How to associate call with interface

To use interface we have to use implements worlds

interface Bulb

{

wattage:number;

setWattage(wattage:number):void;

getWattage():number;

}

// following wont compile.

class AABulb implements Bulb

{

}

/\*

Type 'AABulb' is missing the following properties from type 'Bulb': wattage, setWattage, getWattage

\*/

If we implements any interface then we have to deine its all properties in class.

interface Bulb

{

wattage:number;

setWattage(wattage:number):void;

getWattage():number;

}

function eg()

{

let b:Bulb;

b = new Bulb(); // incorrect interface cannot instantiated.

}

/\*

'Bulb' only refers to a type, but is being used as a value here.

we can not make object of an interface

\*/

Interface can not be instantiated.

So proper code will be

interface Bulb

{

wattage:number;

setWattage(wattage:number):void;

getWattage():number;

}

class AABulb implements Bulb

{

wattage:number;

constructor(){

this.wattage = 0

}

setWattage(wattage:number)

{

this.wattage = wattage;

}

getWattage():number{

return this.wattage;

}

}

function eg3(){

let b:Bulb;

b = new AABulb();

b.setWattage(60);

console.log(b.getWattage());

}

eg3();

If we create a variable with type Bulb and in the AABulb class if we increase the property Dynamcally.

So will it be accessible?

interface Bulb

{

wattage:number;

setWattage(wattage:number):void;

getWattage():number;

}

class AABulb implements Bulb

{

wattage:number;

constructor(){

this.wattage = 0

}

setWattage(wattage:number)

{

this.wattage = wattage;

}

getWattage():number{

return this.wattage;

}

printBrand():void

{

console.log("Philips");

}

}

function eg4(){

let b:Bulb;

b = new AABulb();

b.setWattage(60);

console.log(b.getWattage());

b.printBrand(); // incorrect

}

eg4();

/\*

error TS2339: Property 'printBrand' does not exist on type 'Bulb'.

\*/

Error: printBrand does not exits on type Bulb.

What id we do not implement Bulb to class AABulb and crease same instance.

interface Bulb

{

wattage:number;

setWattage(wattage:number):void;

getWattage():number;

}

class AABulb

{

wattage:number;

constructor(){

this.wattage = 0

}

setWattage(wattage:number)

{

this.wattage = wattage;

}

getWattage():number{

return this.wattage;

}

printBrand():void

{

console.log("Philips");

}

}

function eg5(){

let b:Bulb;

b = new AABulb();

b.setWattage(60);

console.log(b.getWattage());

}

eg5();

Working correctly

But

interface Bulb

{

wattage:number;

setWattage(wattage:number):void;

getWattage():number;

}

class AABulb

{

wattage:number;

constructor(){

this.wattage = 0

}

setWattage(wattage:number)

{

this.wattage = wattage;

}

}

function eg6(){

let b:Bulb;

b = new AABulb();

b.setWattage(60);

}

eg6();

/\*

Property 'getWattage' is missing in type 'AABulb' but required in type 'Bulb'.

\*/

So if you are implmnent class or not,

If your object type is of interface and you have mention all the methods written in terface then it will compile successfully. As long as you are following the guideline it will Work.

Assignment.

Create examples

Class aaa implements bbb,ccc // example of implementing multiple interfaces

Class aaa extends bbb // inheritance

Class aaa extends bbb,ccc // class extends more than one class

Can base class pointer stores address of an object created from its derived class

Can a derived call pointer store address of an object created from its base class

—------—------—------—------—------

Aaa interface has a method declatraion

Bbb interface a method decldation with same name as in aa interface but different parameters (signature)

What will happen

Class ccc implements aaa,bbb