"Slip", "Precio":15, "Oferta":true} ];

productos.forEach( function( pro:Producto ) : void {

    if(pro.Oferta == undefined) {

        console.log(`${pro.Nombre} de ${pro.Precio}€.`);

    }

} );

productos.sort( function (a:Producto, b:Producto) : any {

    if (a.Precio < b.Precio) {

        return 1;

    } else if (a.Precio > b.Precio) {

        return -1;

    } else {

        return 0;

    }

} )

console.log(productos);

2)

class Persona{

    nombre:string;

    edad:number;

    dni:string;

    domicilio:string;

    constructor(n?:string, e?:number, dni?:string, d?:string){

        this.nombre= n ?? "Xarusovic Gebczyk";

        this.edad= e ?? 25;

        this.dni= dni ??"12345678B";

        this.domicilio= d ?? "Avenida Elcumdeaglo";

    }

    private getNombre() { return this.nombre; }

    private getEdad() { return this.edad; }

    private getDni() { return this.dni; }

    private getDomicilio() { return this.domicilio; }

    private setNombre(n:string) { this.nombre=n; }

    private setEdad(e:number) { this.edad=e; }

    private setDni(dni:string) { this.dni=dni; }

    private setDomicilio(d:string) { this.domicilio=d; }

    public mostrarInfo() {

        console.log(`${this.getNombre()} con ${this.getEdad()} años de edad y DNI ${this.getDni()}, vive en la calle "${this.getDomicilio()}".`);

    }

}

let persona1 = new Persona();

persona1.mostrarInfo();

let persona2 = new Persona("Adajio Lameca", 25, "12345678A", "Calle la Palma");

persona2.mostrarInfo();

3)

class Cafetera{

    capacidadMaxima:number;

    cantidadActual:number;

    constructor(cm?:number, ca?:number) {

        this.capacidadMaxima=cm ?? 1000;

        this.cantidadActual=ca ?? 0;

    }

    private getCapacidadMaxima() { return this.capacidadMaxima; }

    private getCantidadActual() { return this.cantidadActual; }

    private setCapacidadMaxima(cm:number) { this.capacidadMaxima=cm; }

    private setCantidadActual(ca:number) { this.cantidadActual=ca; }

    public llenarCafetera() { this.cantidadActual=this.capacidadMaxima; }

    public servirTaza(ct:number) {

        if(ct > this.getCantidadActual()) {

            console.log(`No hay suficiente café, se ha servido una taza con solo ${this.getCantidadActual()}ml.`);

            this.setCantidadActual(0);

        } else {

            console.log(`Se ha llenado la taza`);

            this.setCantidadActual(this.getCantidadActual()-ct);

        }

    }

    public vaciarCafetera() {this.setCapacidadMaxima(0); console.log(`Se ha vaciado la cafetera.`);}

    public agregarCafe(c:number) {this.setCantidadActual(this.getCantidadActual()+c); console.log(`Se ha llenado la cafetera con ${c}ml.`);}

}

let cafetera1 = new Cafetera();

cafetera1.llenarCafetera();

cafetera1.servirTaza(250);

cafetera1.vaciarCafetera();

cafetera1.agregarCafe(500);

4)

class Producto {

    nombre:string;

    lote:number;

    constructor(n?:string, l?:number){

        this.nombre=n ?? "Sin nombre";

        this.lote=l ?? 999;

    }

}

class Fresco extends Producto {

    fechaEnvasado:string;

    paisOrigen:string;

    constructor(n:string, l:number, fe?:string, po?:string) {

        super(n, l);

        this.fechaEnvasado=fe ?? "2022/02/02";

        this.paisOrigen=po ?? "Españita";

    }

}