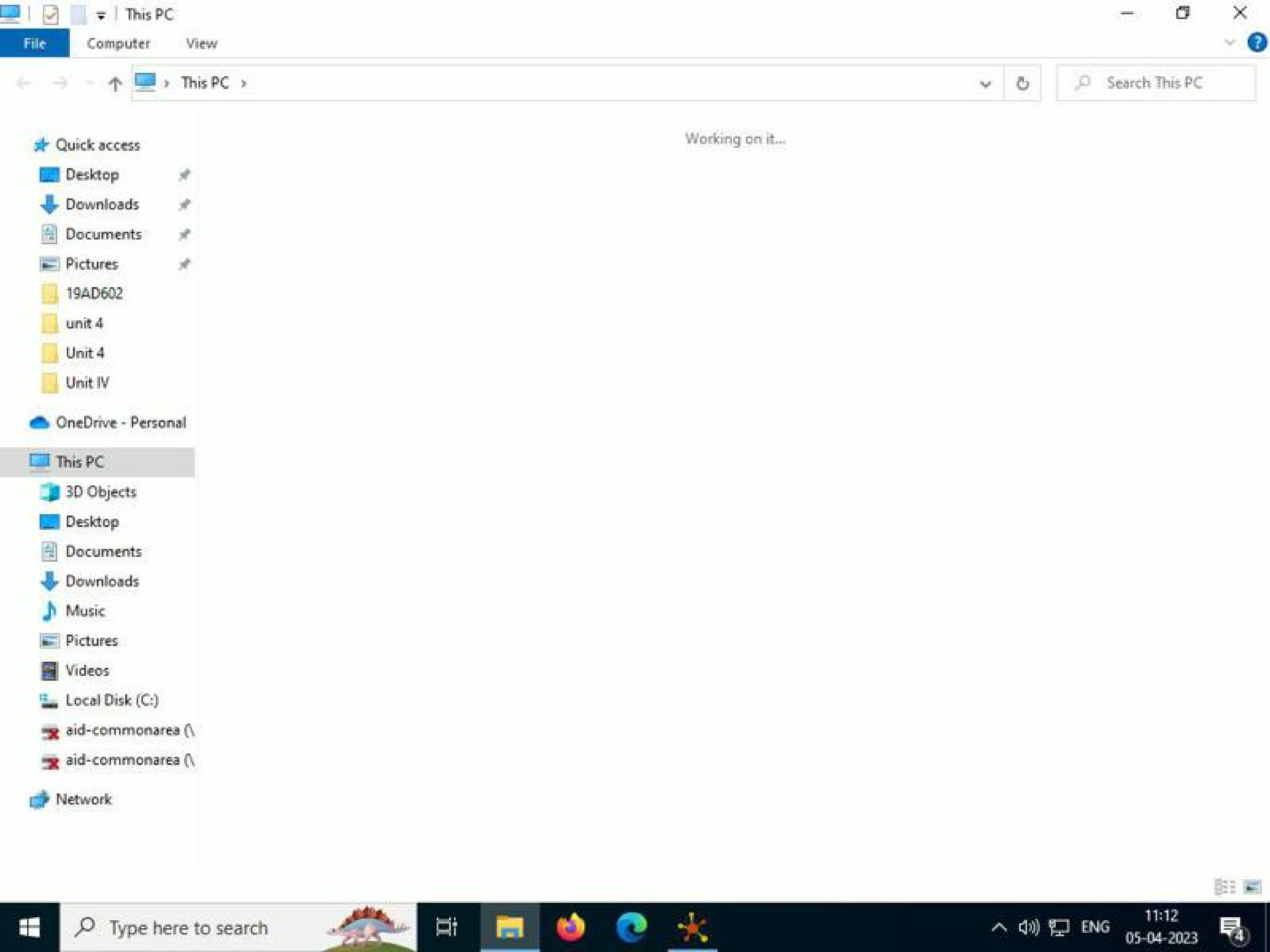
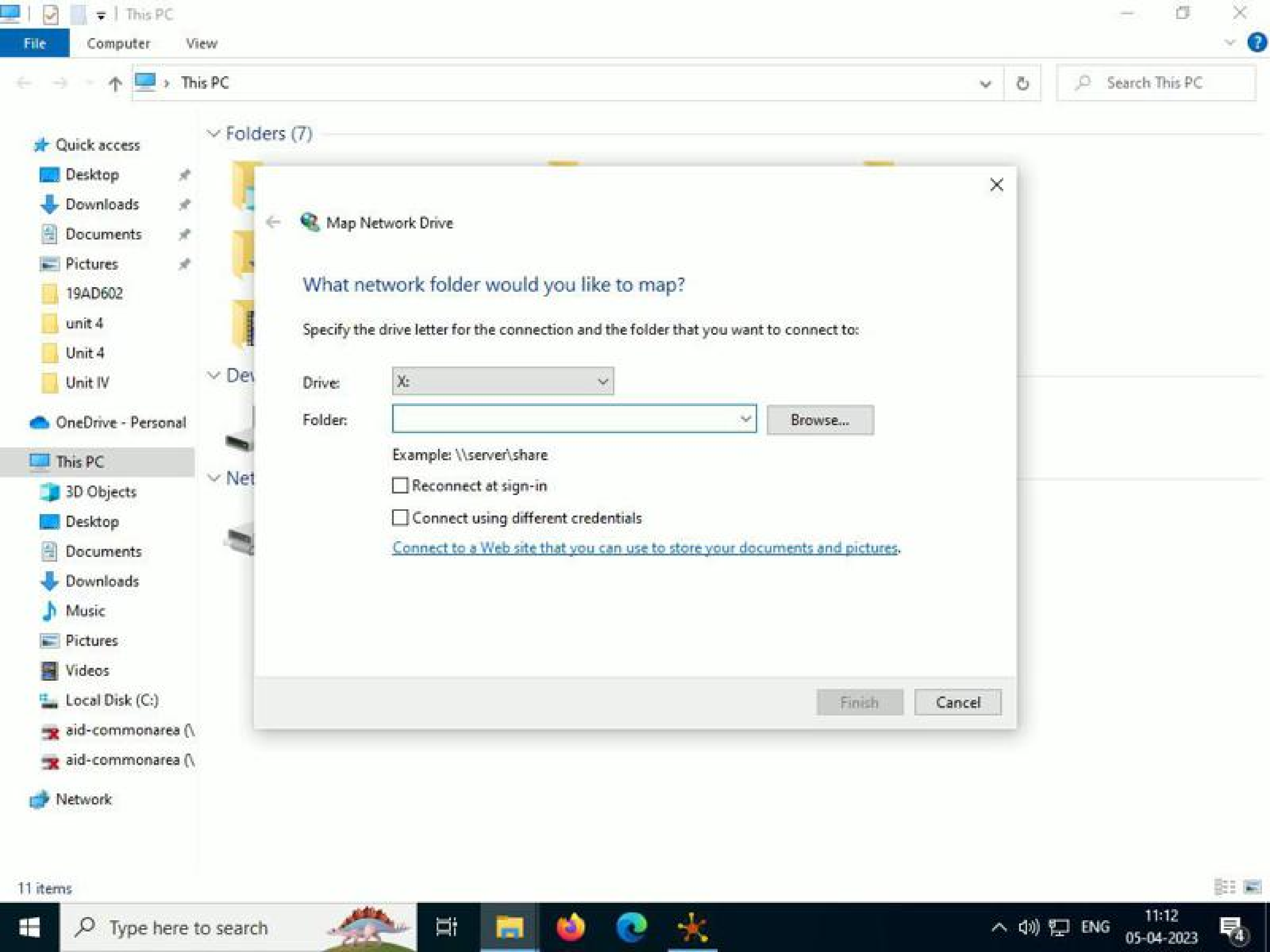

This PC
File
Computer
View
This PC
Search This PC
Working on it...
Quick access
Desktop
Downloads
Documents
Pictures
19AD602
unit 4
Unit 4
Unit IV
OneDrive - Personal
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
aid-commonarea (\
aid-commonarea (\
Network
Type here to search
ENG
11:12
05-04-2023



Map Network Drive

What network folder would you like to map?

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: X:

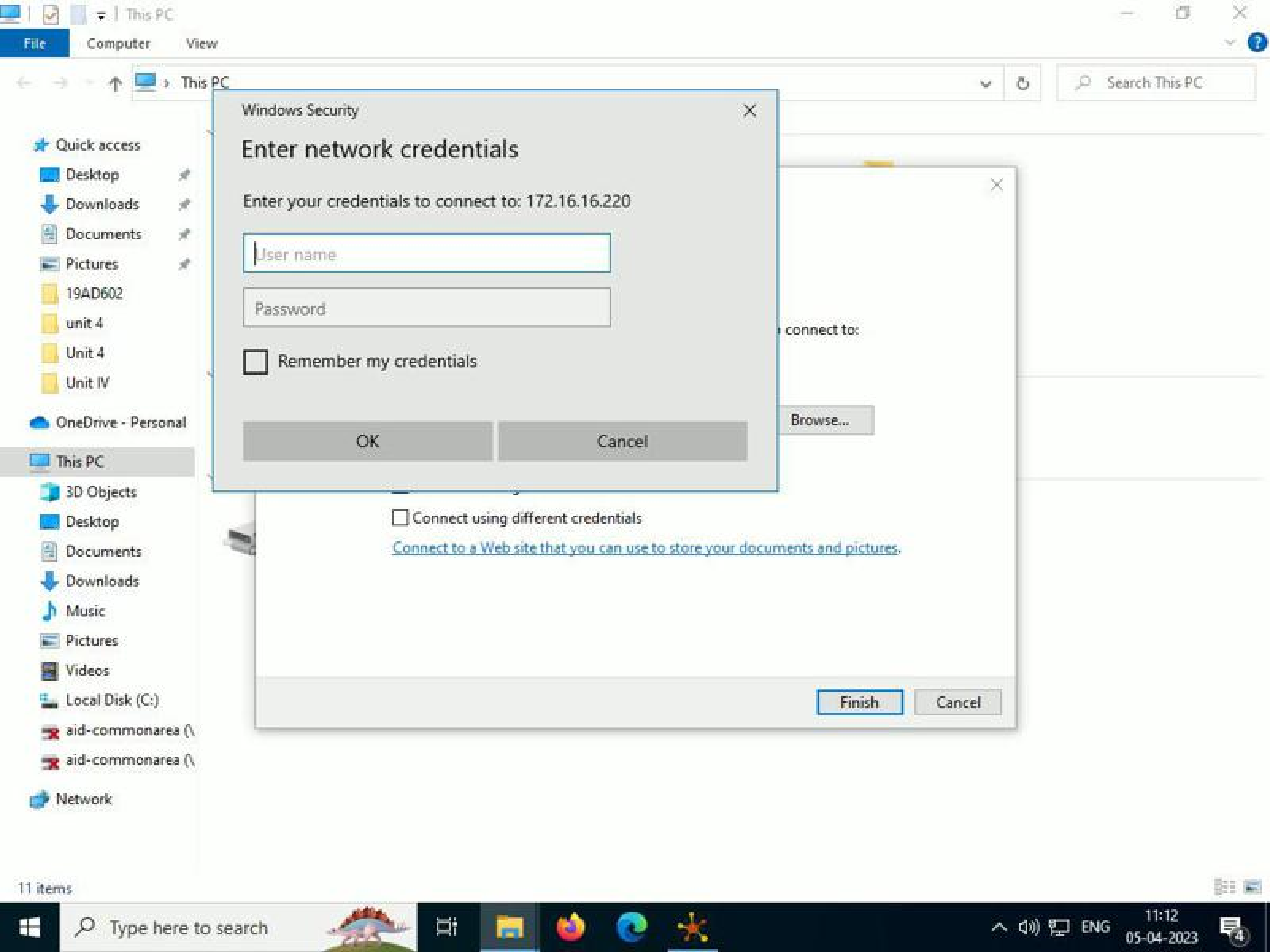
Folder:

Example: \\server\share

☐ Reconnect at sign-in

☐ Connect using different credentials

Connect to a Web site that you can use to store your documents and pictures.



Windows Security

Enter network credentials

Enter your credentials to connect to: 172.16.16.220

User name

Password

Remember my credentials

OK Cancel

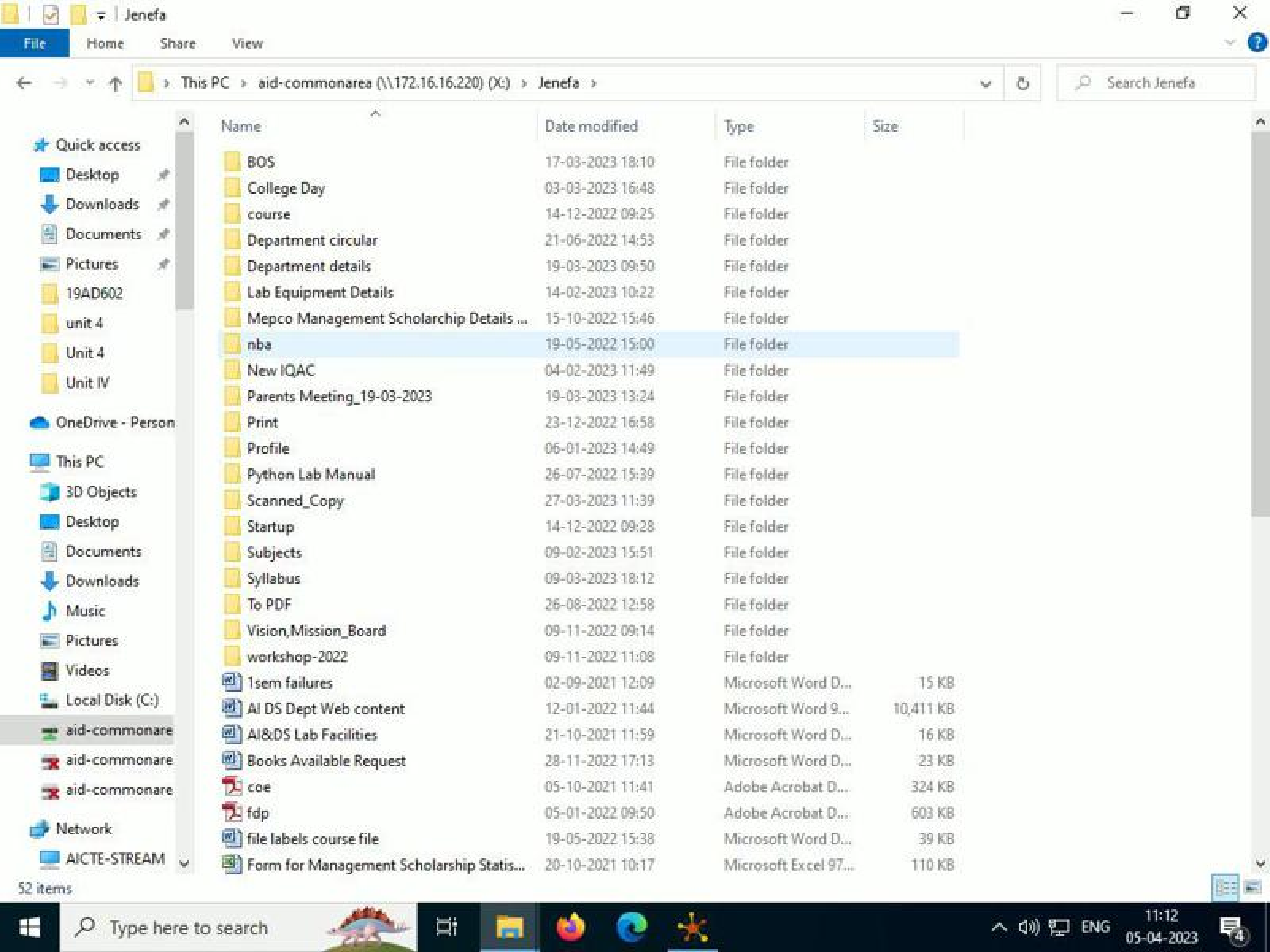
connect to:

Browse...

Connect using different credentials

Connect to a Web site that you can use to store your documents and pictures.

Finish Cancel



Jenefa

File Home Share View

This PC › aid-commonarea (\\172.16.16.220) (X:) › Jenefa ›

Search Jenefa

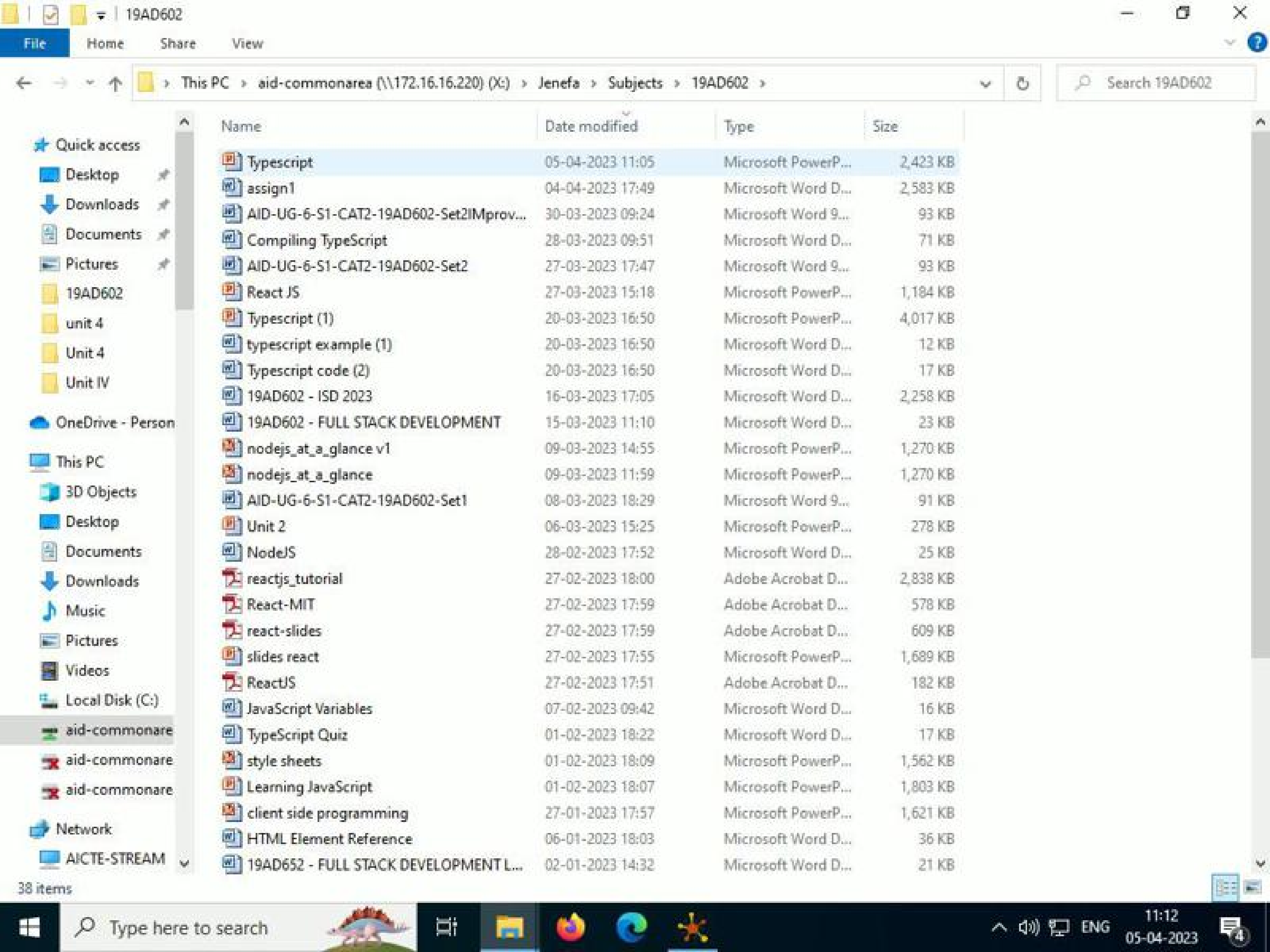
Quick access
Desktop
Downloads
Documents
Pictures
19AD602
unit 4
Unit 4
Unit IV
OneDrive - Person
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
aid-commonare
aid-commonare
aid-commonare
Network
AICTE-STREAM

Name	Date modified	Type	Size
BOS	17-03-2023 18:10	File folder	
College Day	03-03-2023 16:48	File folder	
course	14-12-2022 09:25	File folder	
Department circular	21-06-2022 14:53	File folder	
Department details	19-03-2023 09:50	File folder	
Lab Equipment Details	14-02-2023 10:22	File folder	
Mepco Management Scholarchip Details ...	15-10-2022 15:46	File folder	
nba	19-05-2022 15:00	File folder	
New IQAC	04-02-2023 11:49	File folder	
Parents Meeting_19-03-2023	19-03-2023 13:24	File folder	
Print	23-12-2022 16:58	File folder	
Profile	06-01-2023 14:49	File folder	
Python Lab Manual	26-07-2022 15:39	File folder	
Scanned_Copy	27-03-2023 11:39	File folder	
Startup	14-12-2022 09:28	File folder	
Subjects	09-02-2023 15:51	File folder	
Syllabus	09-03-2023 18:12	File folder	
To PDF	26-08-2022 12:58	File folder	
Vision,Mission_Board	09-11-2022 09:14	File folder	
workshop-2022	09-11-2022 11:08	File folder	
1sem failures	02-09-2021 12:09	Microsoft Word D...	15 KB
AI DS Dept Web content	12-01-2022 11:44	Microsoft Word 9...	10,411 KB
AI&DS Lab Facilities	21-10-2021 11:59	Microsoft Word D...	16 KB
Books Available Request	28-11-2022 17:13	Microsoft Word D...	23 KB
coe	05-10-2021 11:41	Adobe Acrobat D...	324 KB
fdp	05-01-2022 09:50	Adobe Acrobat D...	603 KB
file labels course file	19-05-2022 15:38	Microsoft Word D...	39 KB
Form for Management Scholarship Statis...	20-10-2021 10:17	Microsoft Excel 97...	110 KB

52 items

Type here to search

ENG 11:12 05-04-2023



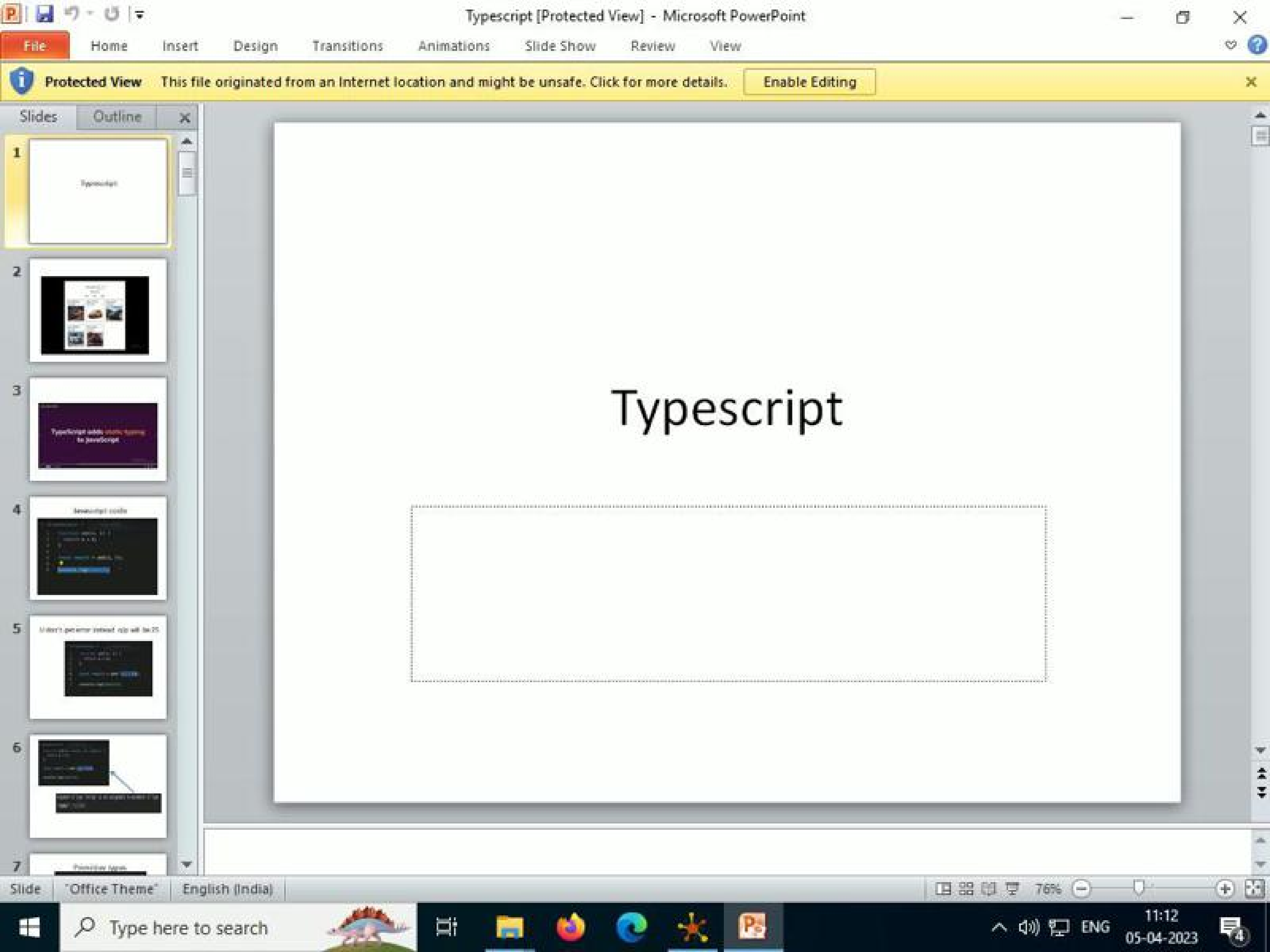
19AD602

File Home Share View

This PC > aid-commonarea (\\172.16.16.220) (X:) > Jenefa > Subjects > 19AD602 >

Search 19AD602

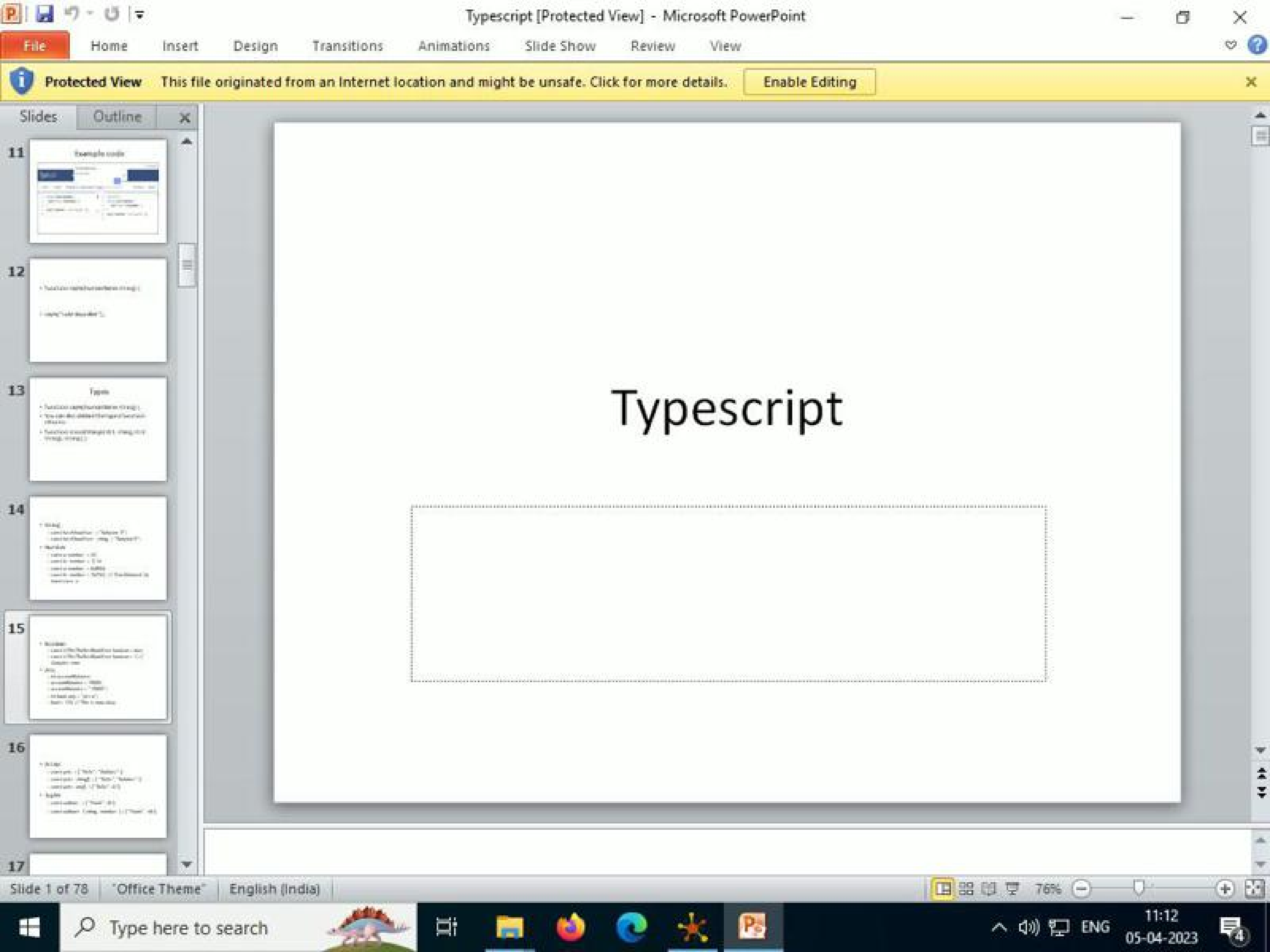
Name	Date modified	Type	Size
Typescript	05-04-2023 11:05	Microsoft PowerP...	2,423 KB
assign1	04-04-2023 17:49	Microsoft Word D...	2,583 KB
AID-UG-6-S1-CAT2-19AD602-Set2IMprov...	30-03-2023 09:24	Microsoft Word 9...	93 KB
Compiling TypeScript	28-03-2023 09:51	Microsoft Word D...	71 KB
AID-UG-6-S1-CAT2-19AD602-Set2	27-03-2023 17:47	Microsoft Word 9...	93 KB
React JS	27-03-2023 15:18	Microsoft PowerP...	1,184 KB
Typescript (1)	20-03-2023 16:50	Microsoft PowerP...	4,017 KB
typescript example (1)	20-03-2023 16:50	Microsoft Word D...	12 KB
Typescript code (2)	20-03-2023 16:50	Microsoft Word D...	17 KB
19AD602 - ISD 2023	16-03-2023 17:05	Microsoft Word D...	2,258 KB
19AD602 - FULL STACK DEVELOPMENT	15-03-2023 11:10	Microsoft Word D...	23 KB
nodejs_at_a_glance v1	09-03-2023 14:55	Microsoft PowerP...	1,270 KB
nodejs_at_a_glance	09-03-2023 11:59	Microsoft PowerP...	1,270 KB
AID-UG-6-S1-CAT2-19AD602-Set1	08-03-2023 18:29	Microsoft Word 9...	91 KB
Unit 2	06-03-2023 15:25	Microsoft PowerP...	278 KB
NodeJS	28-02-2023 17:52	Microsoft Word D...	25 KB
reactjs_tutorial	27-02-2023 18:00	Adobe Acrobat D...	2,838 KB
React-MIT	27-02-2023 17:59	Adobe Acrobat D...	578 KB
react-slides	27-02-2023 17:59	Adobe Acrobat D...	609 KB
slides react	27-02-2023 17:55	Microsoft PowerP...	1,689 KB
ReactJS	27-02-2023 17:51	Adobe Acrobat D...	182 KB
JavaScript Variables	07-02-2023 09:42	Microsoft Word D...	16 KB
TypeScript Quiz	01-02-2023 18:22	Microsoft Word D...	17 KB
style sheets	01-02-2023 18:09	Microsoft PowerP...	1,562 KB
Learning JavaScript	01-02-2023 18:07	Microsoft PowerP...	1,803 KB
client side programming	27-01-2023 17:57	Microsoft PowerP...	1,621 KB
HTML Element Reference	06-01-2023 18:03	Microsoft Word D...	36 KB
19AD652 - FULL STACK DEVELOPMENT L...	02-01-2023 14:32	Microsoft Word D...	21 KB



Typescript [Protected View] - Microsoft PowerPoint

Protected View This file originated from an Internet location and might be unsafe. Click for more details.

Typescript

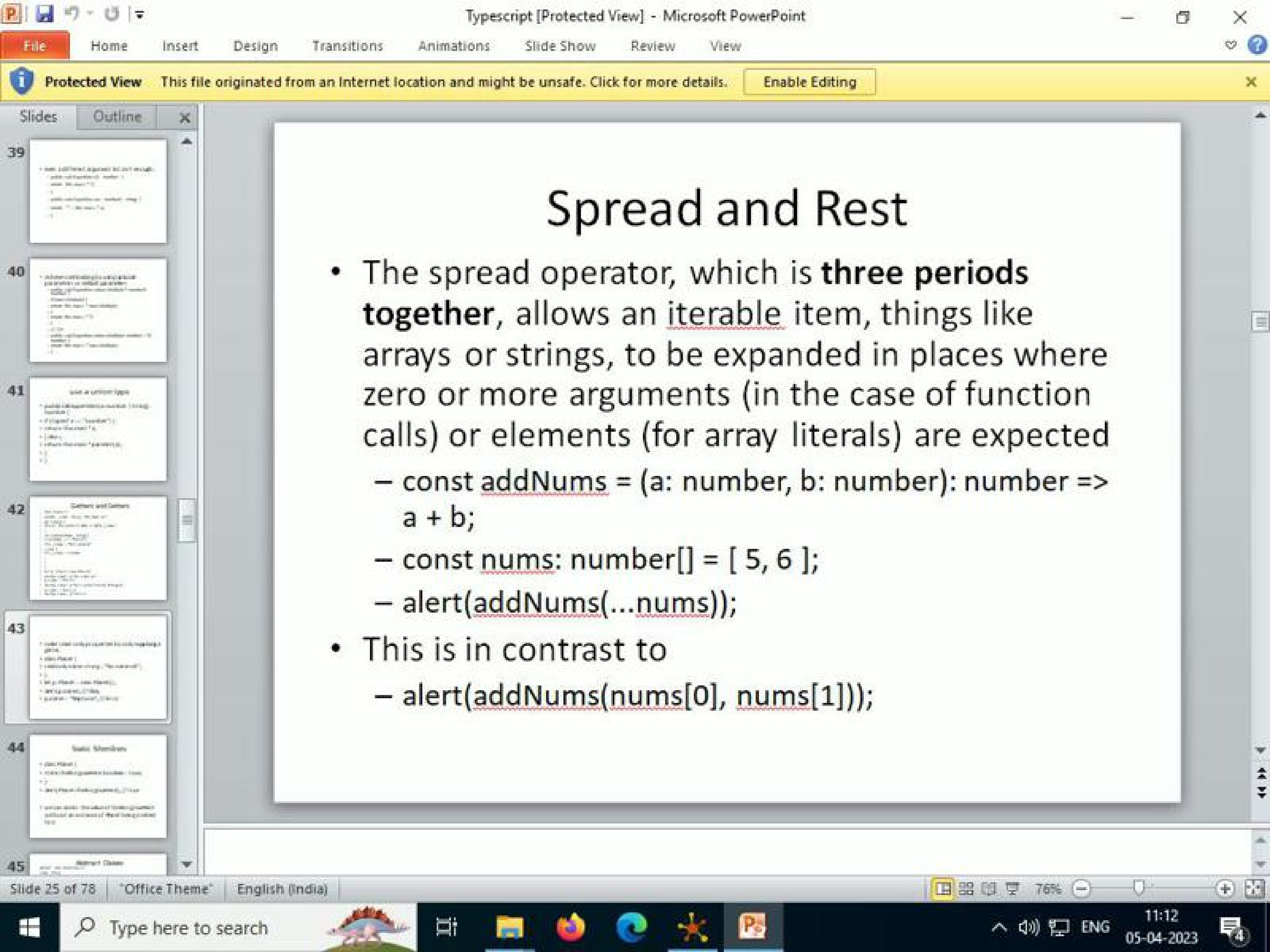


Typescript [Protected View] - Microsoft PowerPoint

Protected View This file originated from an Internet location and might be unsafe. Click for more details. Enable Editing

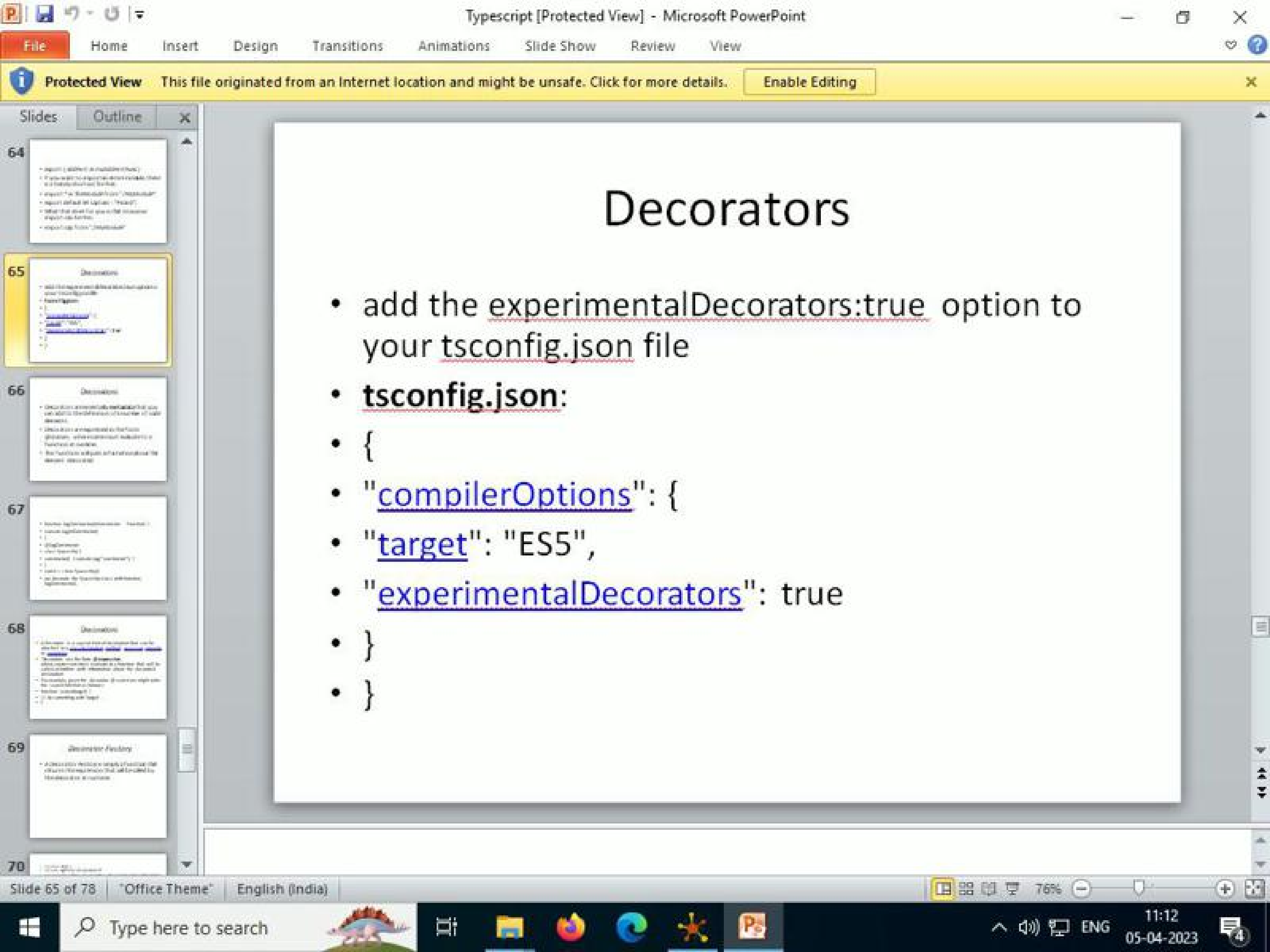
Typescript

Slide 1 of 78 "Office Theme" English (India)



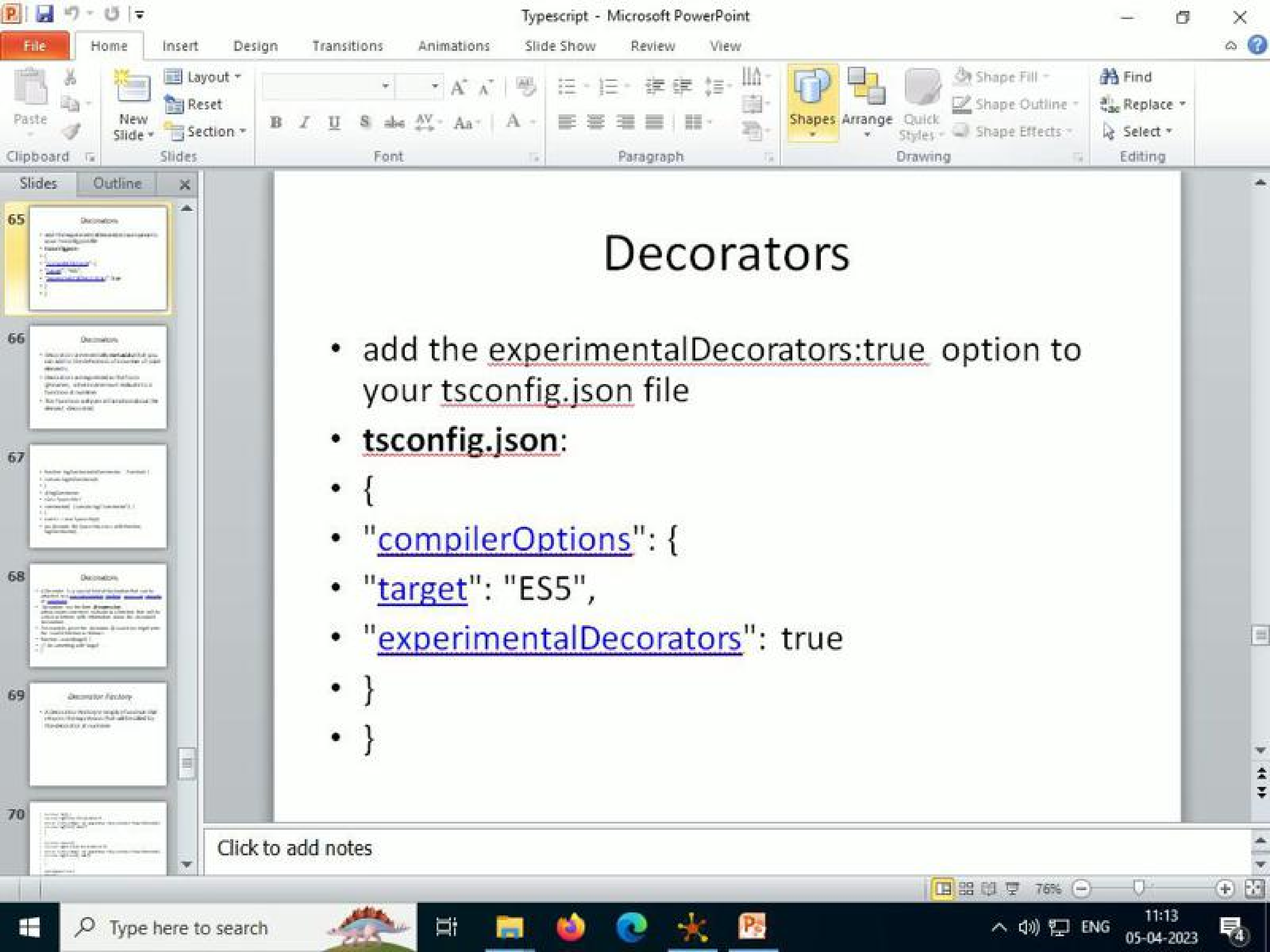
Spread and Rest

- The spread operator, which is **three periods together**, allows an iterable item, things like arrays or strings, to be expanded in places where zero or more arguments (in the case of function calls) or elements (for array literals) are expected
  - const addNums = (a: number, b: number): number => a + b;
  - const nums: number[] = [5, 6];
  - alert(addNums(...nums));
- This is in contrast to
  - alert(addNums(nums[0], nums[1]));


Typescript [Protected View] - Microsoft PowerPoint
File Home Insert Design Transitions Animations Slide Show Review View
Protected View This file originated from an Internet location and might be unsafe. Click for more details. Enable Editing
Slides Outline
Slide 65 of 78 "Office Theme" English (India) 76%
Type here to search
ENG 11:12 05-04-2023

Decorators

- add the experimentalDecorators:true option to your tsconfig.json file
- **tsconfig.json**:
- {
- "compilerOptions": {
- "target": "ES5",
- "experimentalDecorators": true
- }
- }



Decorators

- add the experimentalDecorators:true option to your tsconfig.json file
- **tsconfig.json**:
- {
- "compilerOptions": {
- "target": "ES5",
- "experimentalDecorators": true
- }
- }

Decorators

- add the experimentalDecorators:true option to your tsconfig.json file
- **tsconfig.json**:
- {
- "compilerOptions": {
- "target": "ES5",
- "experimentalDecorators": true
- }
- }

Decorators

- Decorators are essentially **metadata** that you can add to the definition of a number of code elements.
- Decorators are expressed in the form @<name>, where name must evaluate to a function at runtime.
- This function will pass information about the element decorated

- function logConstructor(inConstructor: Function) {
- console.log(inConstructor);
- }
- @logConstructor
- class Spaceship {
- constructor() { console.log("constructor"); }
- }
- const s = new Spaceship();
- we decorate the Spaceship class with function, logConstructor(),

Decorators

- A *Decorator* is a special kind of declaration that can be attached to a classdeclaration, method, accessor, property, or parameter.
- Decorators use the form **@expression**, where expression must evaluate to a function that will be called at runtime with information about the decorated declaration.
- For example, given the decorator @sealed we might write the sealed function as follows:
- function sealed(target) {
- // do something with 'target' ...
- }

- function logConstructor(inConstructor: Function) {
- console.log(inConstructor);
- }
- @logConstructor
- class Spaceship {
- constructor() { console.log("constructor"); }
- }
- const s = new Spaceship();
- we decorate the Spaceship class with function, logConstructor(),

Decorators

- A *Decorator* is a special kind of declaration that can be attached to a classdeclaration, method, accessor, property, or parameter.
- Decorators use the form **@expression**, where expression must evaluate to a function that will be called at runtime with information about the decorated declaration.
- For example, given the decorator @sealed we might write the sealed function as follows:
- function sealed(target) {
- // do something with 'target' ...
- }

Decorator Factory

- A *Decorator Factory* is simply a function that returns the expression that will be called by the decorator at runtime.

- function first() {
- console.log("first(): factory evaluated");
- return function (target: any, propertyKey: string, descriptor: PropertyDescriptor) {
- console.log("first(): called");
- };
- }
-
- function second() {
- console.log("second(): factory evaluated");
- return function (target: any, propertyKey: string, descriptor: PropertyDescriptor) {
- console.log("second(): called");
- };
- }
-
- class ExampleClass {
- @first()
- @second()
- method() {}
- }

```
function first() {
console.log("first(): factory evaluated");
return function (target: any, propertyKey: string, descriptor: PropertyDescriptor) {
console.log("first(): called");
};
}

function second() {
console.log("second(): factory evaluated");
return function (target: any, propertyKey: string, descriptor: PropertyDescriptor) {
console.log("second(): called");
};
}

class ExampleClass {
@first()
@second()
method() {}
}
```

- output to the console:
- first(): factory evaluated
- second(): factory evaluated
- second(): called
- first(): called

```
•   function first() {
•   console.log("first(): factory evaluated");
•   return function (target: any, propertyKey: string, descriptor: PropertyDescriptor) {
•   console.log("first(): called");
•   };
•   }
•
•   function second() {
•   console.log("second(): factory evaluated");
•   return function (target: any, propertyKey: string, descriptor: PropertyDescriptor) {
•   console.log("second(): called");
•   };
•   }
•
•   class ExampleClass {
•   @first()
•   @second()
•   method() {}
•   }
```

- output to the console:
- first(): factory evaluated
- second(): factory evaluated
- second(): called
- first(): called

Decorator Factories

- function logConstructorFactory(inEnabled: boolean) {
- if (inEnabled) {
- return function(inConstructor: Function) {
- console.log(inConstructor);
- }
- } else {
- return function() { };
- }
- }

```
@logConstructorFactory(true)
class Spaceship {
constructor() { console.log("Spaceship constructor"); }
}
@logConstructorFactory(false)
class Spacestation {
constructor() { console.log("Spacestation constructor"); }
}
const s = new Spaceship();
const t = new Spacestation();
```

Decorator Factories

- function logConstructorFactory(inEnabled: boolean) {
- if (inEnabled) {
- return function(inConstructor: Function) {
- console.log(inConstructor);
- }
- } else {
- return function() { };
- }
- }

```
@logConstructorFactory(true)
class Spaceship {
constructor() { console.log("Spaceship constructor"); }
}
@logConstructorFactory(false)
class Spacestation {
constructor() { console.log("Spacestation constructor"); }
}
const s = new Spaceship();
const t = new Spacestation();
```

```
@logConstructorFactory(true)
class Spaceship {
constructor() { console.log("Spaceship constructor"); }
}
@logConstructorFactory(false)
class Spacestation {
constructor() { console.log("Spacestation constructor"); }
}
const s = new Spaceship();
const t = new Spacestation();
```

Decorator Factories

- function logConstructorFactory(inEnabled: boolean) {
- if (inEnabled) {
- return function(inConstructor: Function) {
- console.log(inConstructor);
- }
- } else {
- return function() { };
- }
- }

- When executed, in the console you'll see
- VM73:11 class Spaceship {
- constructor() { console.log("Spaceship constructor"); }
- }
- VM73:16 Spaceship constructor
- VM73:22 Spacestation constructor

```
@logConstructorFactory(true)
class Spaceship {
constructor() { console.log("Spaceship constructor"); }
}
@logConstructorFactory(false)
class Spacestation {
constructor() { console.log("Spacestation constructor"); }
}
const s = new Spaceship();
const t = new Spacestation();
```

Decorator Factories

- function logConstructorFactory(inEnabled: boolean) {
- if (inEnabled) {
- return function(inConstructor: Function) {
- console.log(inConstructor);
- }
- } else {
- return function() { };
- }
- }

- When executed, in the console you’ll see
- VM73:11 class Spaceship {
- constructor() { console.log("Spaceship constructor"); }
- }
- VM73:16 Spaceship constructor
- VM73:22 Spacestation constructor

```
@logConstructorFactory(true)
class Spaceship {
constructor() { console.log("Spaceship constructor"); }
}
@logConstructorFactory(false)
class Spacestation {
constructor() { console.log("Spacestation constructor"); }
}
const s = new Spaceship();
const t = new Spacestation();
```

```
@logConstructorFactory(true)
class Spaceship {
constructor() { console.log("Spaceship constructor"); }
}
@logConstructorFactory(false)
class Spacestation {
constructor() { console.log("Spacestation constructor"); }
}
const s = new Spaceship();
const t = new Spacestation();
```

```
@logConstructorFactory(true)
class Spaceship {
constructor() { console.log("Spaceship constructor"); }
}
@logConstructorFactory(false)
class Spacestation {
constructor() { console.log("Spacestation constructor"); }
}
const s = new Spaceship();
const t = new Spacestation();
```

```
@logConstructorFactory(true)
class Spaceship {
constructor() { console.log("Spaceship constructor"); }
}
@logConstructorFactory(false)
class Spacestation {
constructor() { console.log("Spacestation constructor"); }
}
const s = new Spaceship();
const t = new Spacestation();
```

Third-Party Libraries

- Add in third-party TypeScript libraries
- Example, to use the popular Lodash library in your code
- npm install --save lodash
- also import another related library:
- npm install --save-dev @types/lodash
- This extra library is called a type declaration file, or a type binding file sometimes, and it's what tells TypeScript (tsc, more specifically) all about the types that Lodash uses and provides.

Debugging TypeScript Apps

- Source Maps
- tsc --sourceMap app.ts
- {
- "version": 3,
- "file": "app.js",
- "sourceRoot": "",
- "sources": ["app.ts"],
- "names": [ ],
- "mappings": "AAAA,SAAS,KAAK,CAAC,SAAiB;IAC9B,KAAK,CAAC,YAAU,SAAS,MAAAG,CAAC,
- CAAC;AAChC,CAAC;AACD,KAAK,CAAC,gBAAgB,CAAC,CAAC"
- }

Debugging TypeScript Apps

- Source Maps
- tsc --sourceMap app.ts
- {
- "version": 3,
- "file": "app.js",
- "sourceRoot": "",
- "sources": ["app.ts"],
- "names": [ ],
- "mappings": "AAAA,SAAS,KAAK,CAAC,SAAiB;IAC9B,KAAK,CAAC,YAAU,SAAS,MAAG,CAAC,
- CAAC;AAChC,CAAC;AACD,KAAK,CAAC,gBAAgB,CAAC,CAAC"
- }