

Java Interfaces Part2

Difference b/w extends vs implements
extends :: One class can extends only class at a time
eg:: class One{}
class Two extends One{}

```
interface IOne{
    void m1();
}

interface ITwo{
    void m2();
}

class CommonImpl implements IOne,ITwo{
    public void m1(){}
}
```

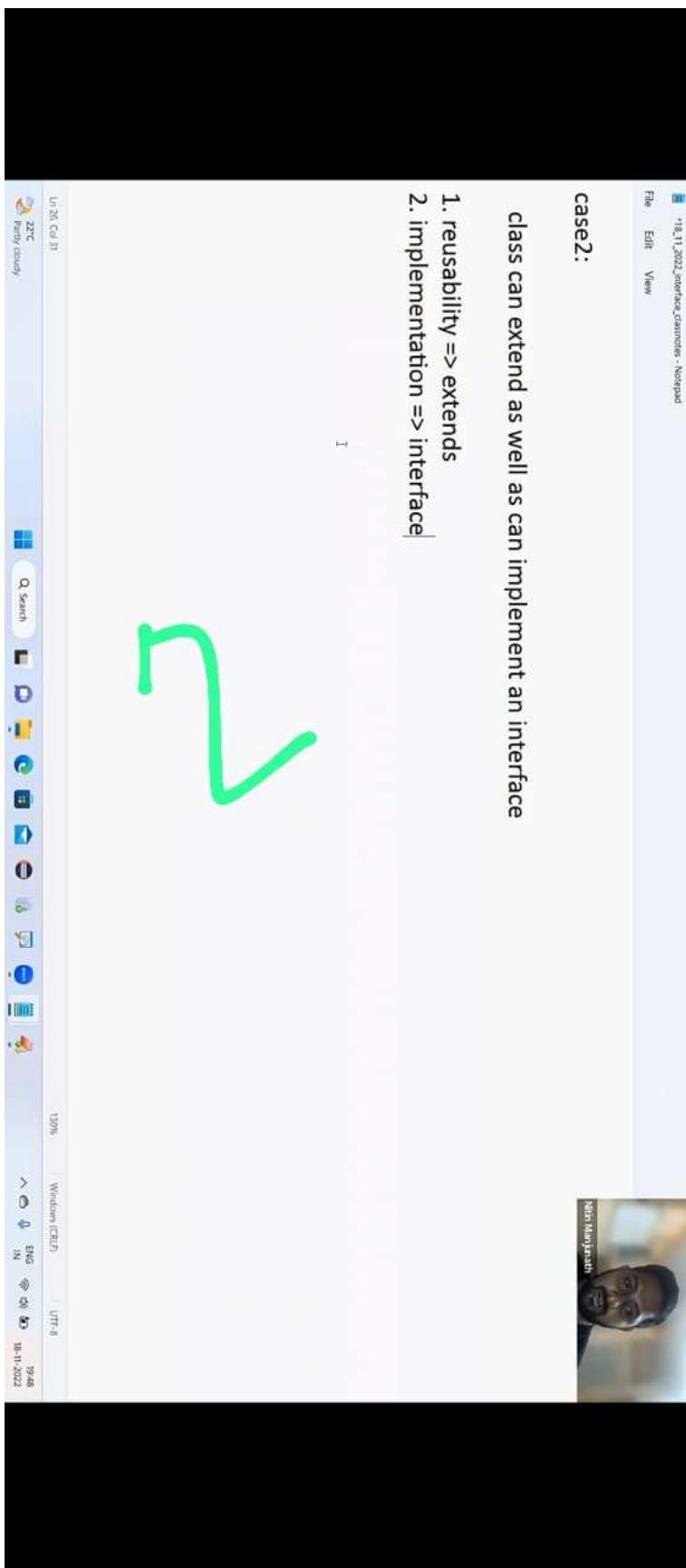
File Edit View

Nitin Manjrekar

18.11.2022_interface_classes - Notepad
22°C
Polluted air



Ln 14 Col 16 15% Windows (CEP) UTF-8 ENG IN 18.11.2022 19:43



Editor - (D:\NetBeansProjects)

File Edit View Search Document Project Tools Browser Emmet Window Help

Directory Object Functions

[D:] New Volume

D:\

- SECOND BIN
- Codebase
- Enterprise
- enterpriseEdition
- FieldPrograms
- Front end codes
- gitHub
- gitment
- miniproject

Test.java

TestApp.java

```
1 class Sample
2 {
3     public void m1(){
4
5     }
6 }
7 }
8 interface IDemo
9 {
10     void m2();
11 }
12
13 class DemoImpl extends Sample implements IDemo
14 {
15 }
16
17
18 public class TestApp {
19
20     public static void main(String[] args){
21
22     }
23 }
```

Java (*.java)

TestApp.java

For Help press F1

22°C Party cloudy

Q. Search

In 15 col 1 24 00 PC ANSI

ENG ⇔ ⏺ IN 19:11:19:50 19-11-2022

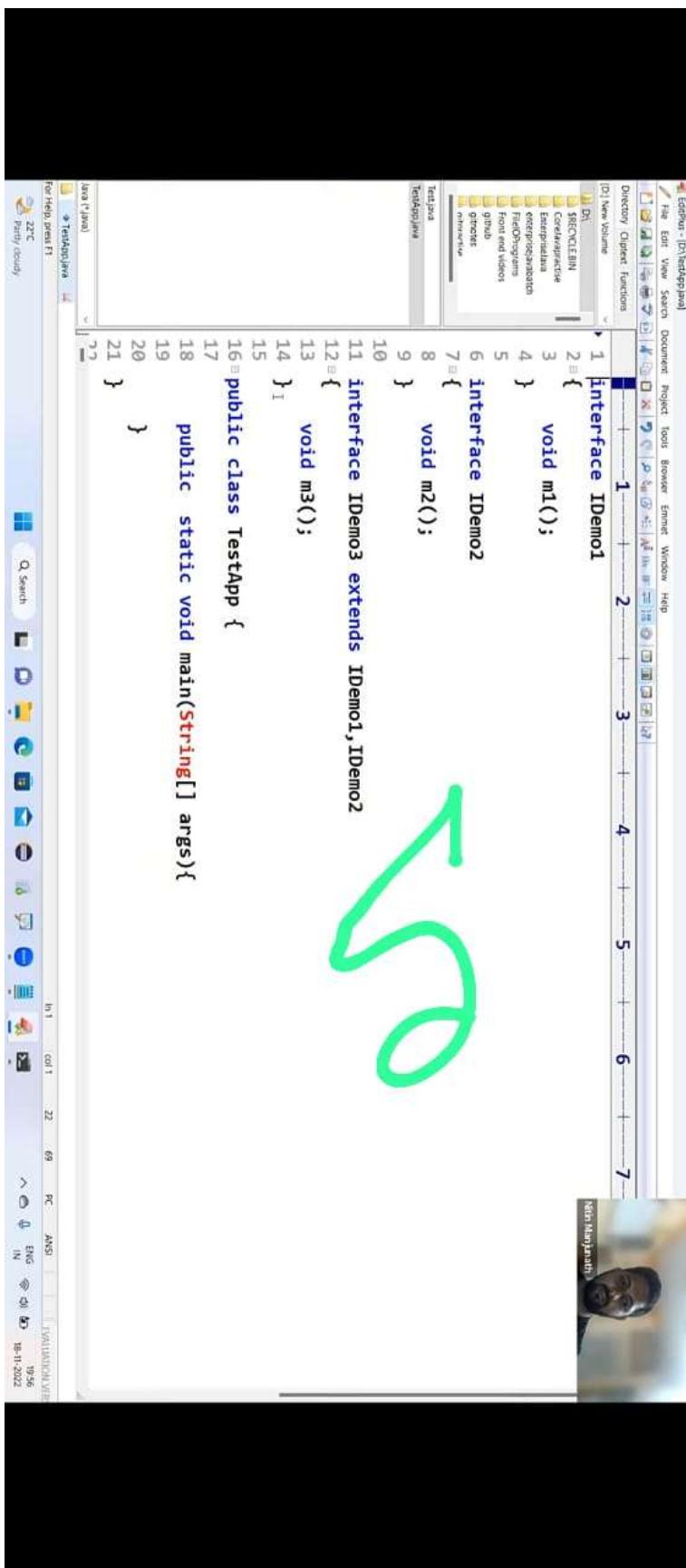


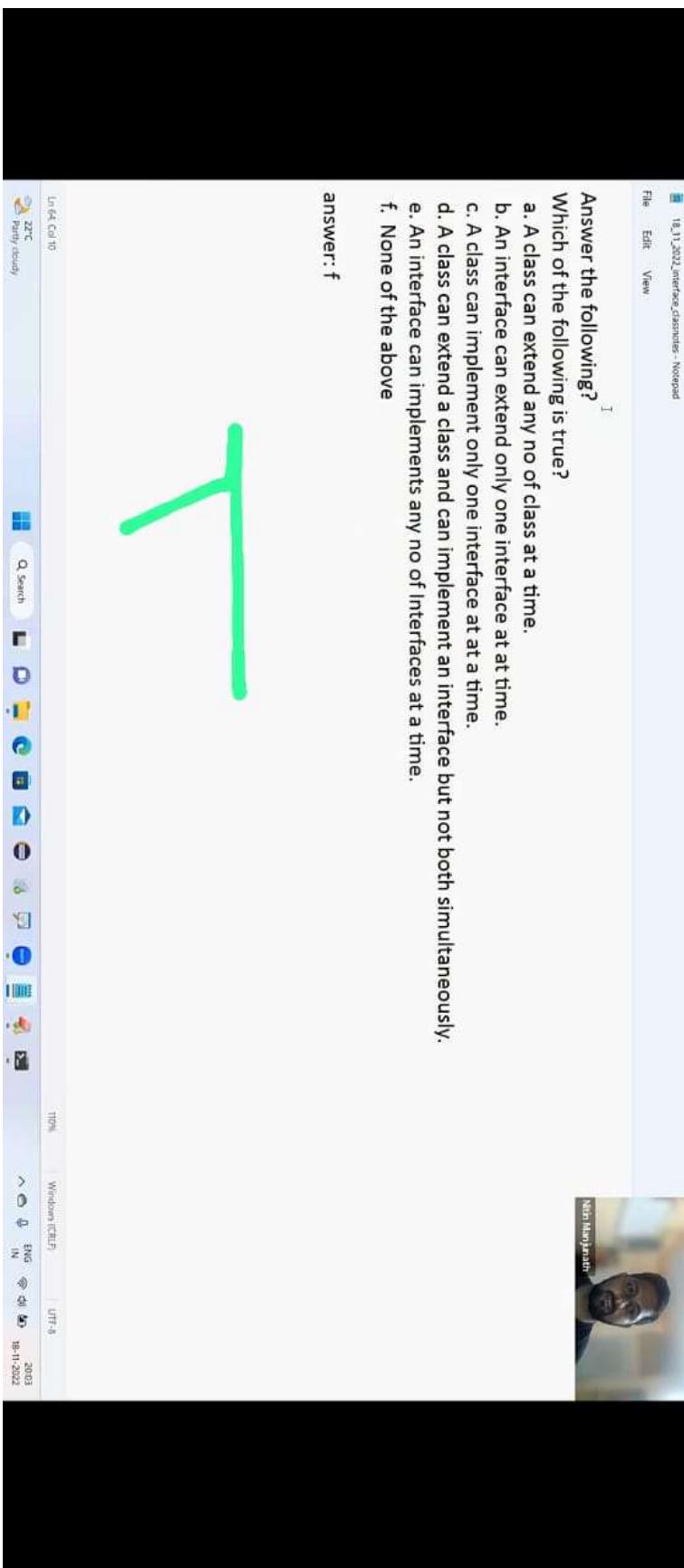
case2:
A class can extend a class and can implements any no of interfaces simultaneously.

```
eg: interface One{  
    public void methodOne();  
}  
class Two{  
    public void methodTwo(){  
}  
}  
class Three extends Two,implements One{  
    @Override  
    public void methodOne(){  
        ...  
    }  
    @Override  
    public void methodTwo(){  
        ..  
    }  
}
```



✓





File Edit View

f. None of the above

answer: f

Consider the expression X extends Y which of the possibility of X and Y expression is true?

1. Both x and y should be classes.
2. Both x and y should be interfaces.
3. Both x and y can be classes or can be interfaces.
4. No restriction.

Equation : X extends Y ==> true
class extends class ==> true
interface extends interface ==> true

Answer: 3



18.11.2022 interface classmate - Narend

Q. Search

10% Windows (CEP) | UTT-3
18.11.2022

20°C Party cloudy



*18.11.2022_interface_diaries - Notepad
File Edit View

interface extends interface => true

Answer: 3

Q>

Predict X,Y,Z

- a. X extends Y,Z?
- b. X extends Y implements Z?
- c. X implements Y,Z?
- d. X implements Y extends Z?

9

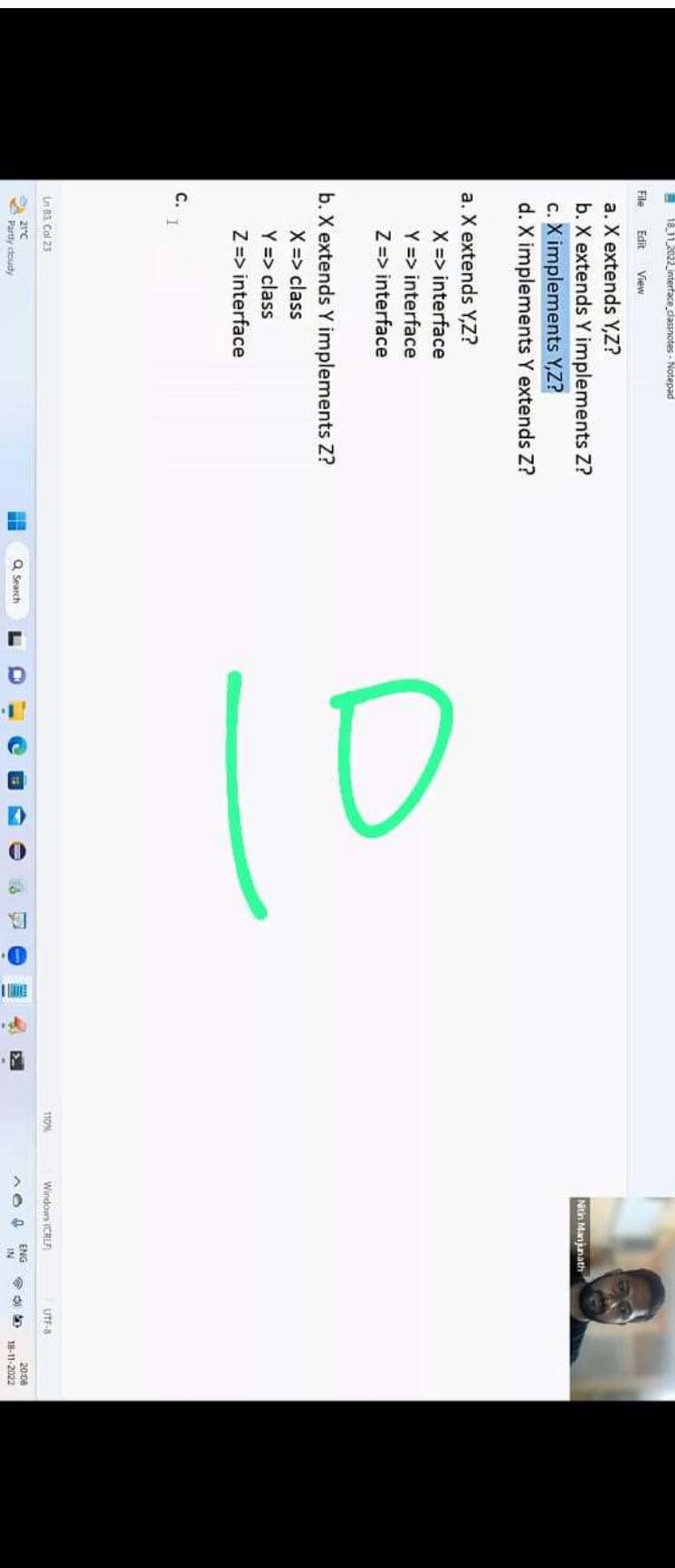


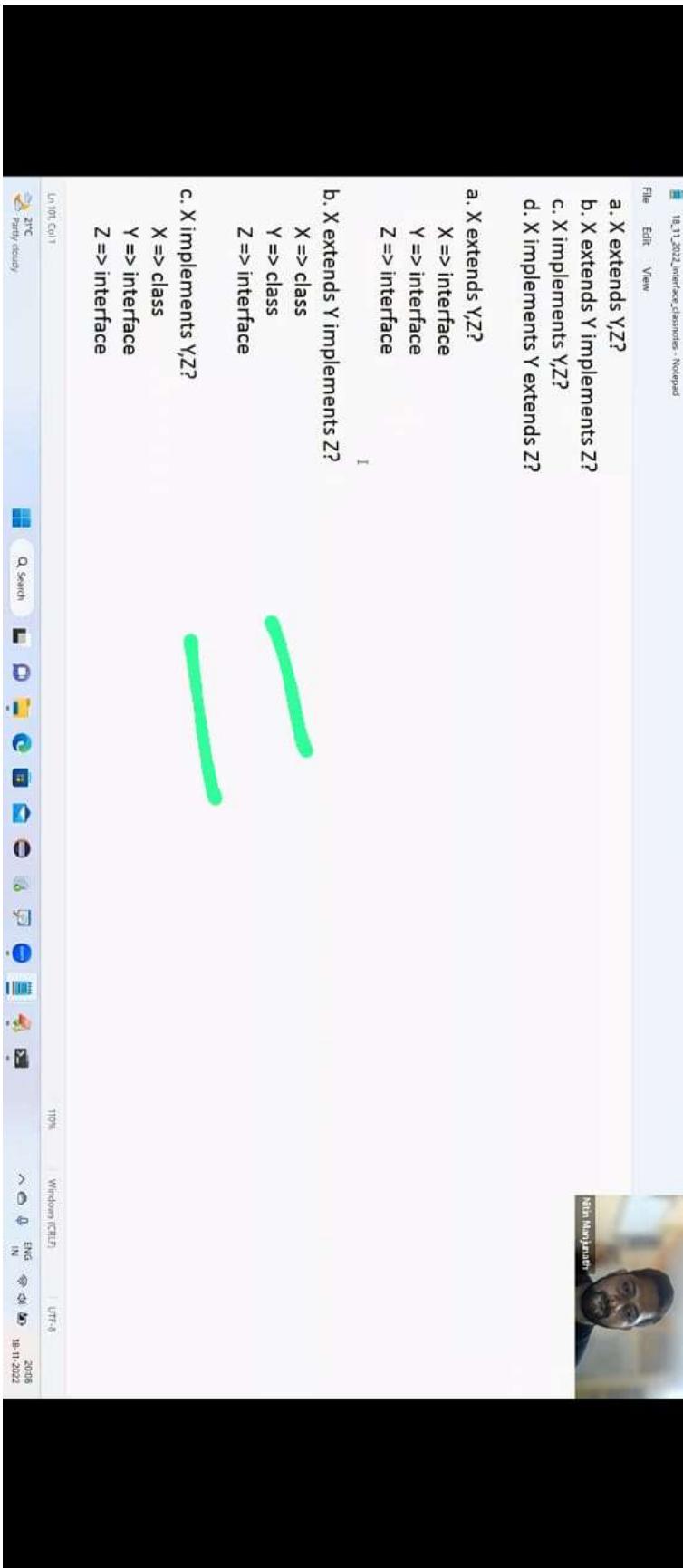
Ln: 91 Col: 29

20°C Party cloudy

10% Windows (CEP7) | UTF-8

^ < > ENG IN 19.11.2022





File Edit View

d. X implements Y extends Z
combination is illegal

Interface Methods

Every method present inside the interface is public.
Every method present inside the interface is abstract.
How many declaration are valid?

- b. public void methodOne();
 - c. abstract void methodOne();
 - d. public abstract void methodOne();

answer: All are valid



The image shows a Windows taskbar at the bottom of a screen. It features several pinned icons on the left, including Start, Search, Task View, File Explorer, Edge, Mail, Photos, OneDrive, Settings, Control Panel, and a Microsoft Store icon. On the right side, there are system status indicators for battery level (110%), signal strength, volume, and a clock showing 10:11. The taskbar has a light blue background.

100_1_JavaScript_in_het_Web_developers_handboek - Index.html

Every method present inside the interface is abstract.
How many declaration are valid?

- b. public void methodOne();
 - c. abstract void methodOne();
 - d. public abstract void methodOne();

Answer: All are valid

public => To make the method available for every implementation class.
abstract => Implementation class is responsible for providing the implementation.

eg: jdbc api (java.sql.*)

implementation should be given by

- b. oracle
 - c. postgresql

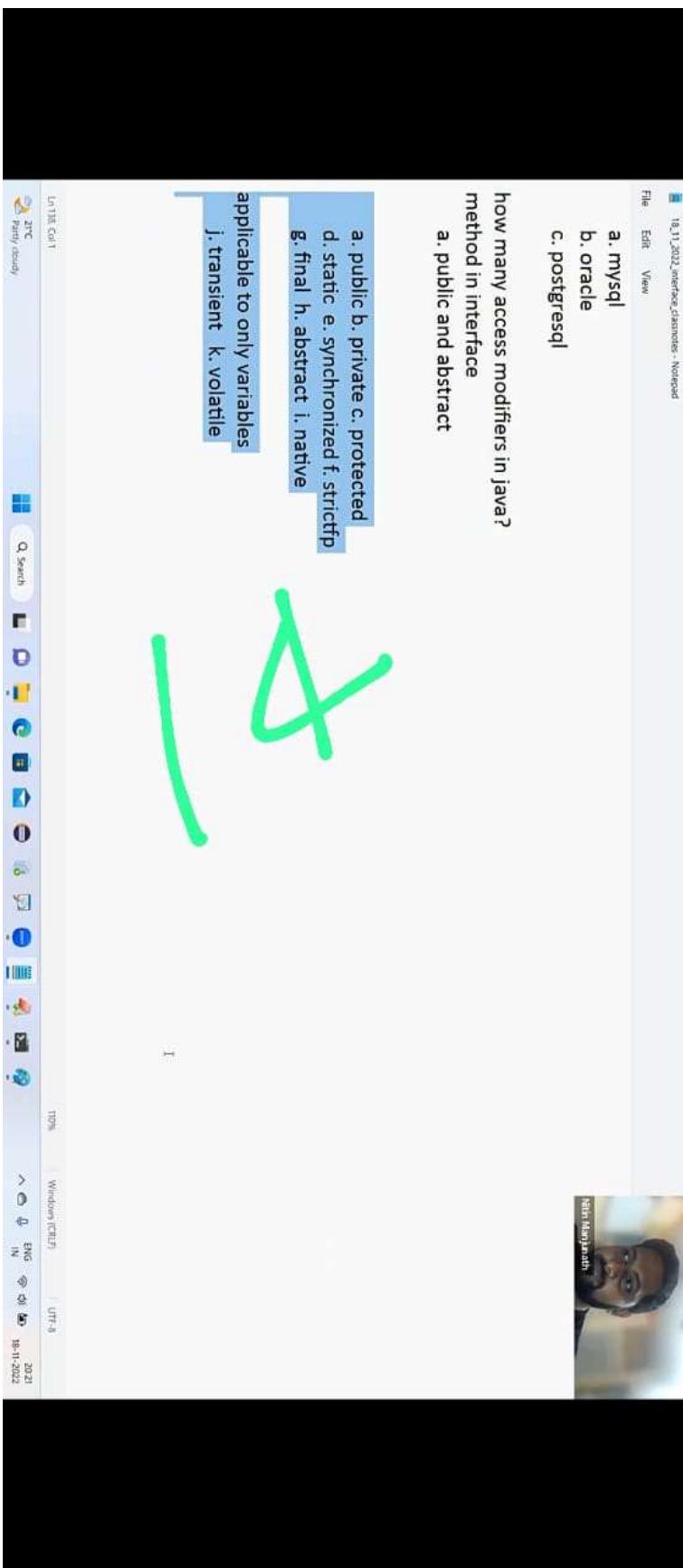


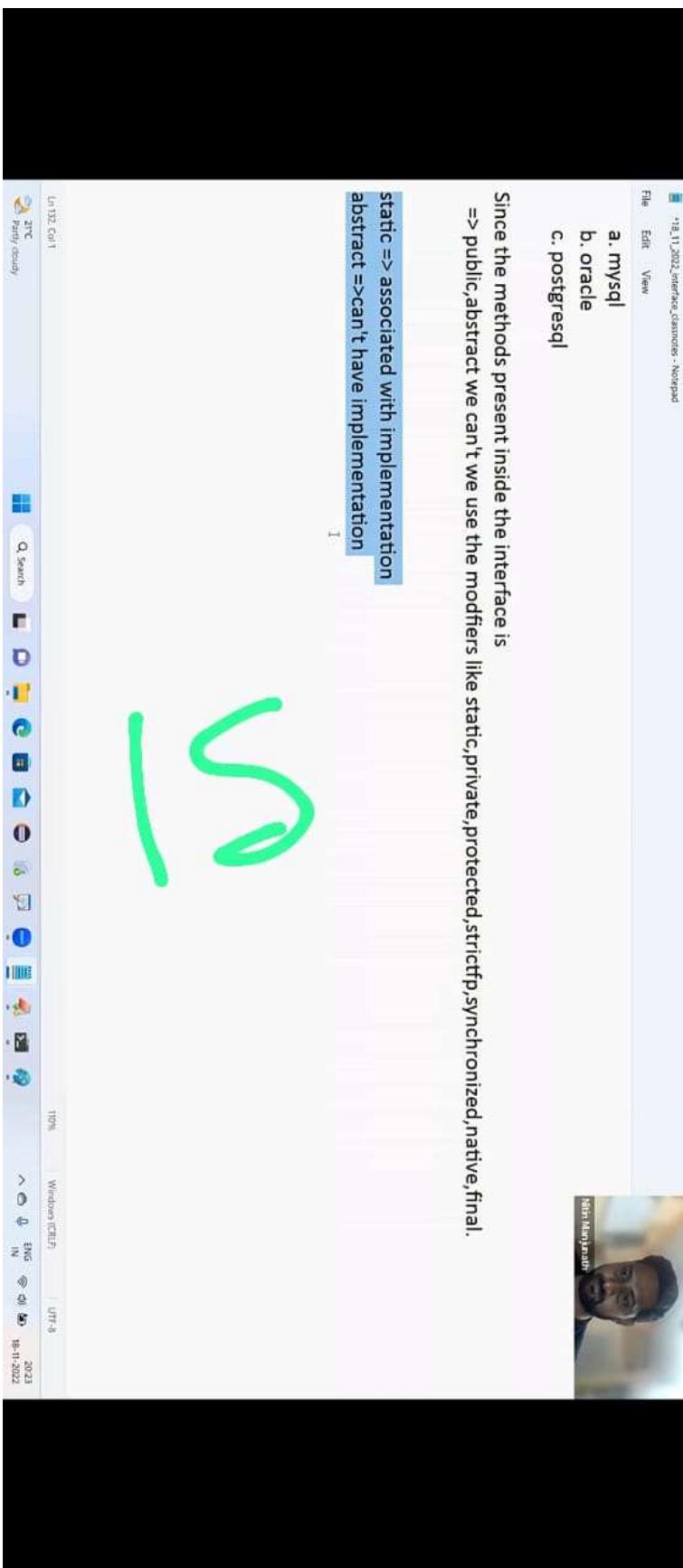
Int'l Col't



110% Windows (C:\LF) UTF-8







Since the methods present inside the interface is
=> public,abstract we can't we use the modifiers like static,private,protected,strictfp,synchronized,native,final.

static => associated with implementation
abstract =>can't have implementation

File Edit View
18.11.2022_Interface_classeses - Notepad



Interface variables

- => Inside the interface we can define variables.
- => Inside the interface variables define requirement level constants.
- => Every variable present inside the interface is by default public static final.

```
eg:- interface ISample{
```

```
    int x=10;
```

```
}
```

public :: To make it available for implementation class Object.
static :: To access it without using implementation class Name.
final :: Implementation class can access the value without any modification.

100% Windows (CE/P) UTF-8
18.11.2022 20:31
Nitin Manjanna
21°C Partly cloudy

18.11.2022_interface_classnotes - Notepad

File Edit View

static :: To access it without using implementation class Name.

final :: Implementation class can access the value without any modification.

variable declaration inside interface

- a. int x=10;
- b. public int x=10;
- c. static int x=10;
- d. final int x=10;
- e. public static int x=10;
- f. public final int x=10;
- g. static final int x=10;
- h. public static final int x=10;

Answer: All are legal



Ln 1% Col 1 20°C Party cloudy



100% Windows (C:\EF7) UTF-8

ANSI

UTF-8

File Edit View
=> interface variables can be accessed from implementation class, but cannot modify if we try to modify it would result in compile time error.

```
eg:: interface Remote{  
    int VOLUME = 100;  
}  
  
class Lg implements Remote{  
    public static void main(String... args){  
        VOLUME=0; //CE:: cannot assign a value to final variable VOLUME  
        System.out.println("value of volume is ::"+VOLUME);  
    }  
}  
  
eg:: interface Remote{  
    int VOLUME = 100;  
}  
  
class Lg implements Remote{  
    public static void main(String... args){  
        int VOLUME=0;  
        System.out.println("value of volume is ::"+VOLUME); //0
```

```
18.11.2022_interface_classeses - Notepad
File Edit View
Nitin Manjanna
class Lg implements Remote{
public static void main(String... args){
    int VOLUME=0;//CE:: cannot assign a value to final variable VOLUME
    System.out.println("value of volume is ::"+VOLUME);
}
}

eg:: interface Remote{
}
int VOLUME = 100;
class Lg implements Remote{
public static void main(String... args){
    int VOLUME=0;//local variable
    System.out.println("value of volume is ::"+VOLUME);
}
}

19
```

Editor - (D:\TestApp.java)

File Edit View Search Document Project Tools Brower Ethernet Window Help

Directory Object Functions

[D:] New Volume

D:
RECYCLE.BIN
CorbaApache
EnterpriseJavaBeans
FieldPrograms
Front end servers
github
gitment
miniproj

1 interface ISample

2 {

3 int a=10;//public static final

4 }

5 public class TestApp implements ISample{

6 public static void main(String[] args){

7

8 int a = 20;

9 System.out.println(a);// 20

10 System.out.println(ISample.a);//10

11 }

12 }

13 }

14 }

Java (*.java)

* TestApp.java

For Help press F1

20C Party cloudy

Q Search

In 10 col 43 14 00 PC ANSI ↑ ↓ 192.168.1.100:2054

ENG ⇔ CHI 🇮🇳 IN 19-11-2022

20

18.11.2022_interface_classnotes - Notepad

File Edit View

=====

Case 1::

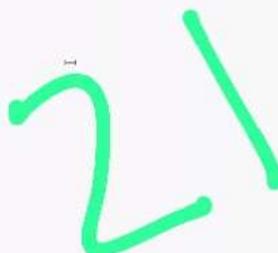
If 2 interfaces contain a method with same signature and same return type in the implementation class only one method implementation is enough.

e.g.:

```
interface Left{public void methodOne();}
interface Right{public void methodOne();}
class Test implements Left,Right{
    @Override
    public void methodOne(){
        ...
    }
}
```

Case2:

If 2 interfaces contain a method with same name but different arguments in the implementation class we have to provide implementation for both methods and these methods acts as a Overload methods.



```
18.11.2022_interface_classeses - Notepad
File Edit View
}
}

Niran Marjirath

18.11.2022_interface_classeses - Notepad
File Edit View
}
}

Niran Marjirath

18.11.2022_interface_classeses - Notepad
File Edit View
}
}

Niran Marjirath
```

Case2:

If 2 interfaces contain a method with same name but different arguments in the implementation class we have to provide implementation for both methods and these methods acts as a Overload methods.

e.g.:

```
interface Left{ public void methodOne();}
interface Right{public void methodOne(int i);}
class Test implements Left,Right{
    @Override
    public void methodOne(){
        ...
    }

    @Override
    public void methodOne(int i){}
```

File Edit View Search Document Project Tools Browser Internet Window Help

D:\ New Volume SRECYCLEBIN Confluence EnterpriseArchitect enterprisebatch FieldPrograms Front end videos gribb greatest primitive

Test.java TestApp.java

```
10
11 class CommonImpl implements ISample, IDemo
12 {
13     @Override
14     public void m1(){}/valid
15
16     @Override
17     public int m1(){}/valid
18 }
19 public class TestApp {
20     public static void main(String[] args){
21
22
23     }
24 }
25
26 Q> Can a java class implements 2 interfaces simultaneously?
27 yes possible, but in both the interfaces method signature should be same, but not
28 different return types.
29
30
```

In 28 col 28 30 00 PC ANSI VALIDATE(MVIE)

Java (*.java) * TestApp.java

Re-Hit, press F1 20°C Mostly clear

Nitin Chiplapath

Editor - (D:\testapp)

File Edit View Search Document Project Tools Browser Ethernet Window Help

Directory Object Functions

D:\

1 RECENT BIN
2 CacheManager
3 Configuration
4 Enterprise
5 Front end services
6 GitHub
7 miniproject

Test.java

TestApp.java

Java (*.java)

For Help press F1

20°C
Mostly clear

Q Search

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

```
1 interface ISample
2 {
3     int x = 8888;
4 }
5
6 interface IDemo
7 {
8     int x = 9999;
9 }
10
11 class TestApp implements ISample, IDemo
12 {
13     public static void main(String[] args){
14         System.out.println("ISample variable x is :: "+ISample.x);
15         System.out.println("IDemo variable x is :: "+IDemo.x);
16     }
17 }
18
19
20
21
```

X



File Edit View

Note:

Q> Can a java class implements 2 interfaces simultaneously?
yes possible, except if two interfaces contains a method w

Variable naming conflicts:
Two variables can contain a variable with same name and there may be a chance variable naming

example:

```
interface Left{ int x=888;}  
interface Right{ int x=999;}  
public class Test implements Left,Right{  
    public static void main(String... args){  
        System.out.println(Left.x);  
        System.out.println(Right.x);  
    }  
}
```



In 254 CalT
20°C
Moody clear

110% Windows (CRLF) UTF-8

Command prompt

```
D:\>javap java.lang.Runnable  
Compiled from "Runnable.java"  
public interface java.lang.Runnable {  
    public abstract void run();  
  
D:\>javap java.io.Serializable  
Compiled from "Serializable.java"  
public interface java.io.Serializable {  
  
D:\>javap java.lang.Cloneable  
Compiled from "Cloneable.java"  
public interface java.lang.Cloneable {  
  
D:\>
```

20°C
Matty clear

Nitin Marjumath

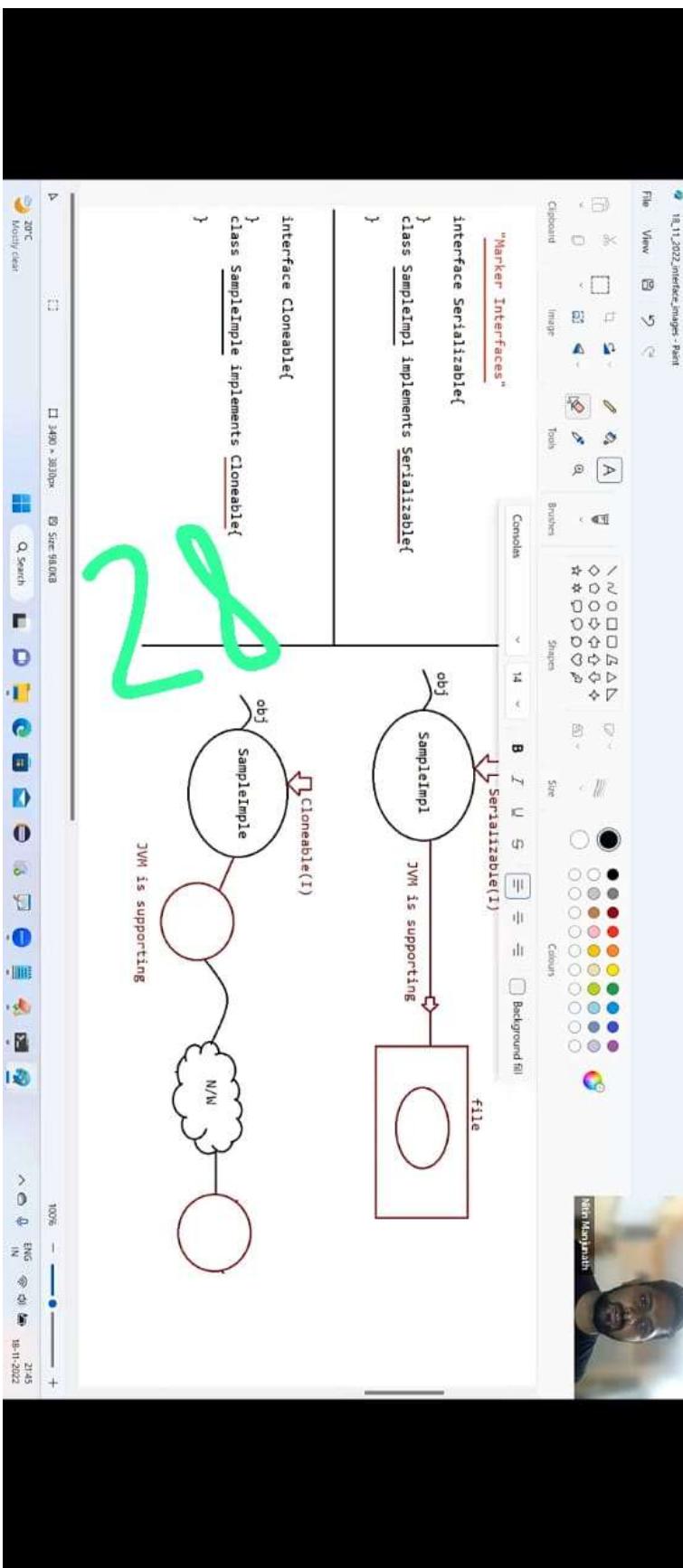


21:34
ENG ⇄ DE IN 19.11.2022

Note:
inside interface the methods are by default "public and abstract".
inside interface the variables are by default " public static and final".
We can also write an interface without any variable or abstract methods.

```
interface Serializable{  
}  
class SampleImpl implements Serializable{  
}  
  
interface Cloneable{  
}  
class SampleImpl implements Cloneable{  
}
```





18.11.2022 interface_classes.txt Notepad

File Edit View

Nitin Mahapatra

example1

By implementing Serializable interface we can send that object across the network and we can save state of an object into the file.

example2

By implementing SingleThreadModel Interface servlet can process only one client request at a time so that we can get "Thread Safety".

example3

By implementing Cloneable Interface our object is in a position to provide exactly duplicate cloned object.

Without having any methods in marker interface how objects will get ability?

Ans. JVM is responsible to provide required ability.

Why JVM is providing the required ability to Marker Interfaces?

Ans. To reduce the complexity of the programming.

Can we create our own marker interface?

29

Ln 286 Col 4 100% Windows (CET) UTF-8 19.11.2022 21:47

20°C Mostly clear

Q Search

18.11.2022_Interface_classeses - Notepad

File Edit View

}

MarkerInterface

=====

=> If an interface does not contain any methods and by implementing that interface if our Object will get some ability such type of interface are called "Marker Interface"/"Tag Interface"/"Ability Interface".

=> example

Serializable,Cloneable,SingleThreadModel.



- example1
By implementing Serializable interface we can send that object across the network and we can save state of an object into the
- example2
By implementing SingleThreadModel interface servlet can process only one client request at a time so that we can get "Three
- example3
By implementing Cloneable interface our object is in a position to provide exactly duplicate cloned object.

Ln 277 Col 10

20°C

Partly clear

Q Search

Windows (CEP)

UTF-8

100%

Windows (CEP)

UTF-8

ENG

DE

IN

18.11.2022

21:46

File Edit View Search Document Project Tools Browser Emmet Window Help

Directory Object Functions

[D] New Volume

D: RECENT BIN Control Apache Examples JavaBeans Front end videos GitHub Gitment Miniproj

Test.java TestApp.java

Java (*.java) * TestApp.java For Help press F1 20°C Mostly clear

1 interface IDemo
2 {
3 }
4
5 class TestApp implements IDemo
6 {
7 public static void main(String[] args){
8 }
9 }
10 }
11
12
13 //JVM ---> customize the jvm by writing lines of code for ur marker interface.
14
15
16
17



Editor - D:\NetBeansProjects

File Edit View Search Document Project Tools Browser Ethernet Window Help

Directory Object Folders

D:\

- SECOND BIN
- Codebase
- Enterprise
- enterprise
- Front end videos
- gitHub
- gitment
- miniproject

Text.java TestApp.java

```
1 interface IDemo
2 {
3     //SRS(100% abstract)
4     void m1();
5     void m2();
6     void m3();
7     void m4();
8     void m5();
9 }
10 abstract class AdapterClass implements IDemo
11 {
12     public void m1(){}
13     public void m2(){}
14     public void m3(){}
15     public void m4(){}
16     public void m5(){}
17 }
18 class DemoImpl extends AdapterClass
19 {
20     public void m3(){
21         System.out.println("hey i can give implementation");
22     }
23 }
```

Java (*.java)

TestApp.java

For Help press F1

IPC

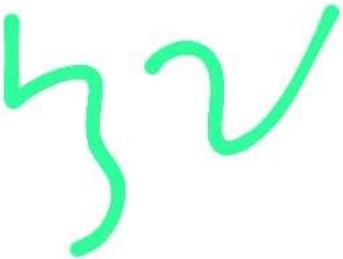
Memory clear

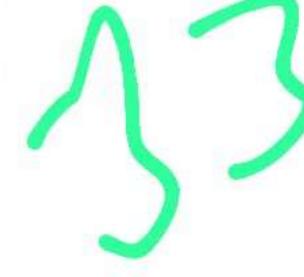
Q Search

In 17 col 2 6 00 PC ANSI

ENGLISH

IN 19-11-2022





```
EditorBar - [UntitledJava]
File Edit View Search Document Project Tools Browser Internet Window Help
D:\ New Volume
SRCYLEBIN
CoreCompetitive
Enterpreneur
entrepreneurbatch
FieldOfGuns
Front end videos
gribib
gratist
niravite
Text.java
TextAdapter
10 } abstract class AdapterClass implements IDemo
11 { public void m1() {}
12     public void m2() {}
13     public void m3() {}
14     public void m4() {}
15     public void m5() {}
16
17 }
18 class DemoImpl extends AdapterClass
19 {
20     public void m3(){
21         System.out.println("hey i can give implementation");
22     }
23 }
24
~1sec. Tackling Implementations Then
```

Java (*.java)

File Help press F1

19°C
Mony clear

Q. Search

In 17 col 2 36 00 PC ANSI | VALIDATIONVIEW

19-11-2022 21:56

Moni Raghavath

File Edit View Search Document Project Tools Browser Ethernet Window Help

Directory Object Folders

[D] New Volume 16 1 2 3 4 5 6 7

D:\ RECYCLE BIN
Core expressive
Enterprise
FieldPrograms
Front end videos
github
gitment
miniproj

17 }
18 class DemoImpl extends AdapterClass
19 {
20 public void m5(){
21 System.out.println("hey i can give implementation");
22 }
23 }
24 TestApp.java
25 class TestApp implements IDemo
26 {
27 public static void main(String[] args){
28 IDemo demo = new DemoImpl();
29 demo.m3();
30 }
31 }
32 }
33 }
34 }
35 }
36 }
37 }

Java (*.java)

* TestApp.java

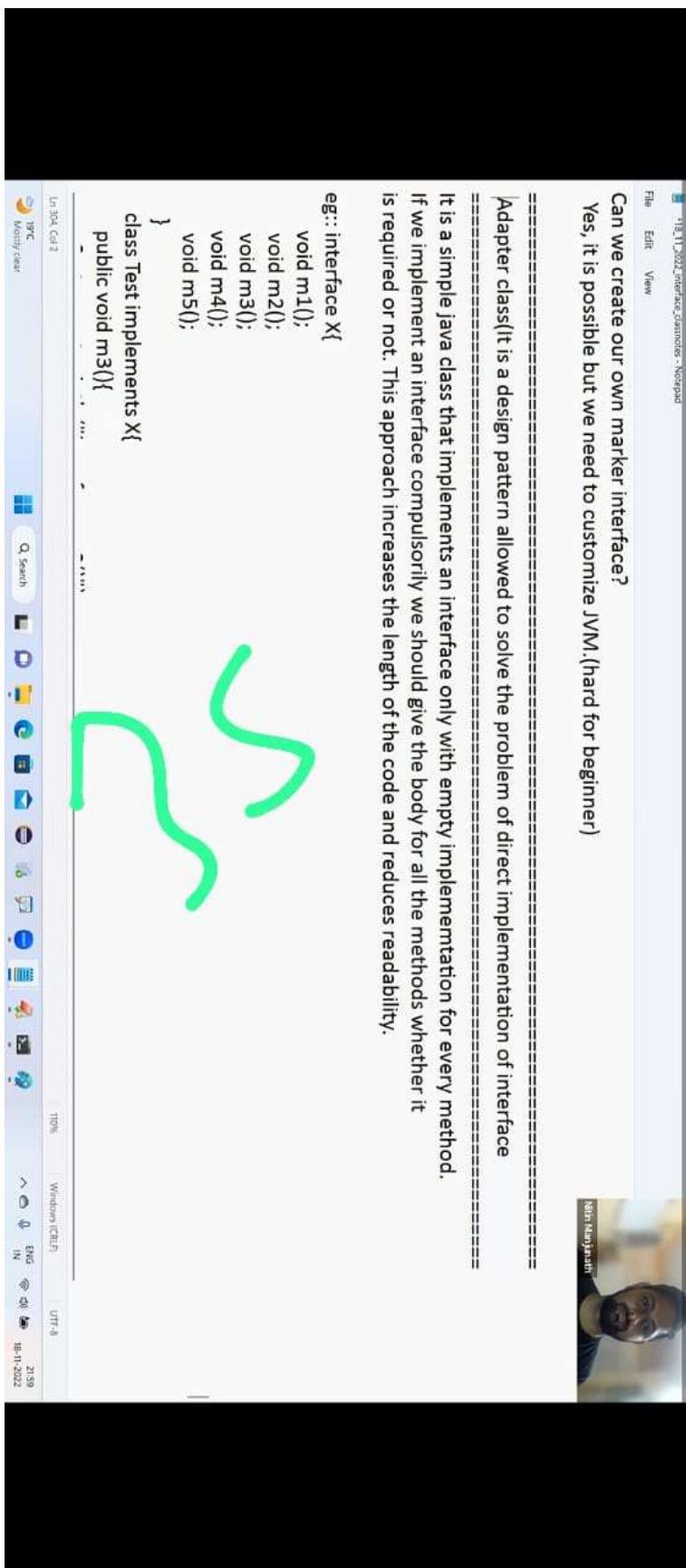
For Help press F1

IPC Monitor clear

Nitin Marajath

Int 29 00:19 2 00 PC ANSI 192.168.1.11 2158 ENG ⌂ ⌃ ⌄ ⌅ ⌆ 19-11-2022

4



Editor - (D:\TestAppJava)

File Edit View Search Document Project Tools Browser Ethernet Window Help

D:\ New Volume Directory Object Folders

1 2 3 4 5 6 7

25 class Testapp implements IDemo

26 {

27 public static void main(String[] args){

28 IDemo demo = new DemoImpl();

29 demo.m3();

30 }

31 }

32

33 // Servlet(I)

34 | implements

35 // GenericServlet(Abstract class)

36 | extends

37 // HttpServlet(Abstract class)

38 | extends

39 // MyServlet(class)

40

41

42

43

44

45

Java (*.java)

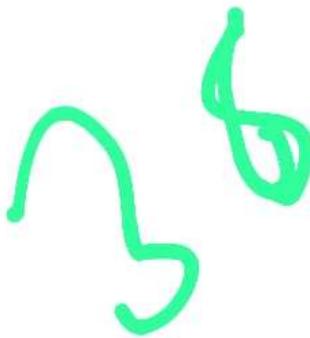
TestAppJava

For Help press F1

IPC

Memory clear

In 39 col 1 45 20 PC ANSI ENG IN 19.11.2022 22:03



18.11.2022 snippets-classnotes - Notepad

File Edit View

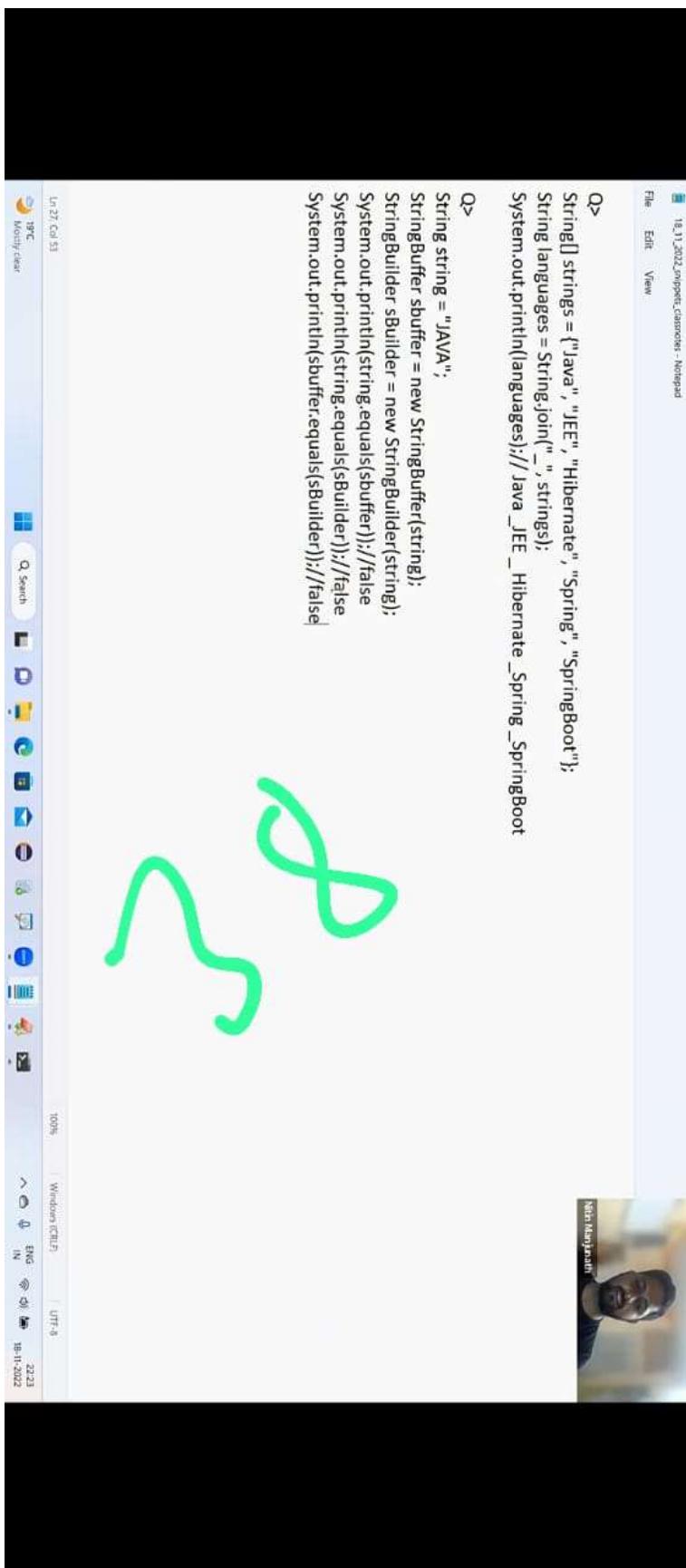
Q> String s1 = "null"+null+1;
System.out.println(s1);//nullnull1

Q>
String s1 = 1+null+"null";
System.out.println(s1);//CE

Q>
String str = "sachin ramesh tendulkar";
System.out.println(str.indexOf('a') + str.indexOf("dulkar"));//18

Q> String name = null;
System.out.println(name.toUpperCase());//NPE





Editor - (D:\TestApp.java)

File Edit View Search Document Project Tools Browser Ethernet Window Help

Directory Object Folders

D:\ New Volume

1 S E C O N D E R Y

2 CoreJava

3 E m p u t a u

4 empergation

5 F i e l d s

6 g i t b u

7 min

Test.java

TestApp.java

Q Search

1 Java (*.java)

2 TestApp.java

3 For Help press F1

4 IPC

5 Battery clear

6 In 10 col 2 17 00 PC ANSI

7 ^ < > & & Eng & & 22:27

8 19-11-2022

9 10 11 12 13 14 15 16 17

39

```
1 class TestApp
2
3     Boolean b[] = new Boolean[2]; // b[0] = null, b[1] = null
4     public static void main(String[] args){
5         TestApp t= new TestApp();
6         System.out.println(t.b[0] + " : " + t.b[1]);
7     }
8
9 }
10
11
12
13
14
15
16
17
```



18.11.2022 snippets-classrooms - Notepad

File Edit View

Q>

```
public class Test{  
    Boolean b[] = new Boolean[2];  
    public static void main(String... args){  
        Test t= new Test();  
        System.out.println(t.b[0] + ":" +t.b[1]);  
    }  
}
```

A. NullPointerException
B. false;false
C. true;true
D. null:null
E. RunTimeException other than NullPointerException

Answer: D

40