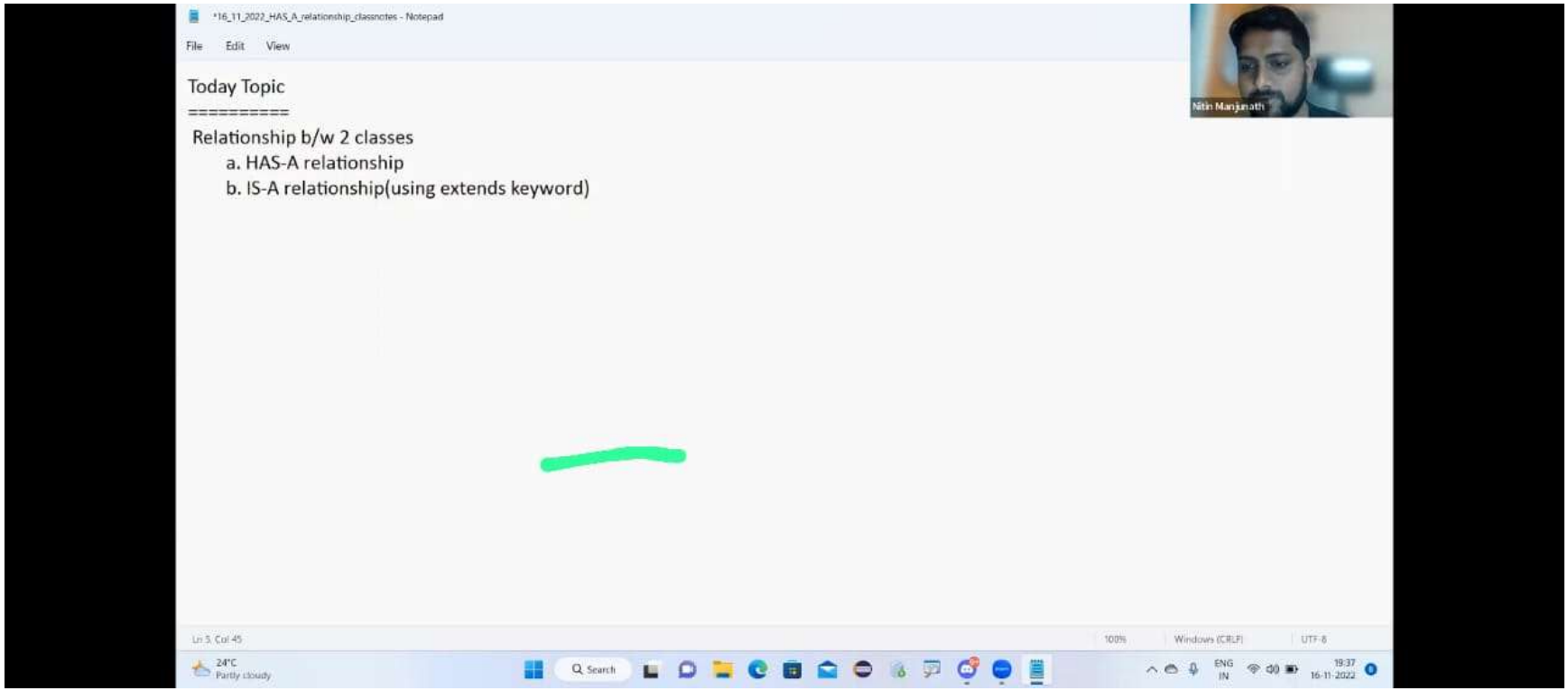
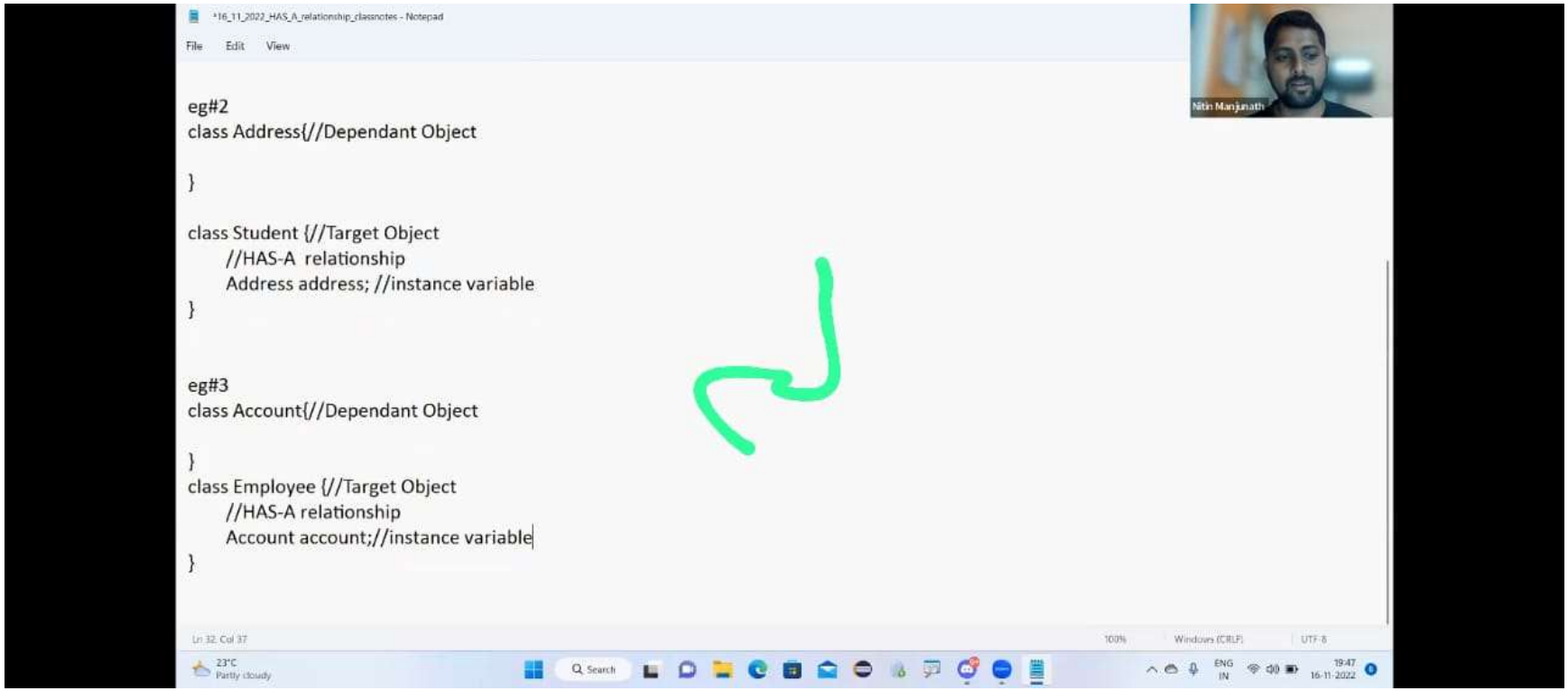
Has Relationship Dependency Injection




*16_11_2022_HAS_A_relationship_classnotes - Notepad
File Edit View
eg#2
class Address{//Dependant Object
}
class Student {//Target Object
//HAS-A relationship
Address address; //instance variable
}
eg#3
class Account{//Dependant Object
}
class Employee {//Target Object
//HAS-A relationship
Account account;//instance variable
}
Nitin Manjunath
Ln 32, Col 37
100%
Windows (CRLF)
UTF-8
23°C
Partly cloudy
Search
ENG
IN
19:47
16-11-2022

16_11_2022_HAS_A_relationship_classnotes - Notepad

File Edit View

Dependancy Injection
==========
The process of injecting dependant object into target object is called as "Dependancy injection".

eg#1
class Engine { //Dependant Object

}
class Car{//Target Object
//HAS-A relationship
Engine engine ; //instance variable
}

eg#2
class Address{//Dependant Object

}

class Student {//Target Object

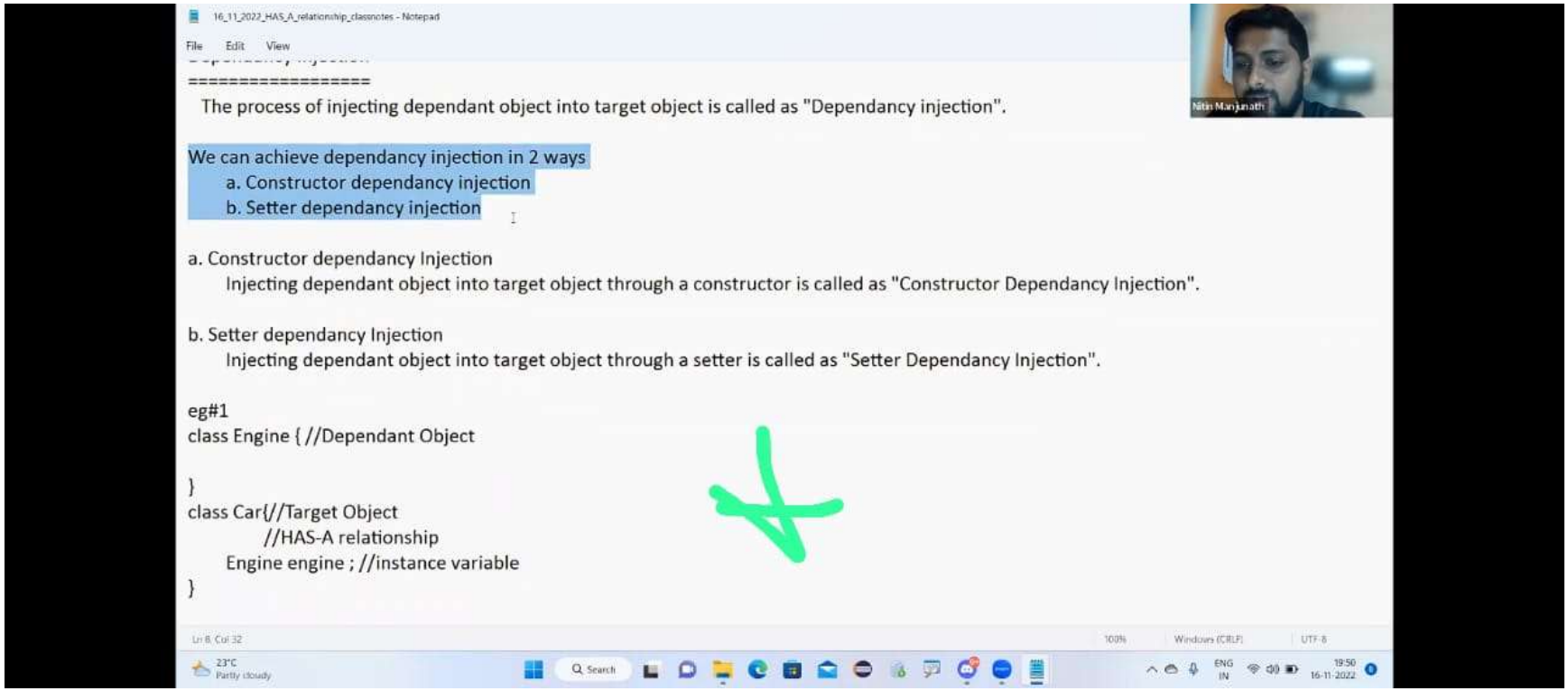
Ln 8, Col 1 100% Windows (CRLF) UTF-8

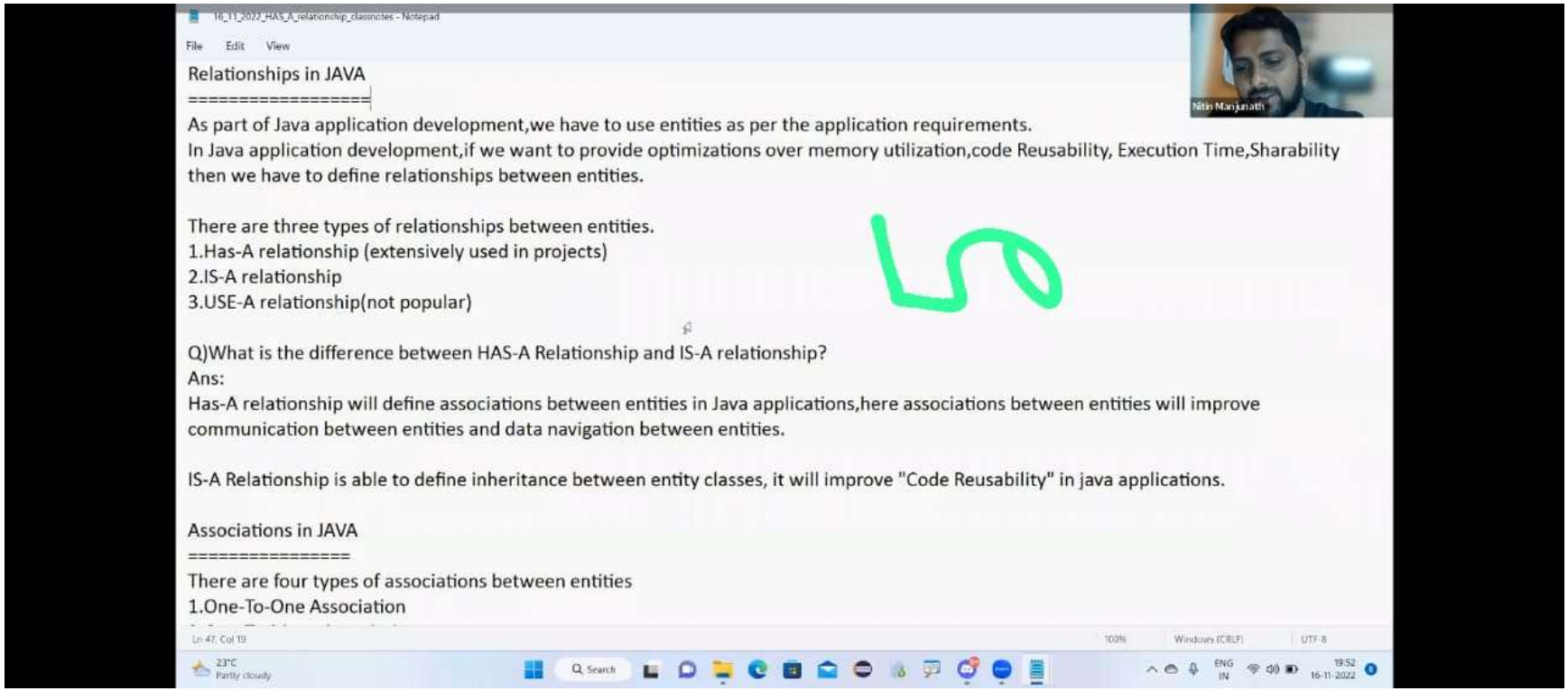
23°C Partly cloudy

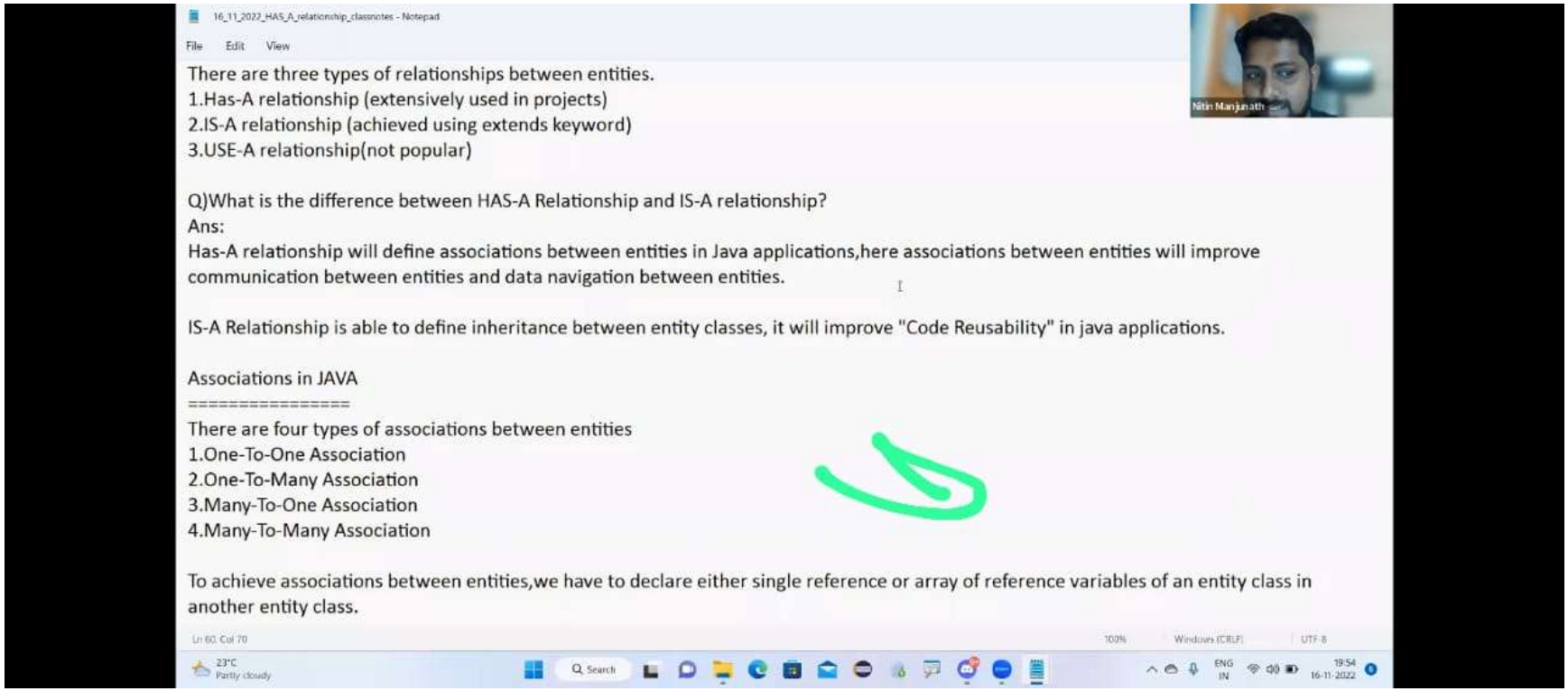
Search

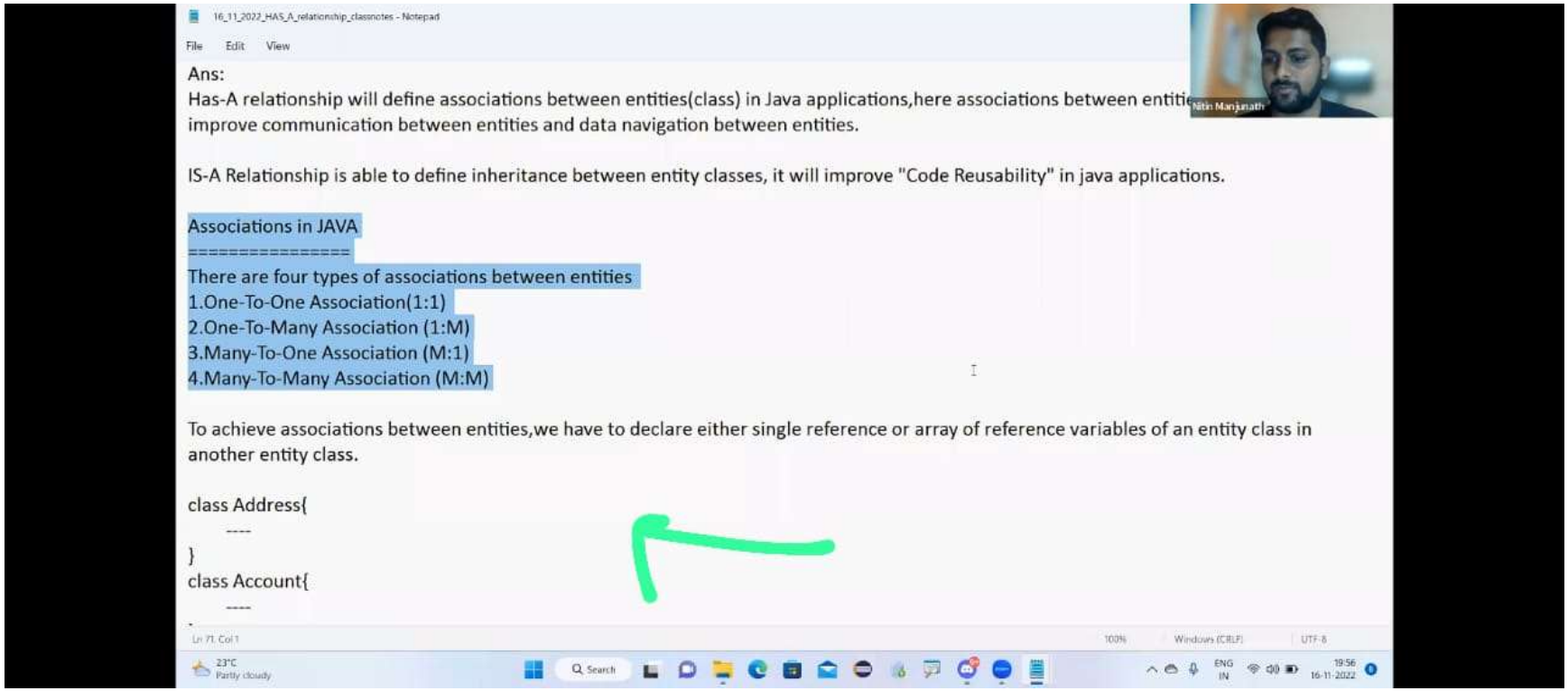
ENG IN

19:48 16-11-2022


16_11_2022_HAS_A_relationship_classnotes - Notepad
File Edit View
==========
The process of injecting dependant object into target object is called as "Dependancy injection".
We can achieve dependancy injection in 2 ways
a. Constructor dependancy injection
b. Setter dependancy injection
a. Constructor dependancy Injection
Injecting dependant object into target object through a constructor is called as "Constructor Dependancy Injection".
b. Setter dependancy Injection
Injecting dependant object into target object through a setter is called as "Setter Dependancy Injection".
eg#1
class Engine { //Dependant Object
}
class Car{//Target Object
//HAS-A relationship
Engine engine ; //instance variable
}
Nitin Manjunath
Ln 8, Col 32
100%
Windows (CRLF)
UTF-8
23°C
Partly cloudy
Search
ENG
IN
19:50
16-11-2022

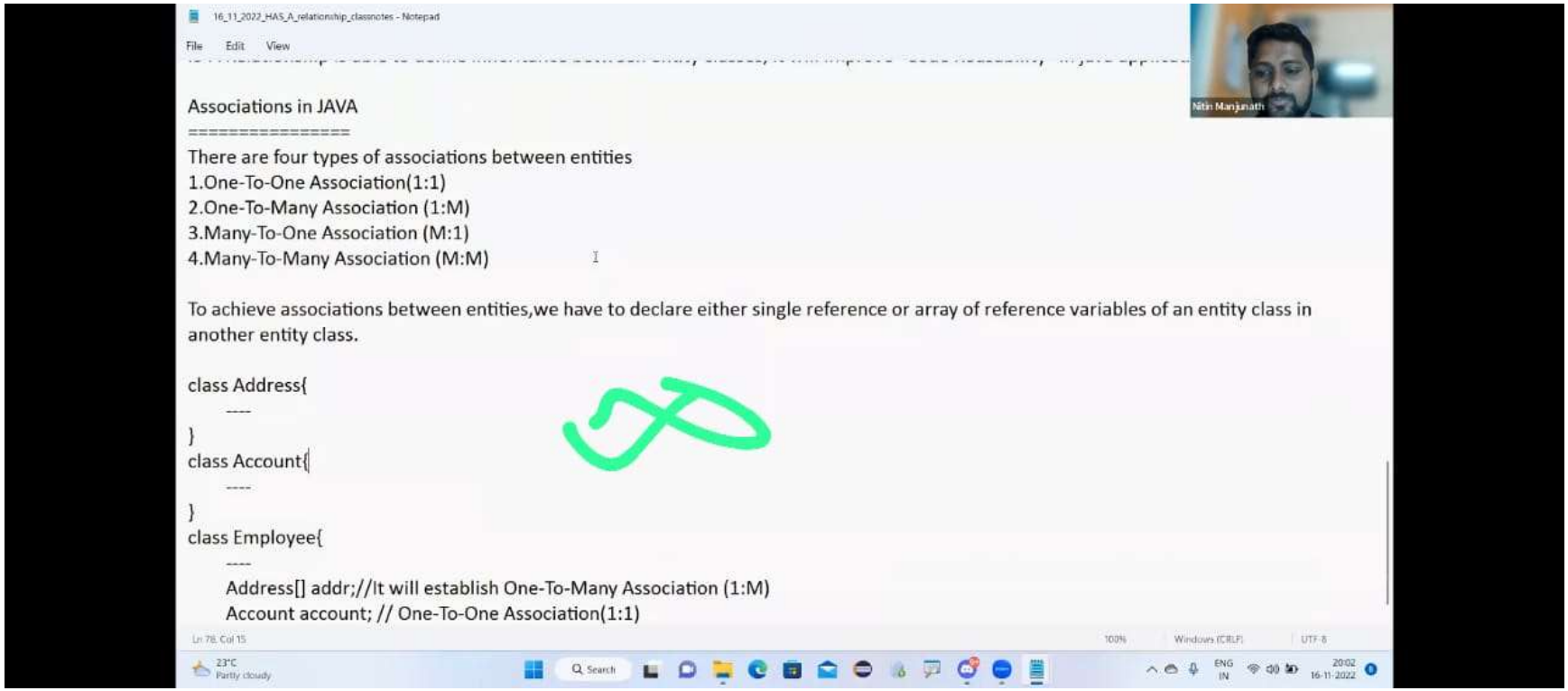

16_11_2022_HAS_A_relationship_classnotes - Notepad
File Edit View
Relationships in JAVA
==========
As part of Java application development,we have to use entities as per the application requirements.
In Java application development,if we want to provide optimizations over memory utilization,code Reusability, Execution Time,Sharability then we have to define relationships between entities.
There are three types of relationships between entities.
1.Has-A relationship (extensively used in projects)
2.IS-A relationship
3.USE-A relationship(not popular)
Q)What is the difference between HAS-A Relationship and IS-A relationship?
Ans:
Has-A relationship will define associations between entities in Java applications,here associations between entities will improve communication between entities and data navigation between entities.
IS-A Relationship is able to define inheritance between entity classes, it will improve "Code Reusability" in java applications.
Associations in JAVA
==========
There are four types of associations between entities
1.One-To-One Association
Nitin Manjunath
Ln 47, Col 19
100%
Windows (CRLF)
UTF-8
23°C
Partly cloudy
Search
ENG
IN
19:52
16-11-2022


16_11_2022_HAS_A_relationship_classnotes - Notepad
File Edit View
There are three types of relationships between entities.
1.Has-A relationship (extensively used in projects)
2.IS-A relationship (achieved using extends keyword)
3.USE-A relationship(not popular)
Q)What is the difference between HAS-A Relationship and IS-A relationship?
Ans:
Has-A relationship will define associations between entities in Java applications,here associations between entities will improve communication between entities and data navigation between entities.
IS-A Relationship is able to define inheritance between entity classes, it will improve "Code Reusability" in java applications.
Associations in JAVA
==========
There are four types of associations between entities
1.One-To-One Association
2.One-To-Many Association
3.Many-To-One Association
4.Many-To-Many Association
To achieve associations between entities,we have to declare either single reference or array of reference variables of an entity class in another entity class.
Nitin Manjunath
Ln 60, Col 70
100%
Windows (CRLF)
UTF-8
23°C
Partly cloudy
Search
ENG
IN
19:54
16-11-2022


16_11_2022_HAS_A_relationship_classnotes - Notepad
File Edit View
Ans:
Has-A relationship will define associations between entities(class) in Java applications,here associations between entitie
improve communication between entities and data navigation between entities.
IS-A Relationship is able to define inheritance between entity classes, it will improve "Code Reusability" in java applications.
Associations in JAVA
==========
There are four types of associations between entities
1.One-To-One Association(1:1)
2.One-To-Many Association (1:M)
3.Many-To-One Association (M:1)
4.Many-To-Many Association (M:M)
To achieve associations between entities,we have to declare either single reference or array of reference variables of an entity class in another entity class.
class Address{

}
class Account{

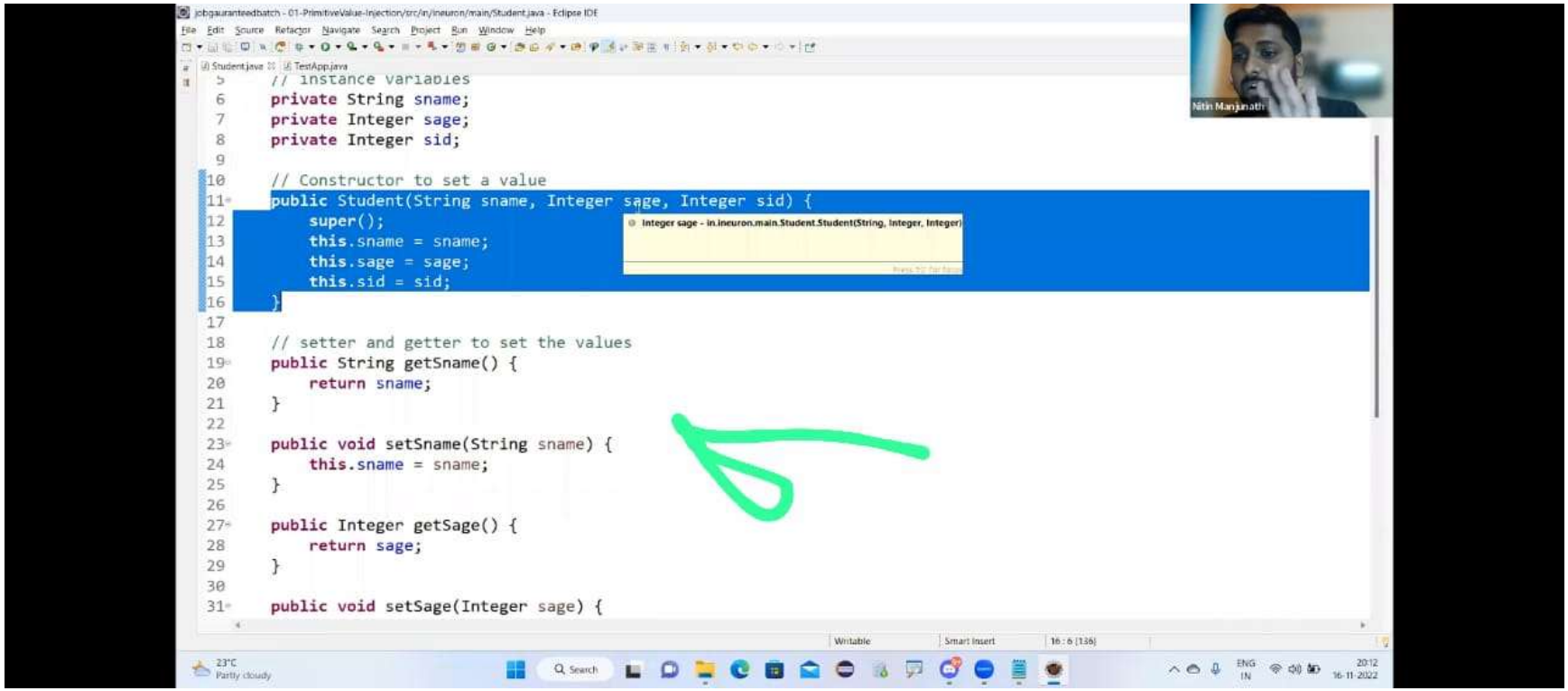
Ln 71, Col 1
100%
Windows (CRLF)
UTF-8
23°C
Partly cloudy
Search
ENG
IN
19:56
16-11-2022

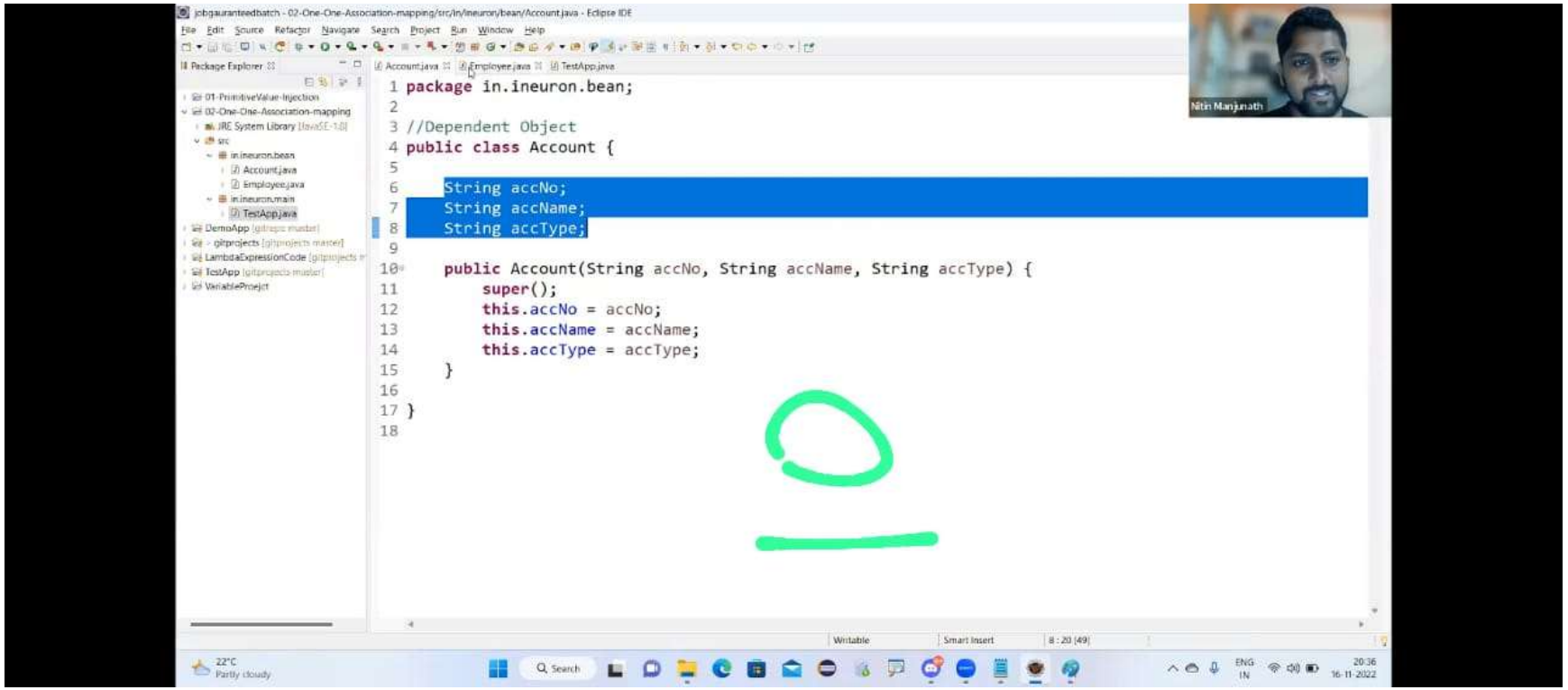

16_11_2022_HAS_A_relationship_classnotes - Notepad
File Edit View
Nitin Manjunath
Associations in JAVA
==========
There are four types of associations between entities
1.One-To-One Association(1:1)
2.One-To-Many Association (1:M)
3.Many-To-One Association (M:1)
4.Many-To-Many Association (M:M)
To achieve associations between entities,we have to declare either single reference or array of reference variables of an entity class in another entity class.
class Address{

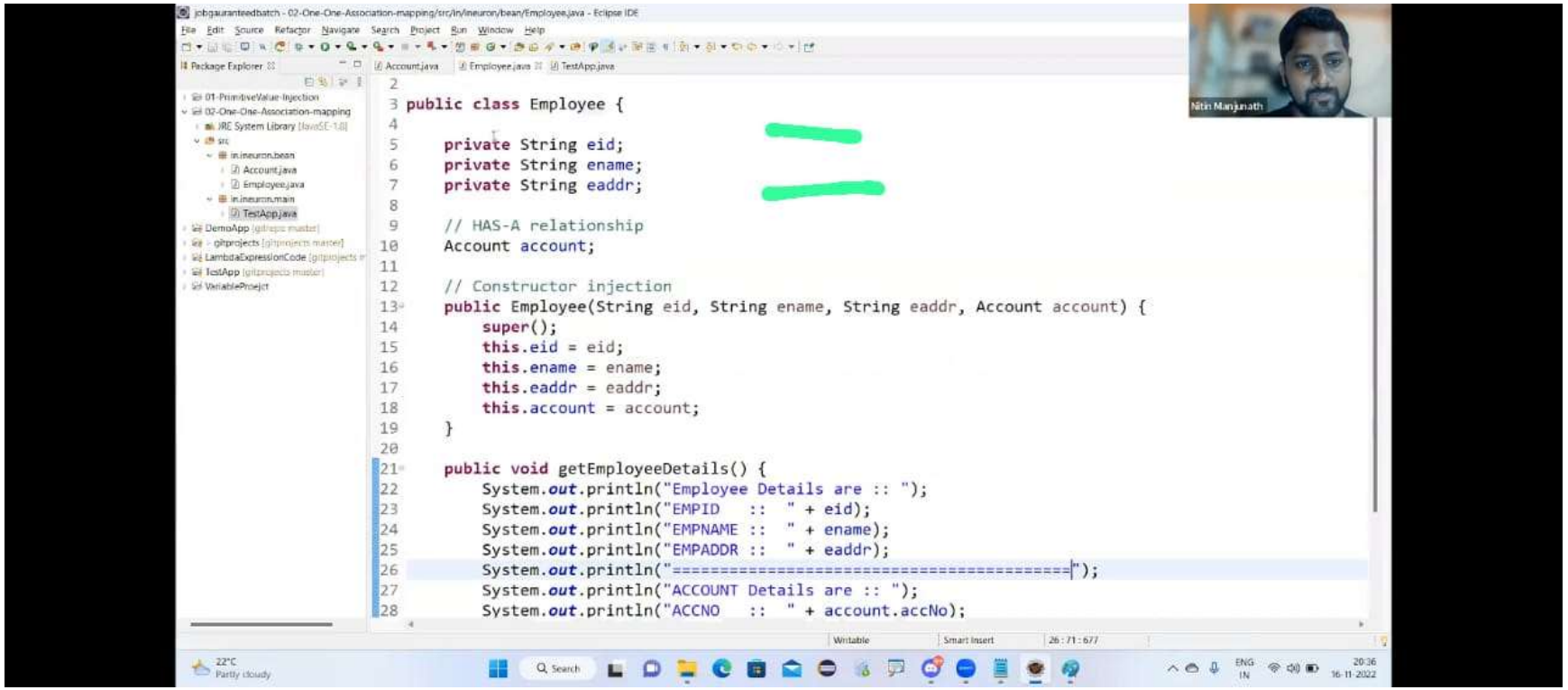
}
class Account{

}
class Employee{

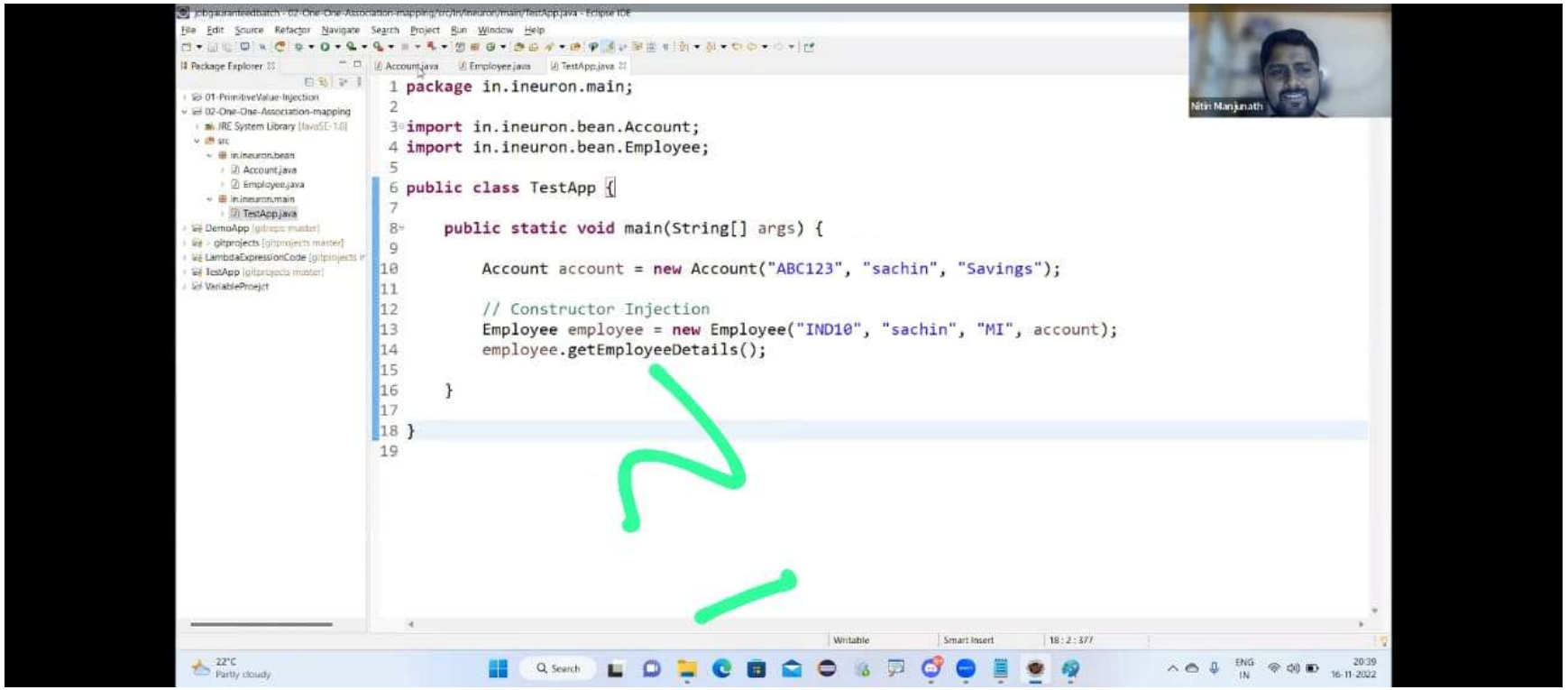
Address[] addr;//It will establish One-To-Many Association (1:M)
Account account; // One-To-One Association(1:1)
Ln 78, Col 15
100%
Windows (CRLF)
UTF-8
23°C
Partly cloudy
Search
ENG
IN
20:02
16-11-2022

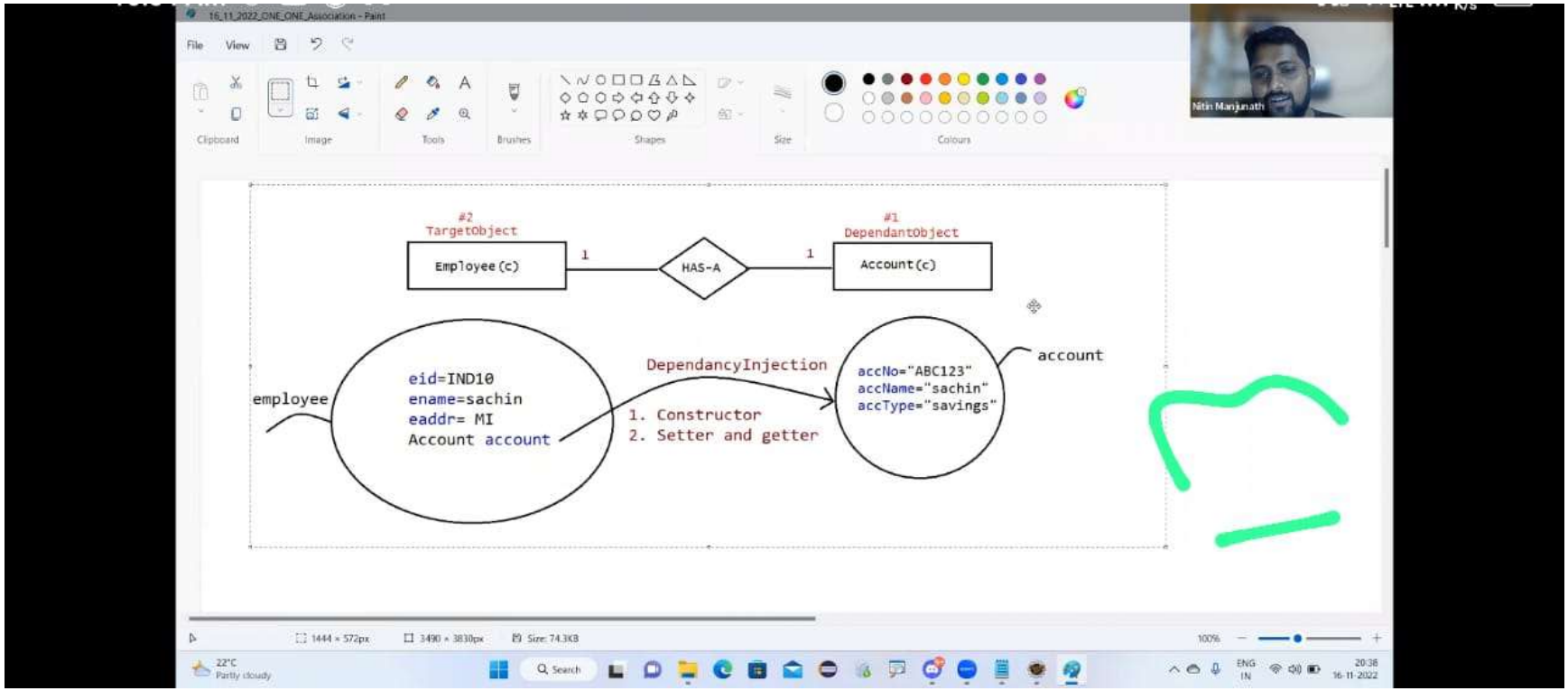

jobguaranteedbatch - 01-PrimitiveValue-Injection/src/in/ineuron/main/Student.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Student.java
TestApp.java
// instance variables
private String sname;
private Integer sage;
private Integer sid;
// Constructor to set a value
public Student(String sname, Integer sage, Integer sid) {
super();
this.sname = sname;
this.sage = sage;
this.sid = sid;
}
// setter and getter to set the values
public String getSname() {
return sname;
}
public void setSname(String sname) {
this.sname = sname;
}
public Integer getSage() {
return sage;
}
public void setSage(Integer sage) {
Integer sage - in.ineuron.main.Student.Student(String, Integer, Integer)
Nitin Manjunath
Writable
Smart Insert
16 : 6 (136)
23°C
Partly cloudy
Search
ENG
IN
20:12
16-11-2022

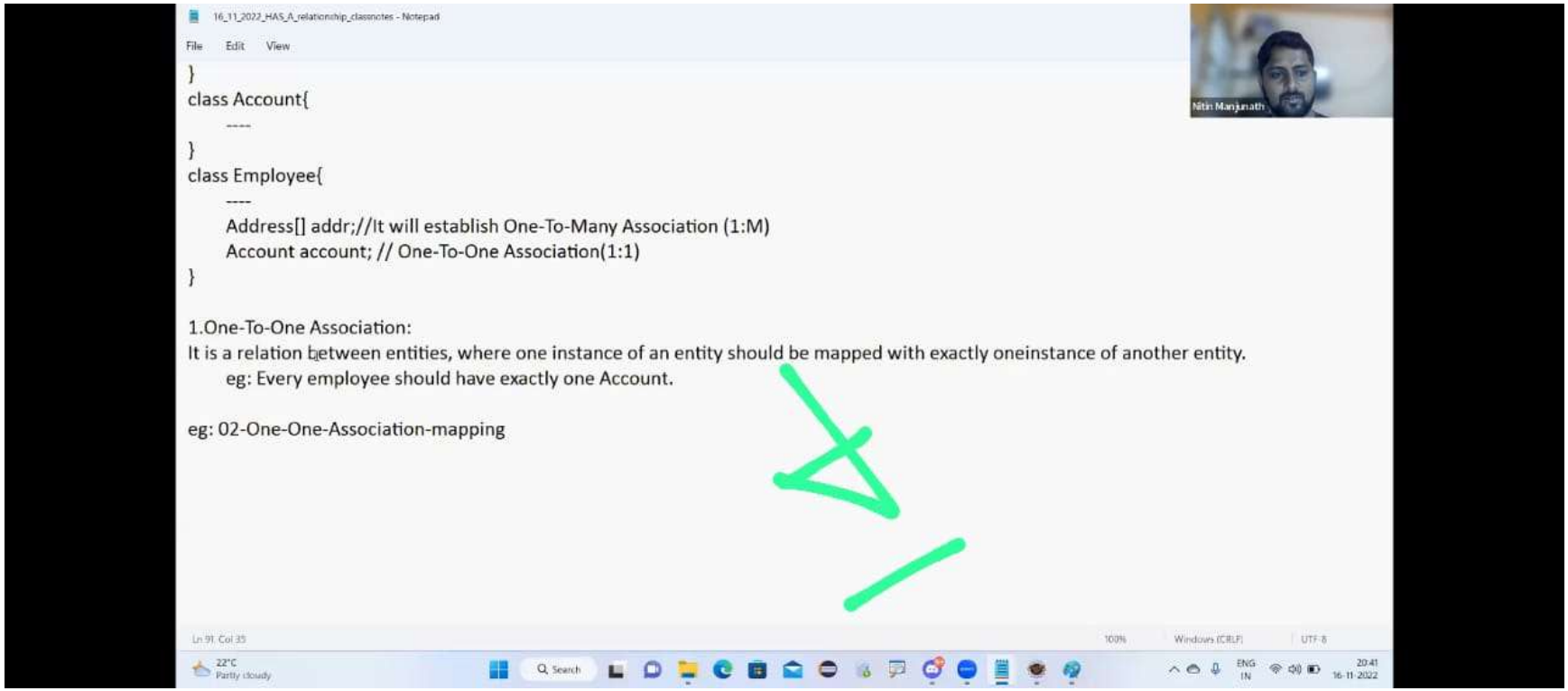

jobguaranteedbatch - 02-One-One-Association-mapping/src/in/ineuron/bean/Account.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
01-PrimitiveValue-Injection
02-One-One-Association-mapping
JRE System Library [JavaSE-1.8]
src
in.ineuron.bean
Account.java
Employee.java
in.ineuron.main
TestApp.java
DemoApp [git repo master]
gitprojects [gitprojects master]
LambdaExpressionCode [gitprojects
TestApp [gitprojects master]
VariableProject
Account.java
Employee.java
TestApp.java
1 package in.ineuron.bean;
2
3 //Dependent Object
4 public class Account {
5
6 String accNo;
7 String accName;
8 String accType;
9
10 public Account(String accNo, String accName, String accType) {
11 super();
12 this.accNo = accNo;
13 this.accName = accName;
14 this.accType = accType;
15 }
16
17 }
18
Nitin Manjunath
Writable
Smart Insert
8 : 20 [49]
22°C
Partly cloudy
Search
ENG
IN
20:36
16-11-2022




```java
public class Employee {

    private String eid;
    private String ename;
    private String eaddr;

    // HAS-A relationship
    Account account;

    // Constructor injection
    public Employee(String eid, String ename, String eaddr, Account account) {
        super();
        this.eid = eid;
        this.ename = ename;
        this.eaddr = eaddr;
        this.account = account;
    }

    public void getEmployeeDetails() {
        System.out.println("Employee Details are :: ");
        System.out.println("EMPID   ::  " + eid);
        System.out.println("EMPNAME ::  " + ename);
        System.out.println("EMPADDR ::  " + eaddr);
        System.out.println("==========================================");
        System.out.println("ACCOUNT Details are :: ");
        System.out.println("ACCNO   ::  " + account.accNo);
```

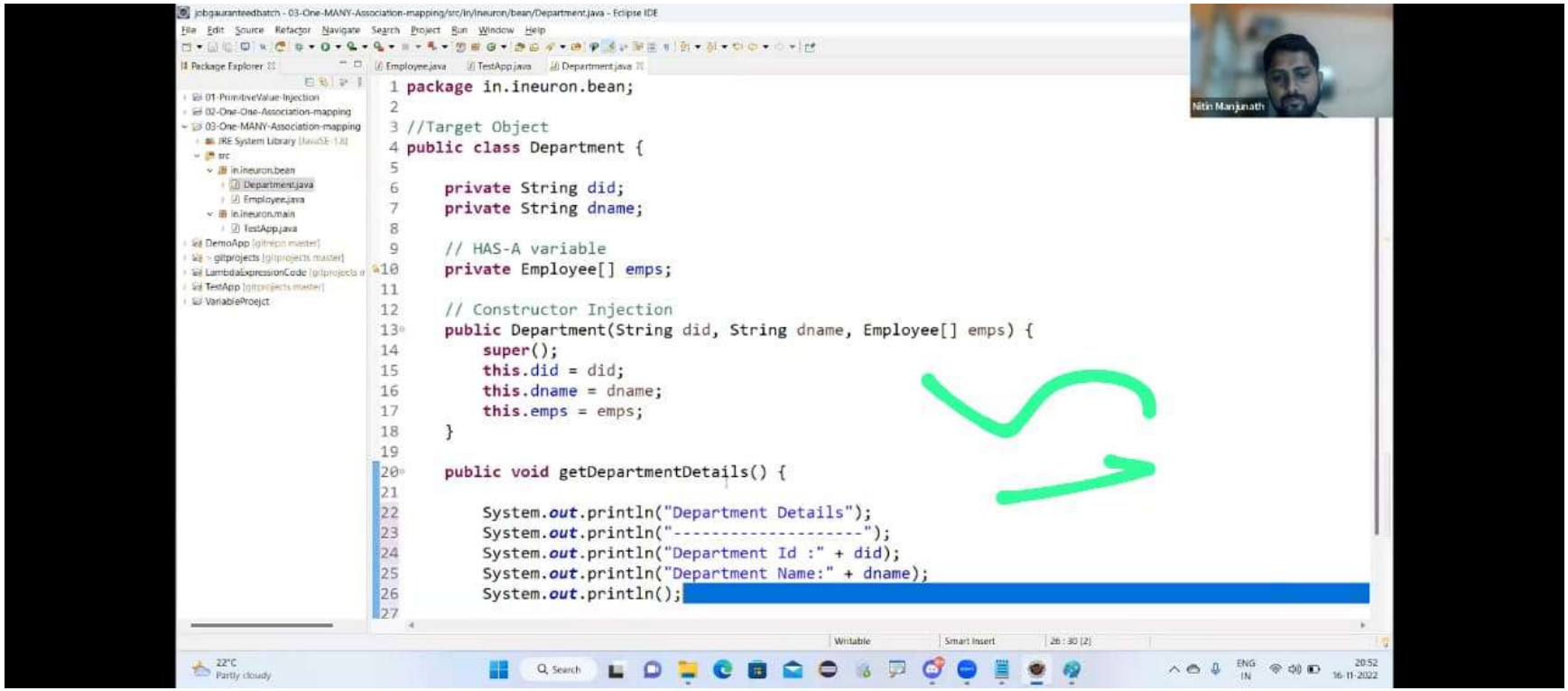

Nitin Manjunath
package in.ineuron.main;
import in.ineuron.bean.Account;
import in.ineuron.bean.Employee;
public class TestApp {
public static void main(String[] args) {
Account account = new Account("ABC123", "sachin", "Savings");
// Constructor Injection
Employee employee = new Employee("IND10", "sachin", "MI", account);
employee.getEmployeeDetails();
}
}

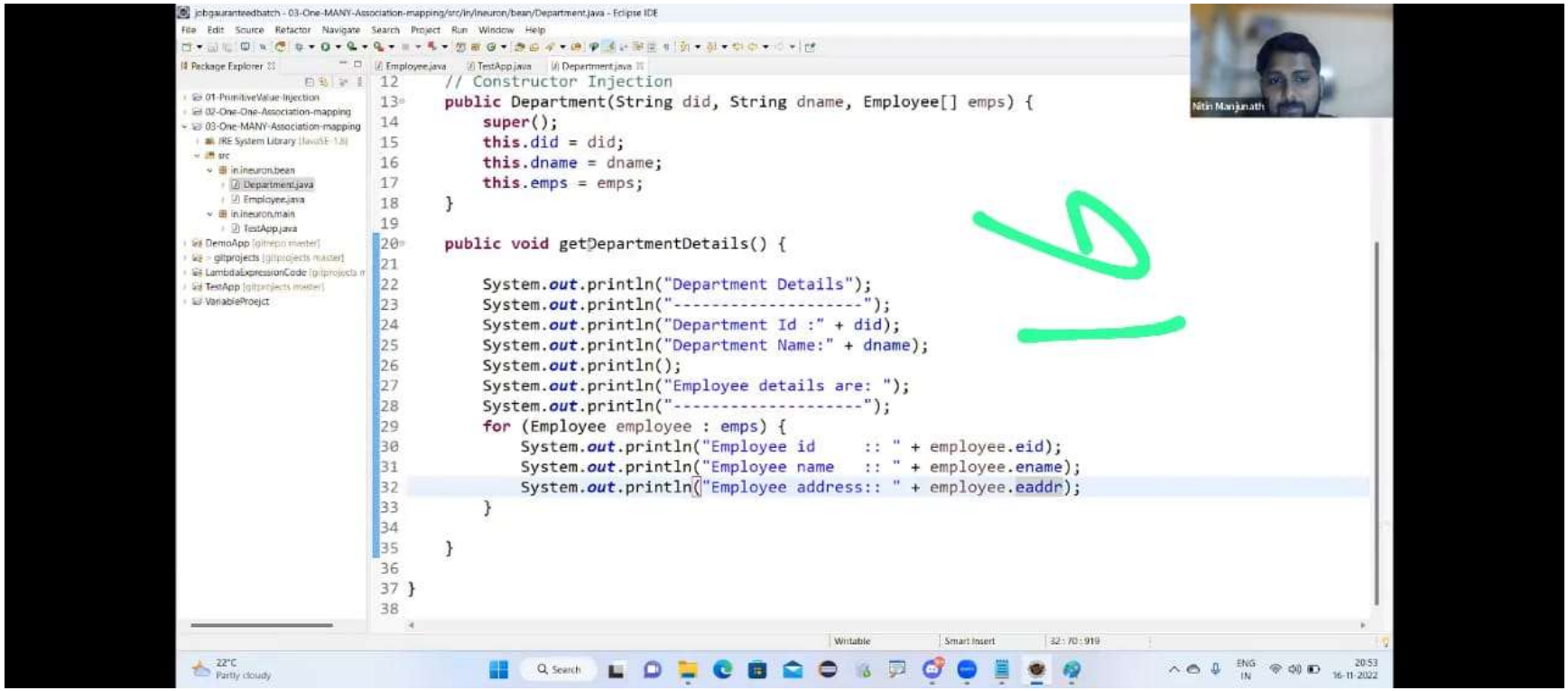

16_11_2022_ONE_ONE_Association - Paint
File
View
Clipboard
Image
Tools
Brushes
Shapes
Size
Colours
Nitin Manjunath
#2
TargetObject
Employee(c)
1
HAS-A
1
#1
DependantObject
Account(c)
employee
eid=IND10
ename=sachin
eaddr= MI
Account account
DependancyInjection
1. Constructor
2. Setter and getter
accNo="ABC123"
accName="sachin"
accType="savings"
account
1444 × 572px
3490 × 3830px
Size: 74.3KB
100%
22°C
Partly cloudy
Search
ENG
IN
20:38
16-11-2022

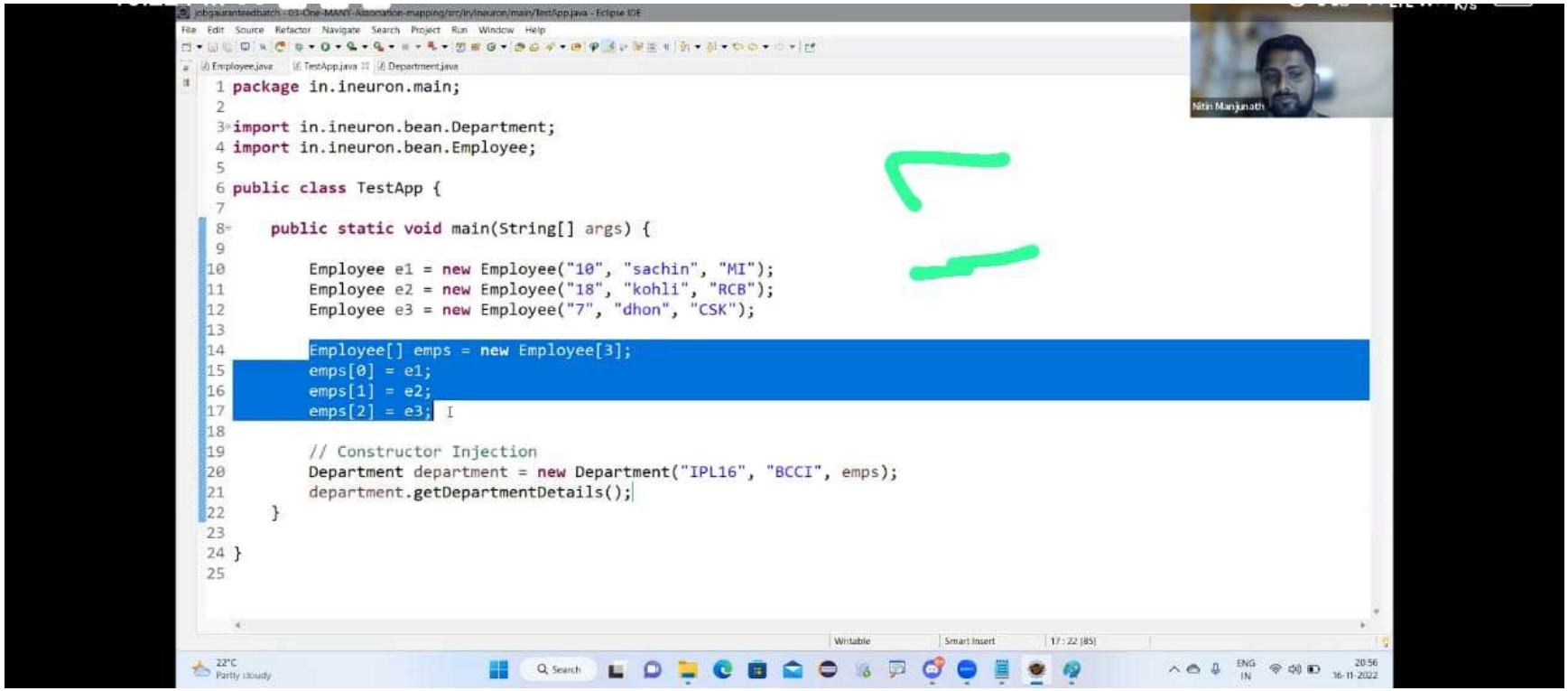

16_11_2022_HAS_A_relationship_classnotes - Notepad
File Edit View
}
class Account{

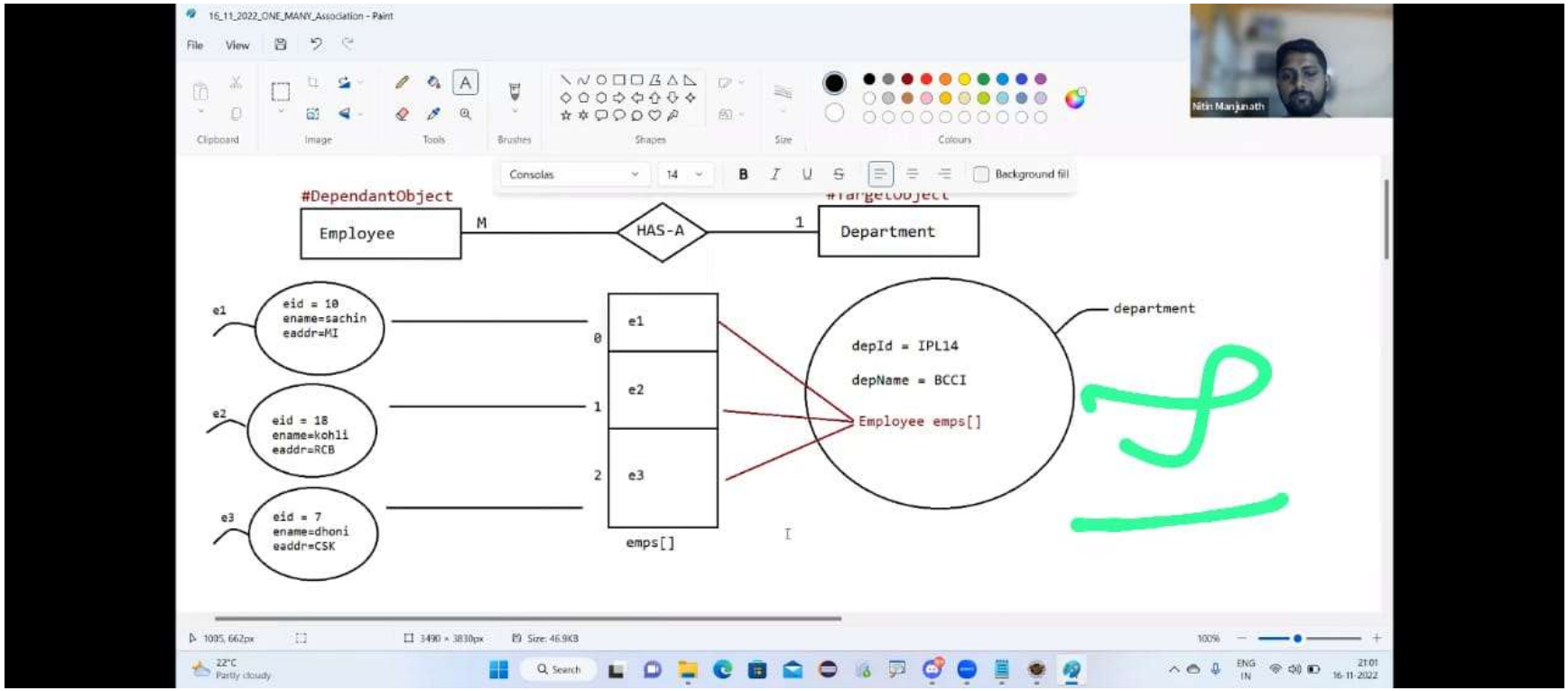
}
class Employee{

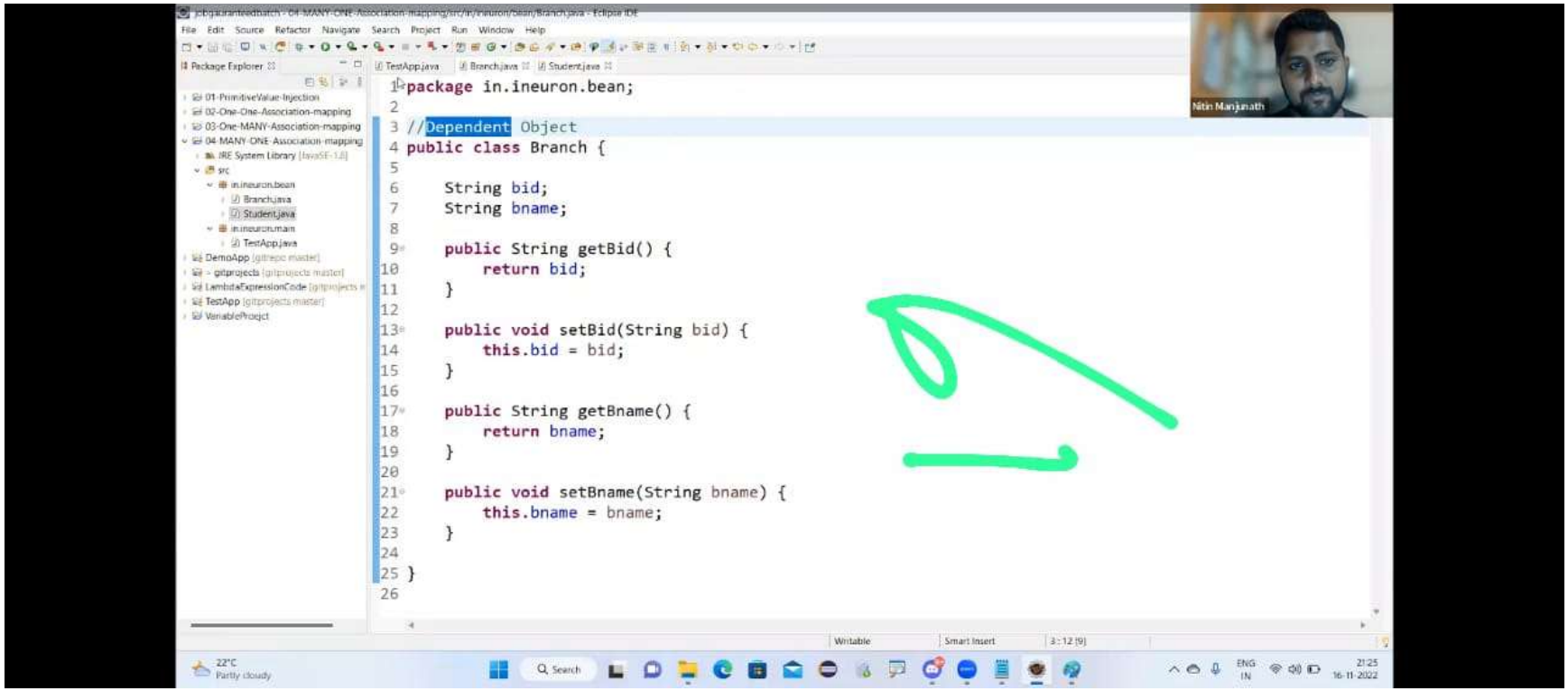
Address[] addr;//It will establish One-To-Many Association (1:M)
Account account; // One-To-One Association(1:1)
}
1.One-To-One Association:
It is a relation between entities, where one instance of an entity should be mapped with exactly oneinstance of another entity.
eg: Every employee should have exactly one Account.
eg: 02-One-One-Association-mapping
Nitin Manjunath
Ln 91, Col 35
100%
Windows (CRLF)
UTF-8
22°C
Partly cloudy
Search
ENG
IN
20:41
16-11-2022

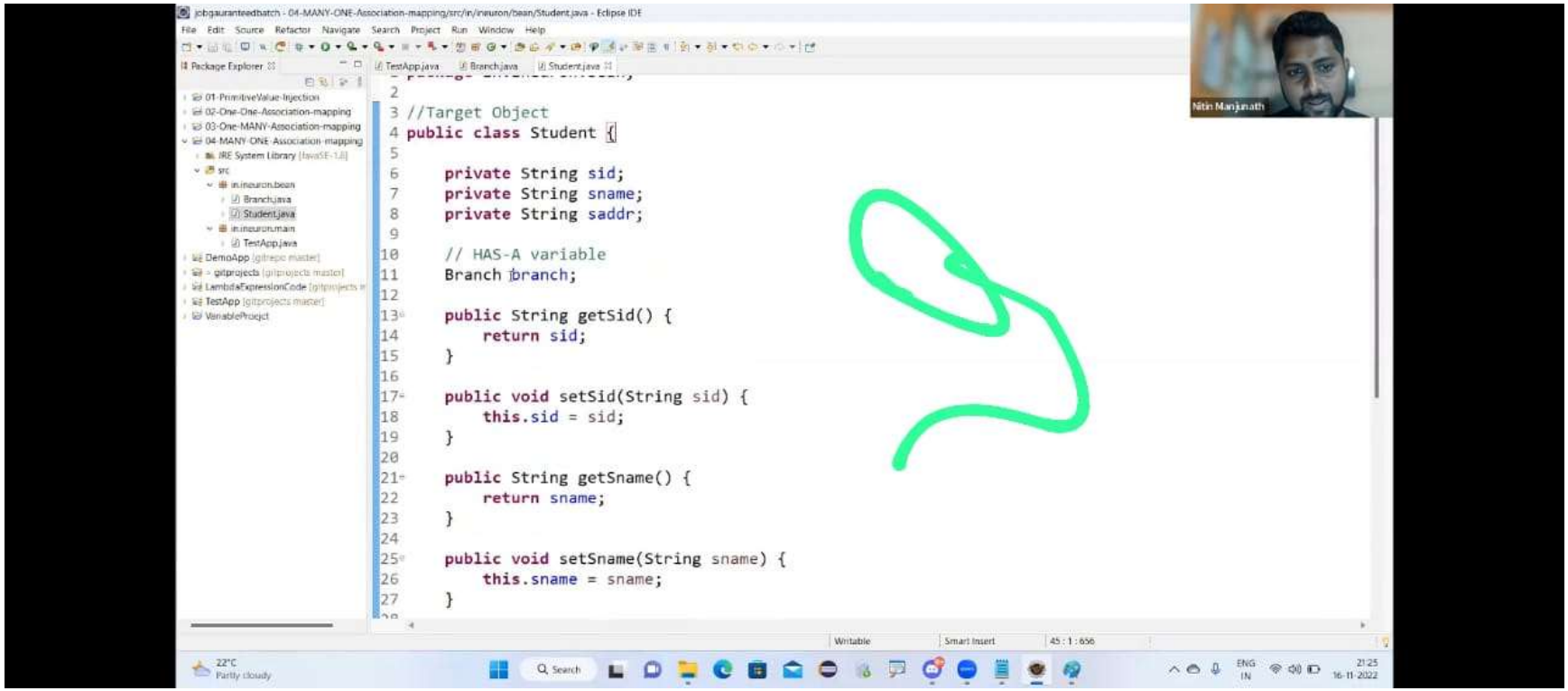

jobguaranteedbatch - 03-One-MANY-Association-mapping/src/in/ineuron/bean/Department.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
01-PrimitiveValue-Injection
02-One-One-Association-mapping
03-One-MANY-Association-mapping
JRE System Library
src
in.ineuron.bean
Department.java
Employee.java
in.ineuron.main
TestApp.java
DemoApp
gitprojects
LambdaExpressionCode
TestApp
VariableProeject
Employee.java
TestApp.java
Department.java
1 package in.ineuron.bean;
2
3 //Target Object
4 public class Department {
5
6 private String did;
7 private String dname;
8
9 // HAS-A variable
10 private Employee[] emps;
11
12 // Constructor Injection
13 public Department(String did, String dname, Employee[] emps) {
14 super();
15 this.did = did;
16 this.dname = dname;
17 this.emps = emps;
18 }
19
20 public void getDepartmentDetails() {
21
22 System.out.println("Department Details");
23 System.out.println("--------------------");
24 System.out.println("Department Id :" + did);
25 System.out.println("Department Name:" + dname);
26 System.out.println();
27
Writable
Smart Insert
26 : 30 [2]
Nitin Manjunath
22°C
Partly cloudy
Search
ENG
IN
20:52
16-11-2022

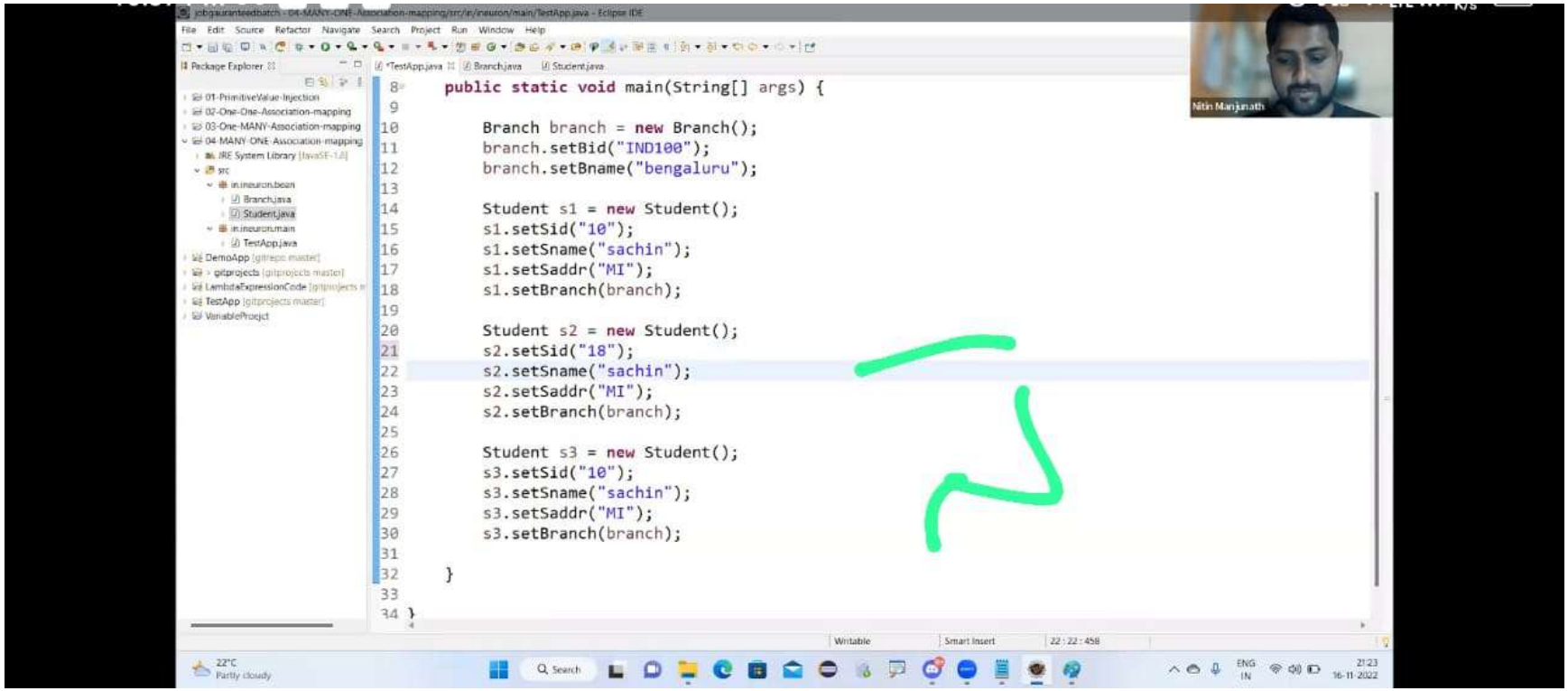

jobgauranteedbatch - 03-One-MANY-Association-mapping/src/in/ineuron/bean/Department.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Employee.java
TestApp.java
Department.java
01-PrimitiveValue-Injection
02-One-One-Association-mapping
03-One-MANY-Association-mapping
JRE System Library [JavaSE-1.8]
src
in.ineuron.bean
Department.java
Employee.java
in.ineuron.main
TestApp.java
DemoApp [git repo master]
gitprojects [gitprojects master]
LambdaExpressionCode [gitprojects
TestApp [gitprojects master]
VariableProeject
12 // Constructor Injection
13 public Department(String did, String dname, Employee[] emps) {
14 super();
15 this.did = did;
16 this.dname = dname;
17 this.emps = emps;
18 }
19
20 public void getDepartmentDetails() {
21
22 System.out.println("Department Details");
23 System.out.println("--------------------");
24 System.out.println("Department Id :" + did);
25 System.out.println("Department Name:" + dname);
26 System.out.println();
27 System.out.println("Employee details are: ");
28 System.out.println("--------------------");
29 for (Employee employee : emps) {
30 System.out.println("Employee id :: " + employee.eid);
31 System.out.println("Employee name :: " + employee.ename);
32 System.out.println("Employee address:: " + employee.eaddr);
33 }
34
35 }
36
37 }
38
Writable
Smart Insert
32 : 70 : 919
Nitin Manjunath
22°C
Partly cloudy
Search
ENG
IN
20:53
16-11-2022

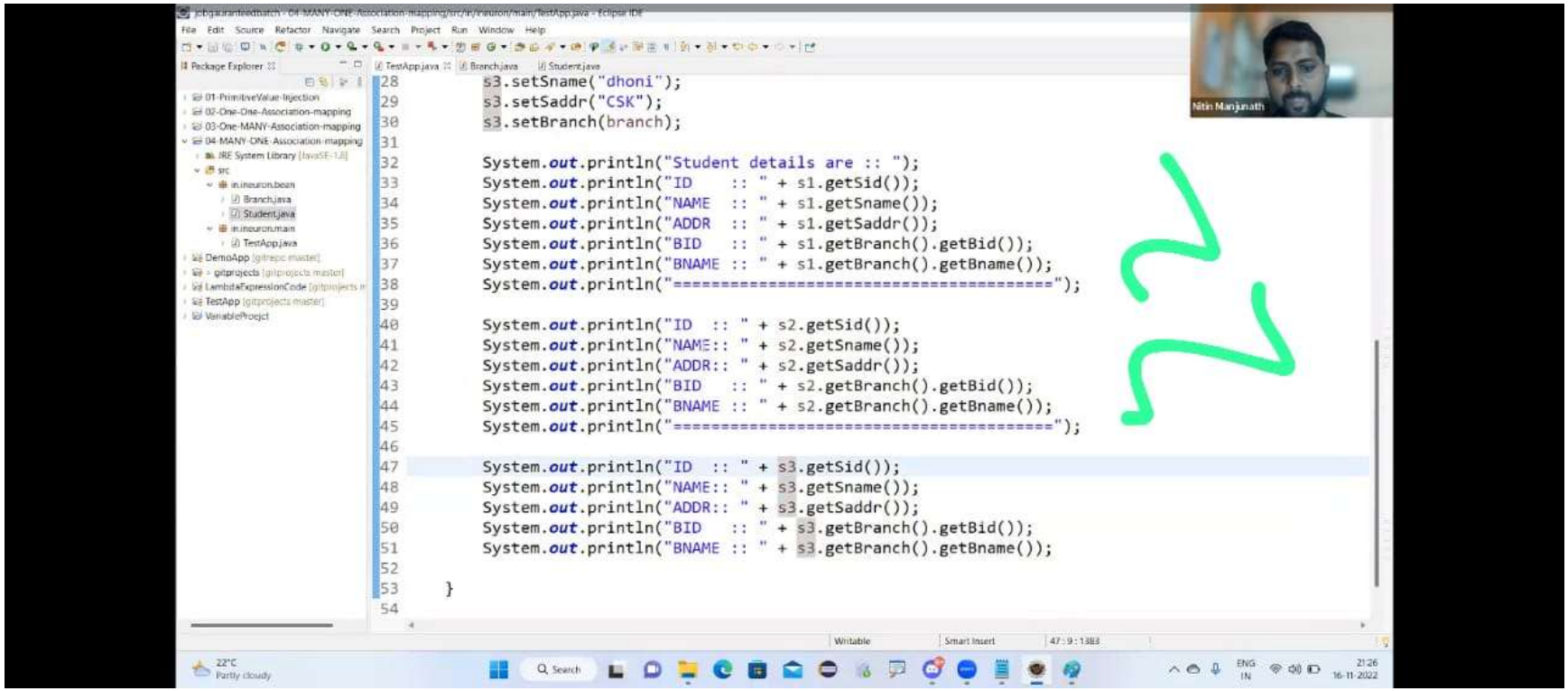

File Edit Source Refactor Navigate Search Project Run Window Help
Employee.java
TestApp.java
Department.java
1 package in.ineuron.main;
2
3 import in.ineuron.bean.Department;
4 import in.ineuron.bean.Employee;
5
6 public class TestApp {
7
8 public static void main(String[] args) {
9
10 Employee e1 = new Employee("10", "sachin", "MI");
11 Employee e2 = new Employee("18", "kohli", "RCB");
12 Employee e3 = new Employee("7", "dhon", "CSK");
13
14 Employee[] emps = new Employee[3];
15 emps[0] = e1;
16 emps[1] = e2;
17 emps[2] = e3;
18
19 // Constructor Injection
20 Department department = new Department("IPL16", "BCCI", emps);
21 department.getDepartmentDetails();
22 }
23
24 }
25
Nitin Manjunath
Writable
Smart Insert
17 : 22 [85]
22°C
Partly cloudy
Search
ENG
IN
20:56
16-11-2022

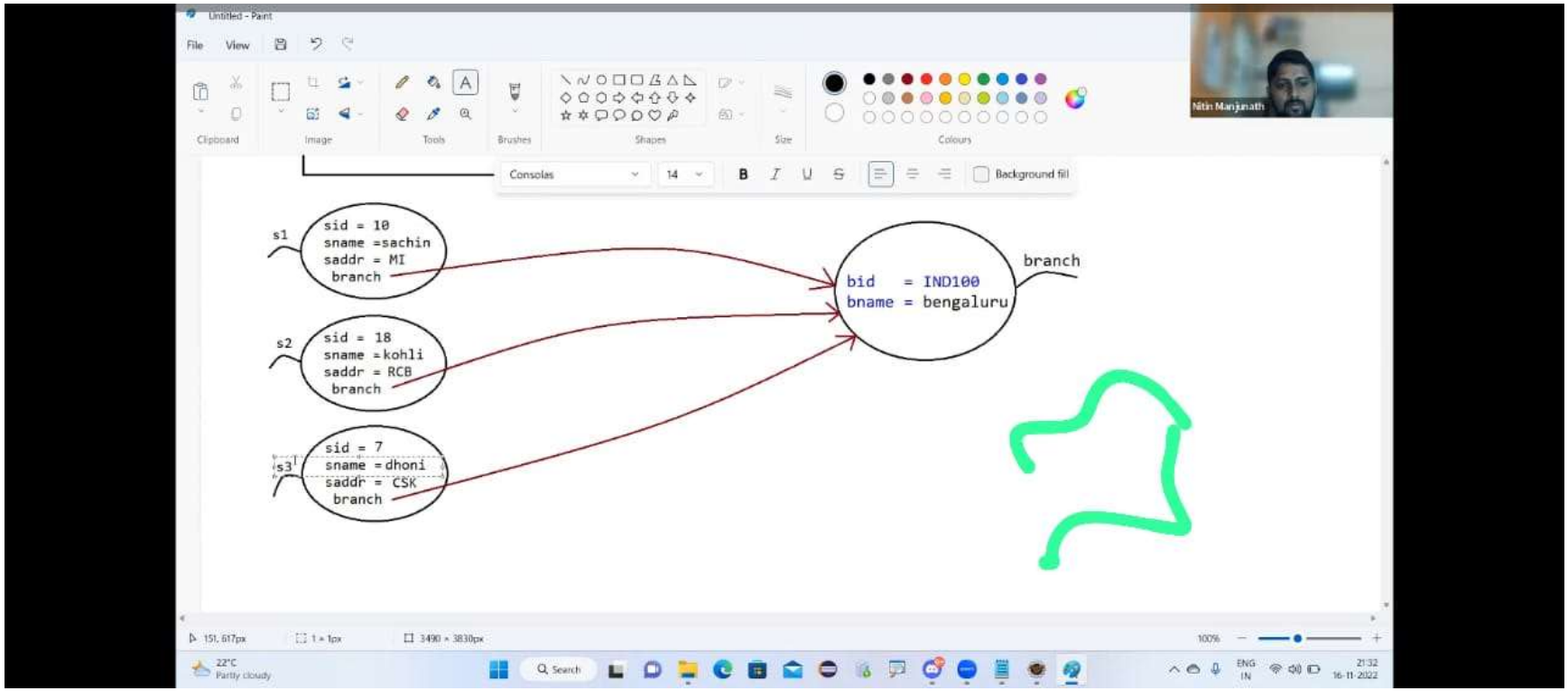

16_11_2022_ONE_MANY_Association - Paint
File
View
Nitin Manjunath
#DependantObject
Employee
M
HAS-A
1
Department
e1
eid = 10
ename=sachin
eaddr=MI
e2
eid = 18
ename=kohli
eaddr=RCB
e3
eid = 7
ename=dhoni
eaddr=CSK
0
1
2
e1
e2
e3
emps[]
department
depId = IPL14
depName = BCCI
Employee emps[]

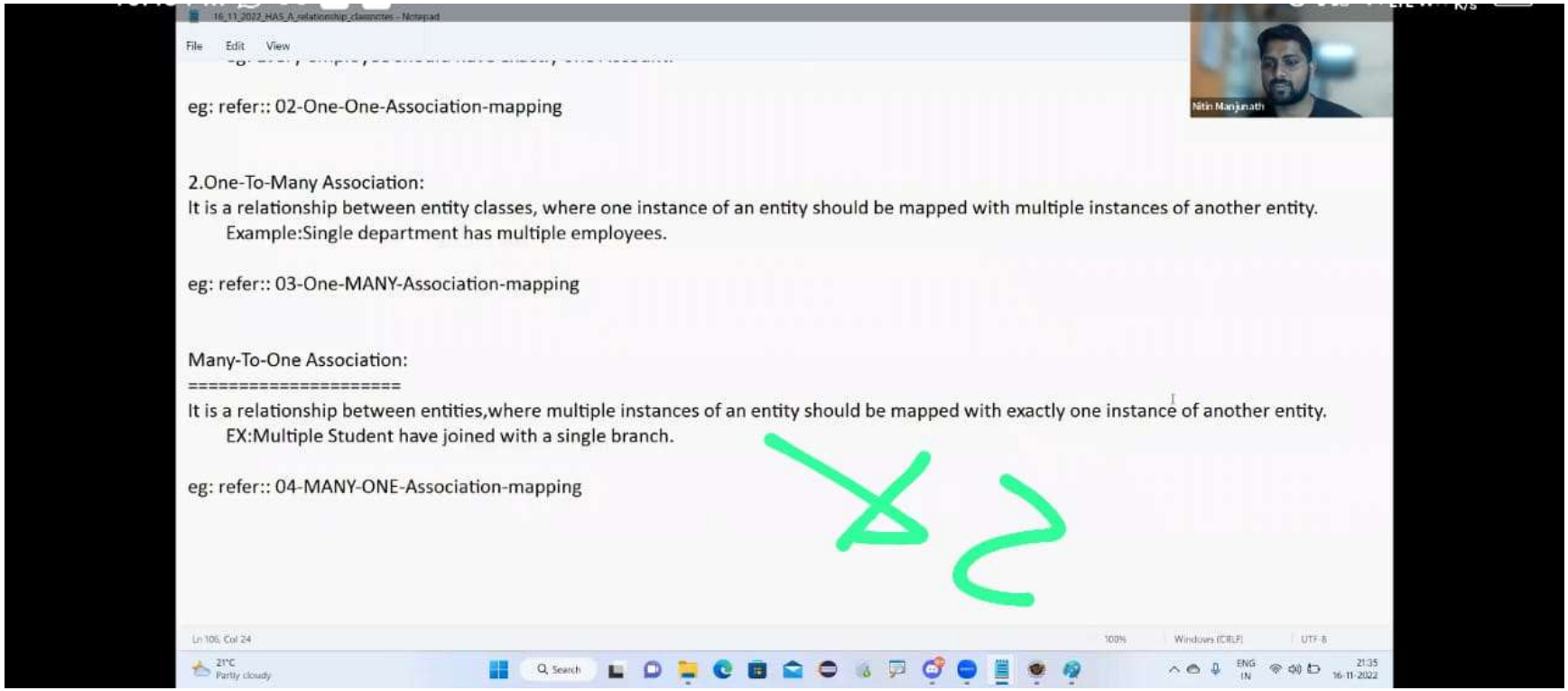

Nitin Manjunath
package in.ineuron.bean;
//Dependent Object
public class Branch {
String bid;
String bname;
public String getBid() {
return bid;
}
public void setBid(String bid) {
this.bid = bid;
}
public String getBname() {
return bname;
}
public void setBname(String bname) {
this.bname = bname;
}
}

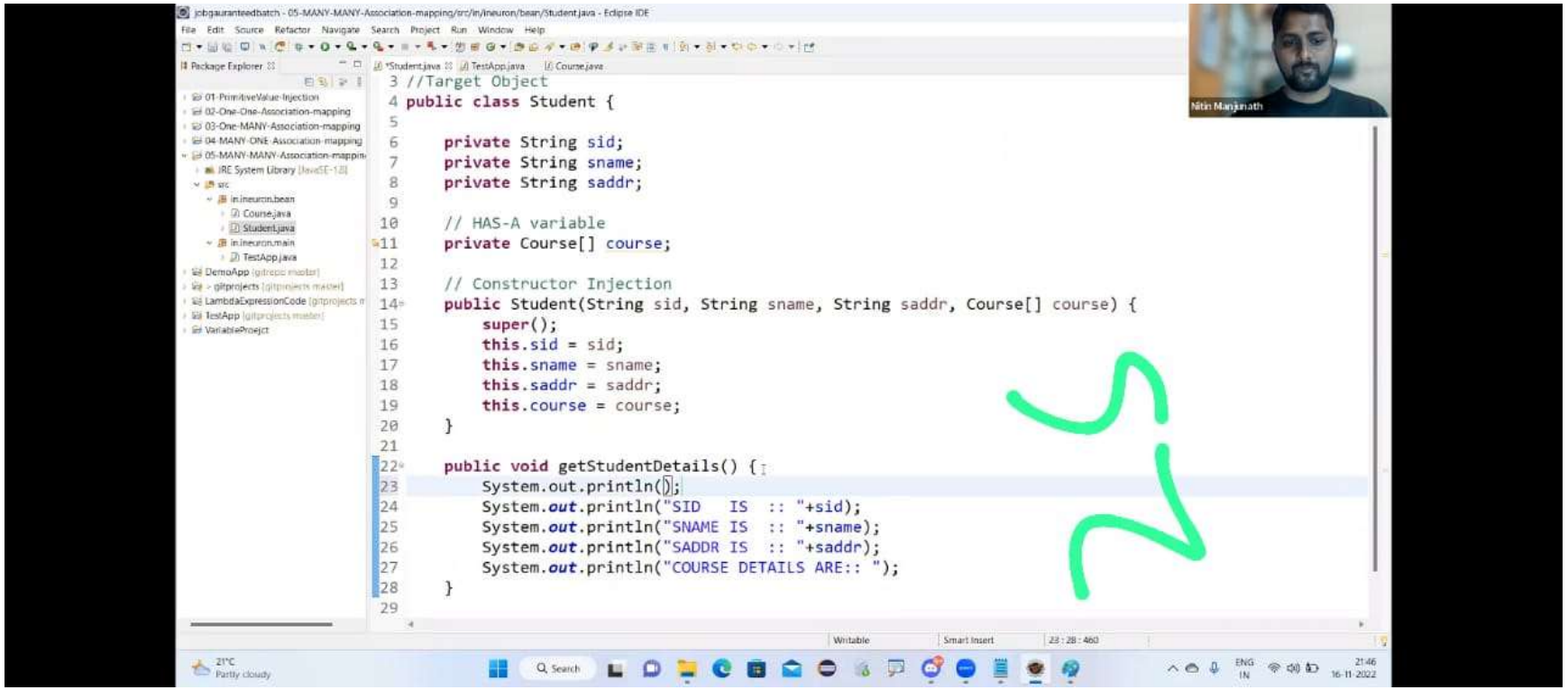

jobguaranteedbatch - 04-MANY-ONE-Association-mapping/src/in/ineuron/bean/Student.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
01-PrimitiveValue-Injection
02-One-One-Association-mapping
03-One-MANY-Association-mapping
04-MANY-ONE-Association-mapping
JRE System Library [JavaSE-1.8]
src
in.ineuron.bean
Branch.java
Student.java
in.ineuron.main
TestApp.java
DemoApp [git repo master]
gitprojects [gitprojects master]
LambdaExpressionCode [gitprojects
TestApp [gitprojects master]
VariableProject
TestApp.java
Branch.java
Student.java
1 package in.ineuron.bean;
2
3 //Target Object
4 public class Student {
5
6 private String sid;
7 private String sname;
8 private String saddr;
9
10 // HAS-A variable
11 Branch branch;
12
13 public String getSid() {
14 return sid;
15 }
16
17 public void setSid(String sid) {
18 this.sid = sid;
19 }
20
21 public String getSname() {
22 return sname;
23 }
24
25 public void setSname(String sname) {
26 this.sname = sname;
27 }
Nitin Manjunath
Writable
Smart Insert
45 : 1 : 656
22°C
Partly cloudy
Search
ENG
IN
21:25
16-11-2022

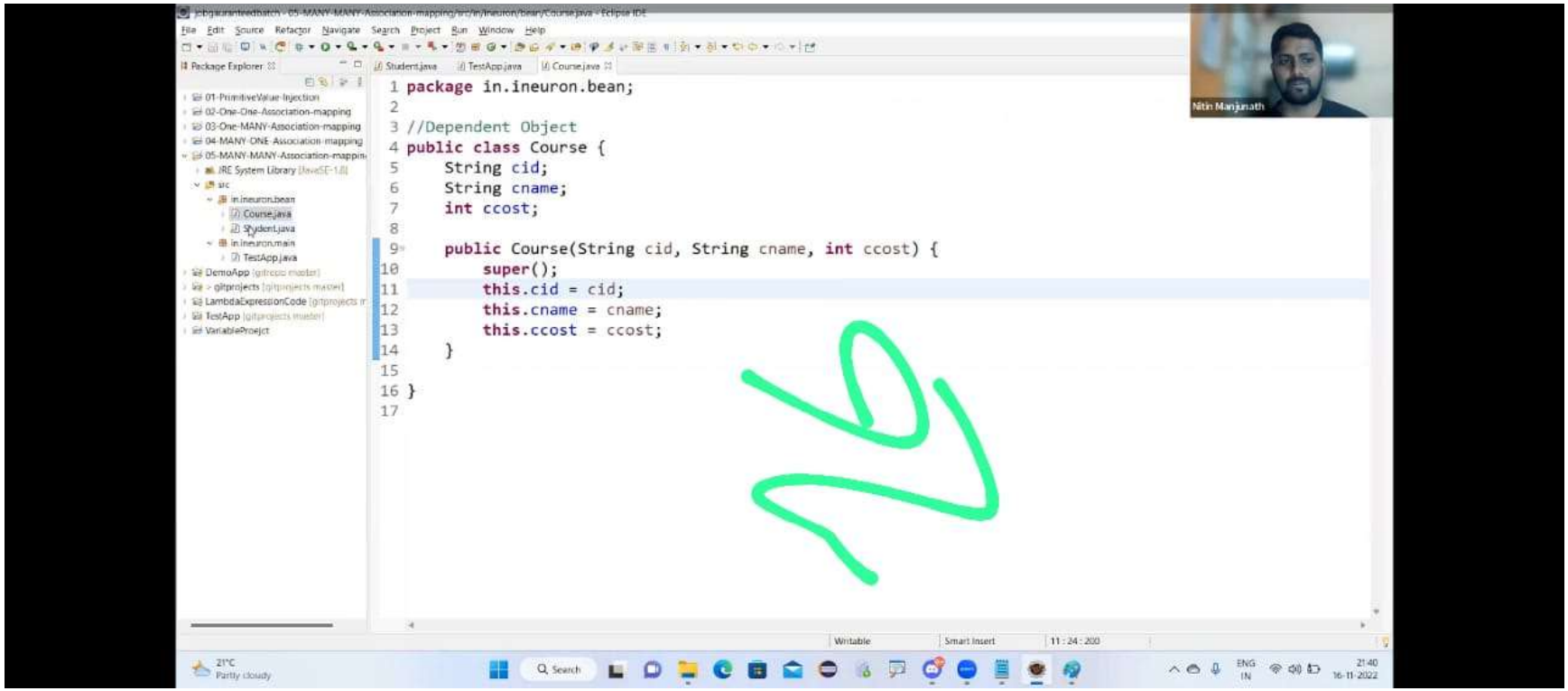

File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
01-PrimitiveValue-Injection
02-One-One-Association-mapping
03-One-MANY-Association-mapping
04-MANY-ONE-Association-mapping
JRE System Library [JavaSE-1.8]
src
in.ineuron.bean
Branch.java
Student.java
in.ineuron.main
TestApp.java
DemoApp [gitrepo master]
gitprojects [gitprojects master]
LambdaExpressionCode [gitprojects m
TestApp [gitprojects master]
VariableProject
*TestApp.java
Branch.java
Student.java
public static void main(String[] args) {
Branch branch = new Branch();
branch.setBid("IND100");
branch.setBname("bengaluru");
Student s1 = new Student();
s1.setSid("10");
s1.setSname("sachin");
s1.setSaddr("MI");
s1.setBranch(branch);
Student s2 = new Student();
s2.setSid("18");
s2.setSname("sachin");
s2.setSaddr("MI");
s2.setBranch(branch);
Student s3 = new Student();
s3.setSid("10");
s3.setSname("sachin");
s3.setSaddr("MI");
s3.setBranch(branch);
}
}
Writable
Smart Insert
22 : 22 : 458
Nitin Manjunath
22°C
Partly cloudy
Search
ENG
IN
21:23
16-11-2022

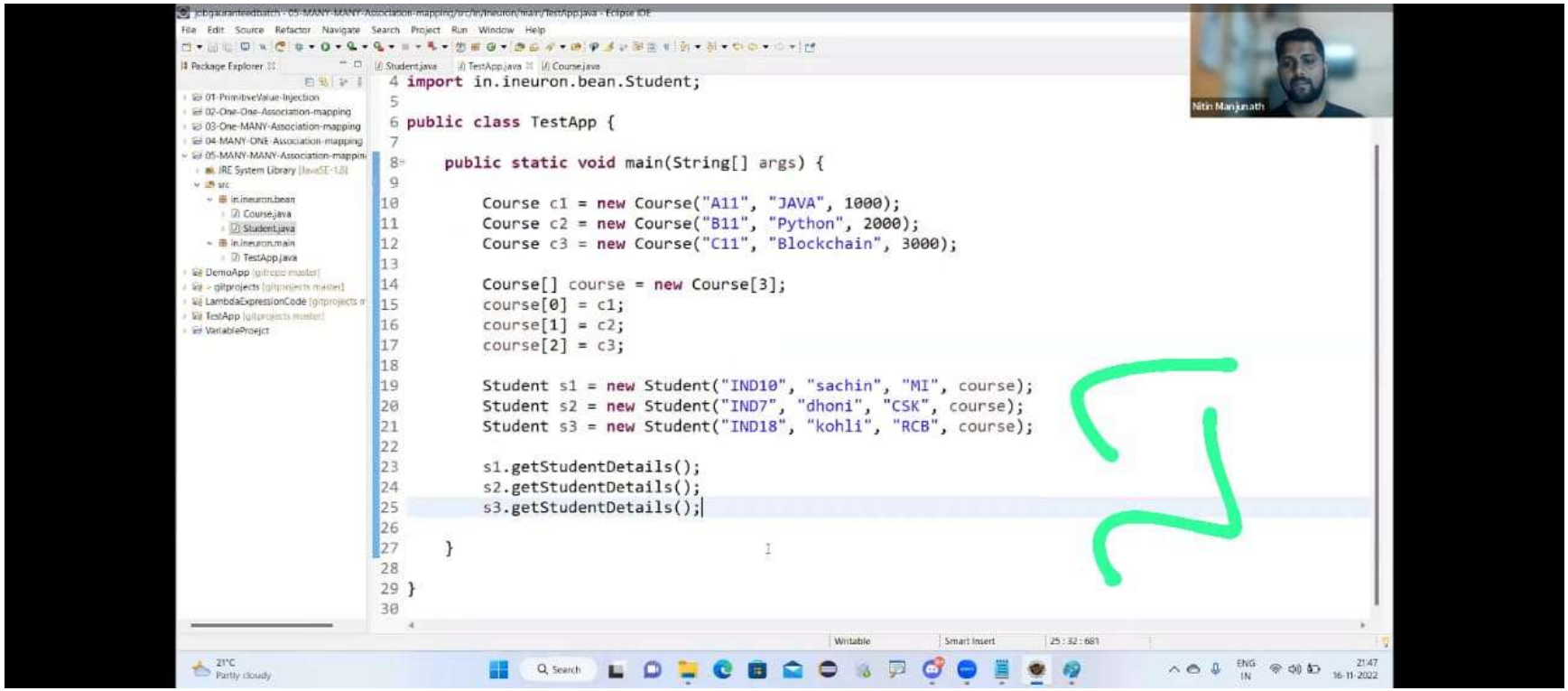

Nitin Manjunath
s3.setSname("dhoni");
s3.setSaddr("CSK");
s3.setBranch(branch);
System.out.println("Student details are :: ");
System.out.println("ID :: " + s1.getSid());
System.out.println("NAME :: " + s1.getSname());
System.out.println("ADDR :: " + s1.getSaddr());
System.out.println("BID :: " + s1.getBranch().getBid());
System.out.println("BNAME :: " + s1.getBranch().getBname());
System.out.println("==");
System.out.println("ID :: " + s2.getSid());
System.out.println("NAME:: " + s2.getSname());
System.out.println("ADDR:: " + s2.getSaddr());
System.out.println("BID :: " + s2.getBranch().getBid());
System.out.println("BNAME :: " + s2.getBranch().getBname());
System.out.println("==");
System.out.println("ID :: " + s3.getSid());
System.out.println("NAME:: " + s3.getSname());
System.out.println("ADDR:: " + s3.getSaddr());
System.out.println("BID :: " + s3.getBranch().getBid());
System.out.println("BNAME :: " + s3.getBranch().getBname());
}

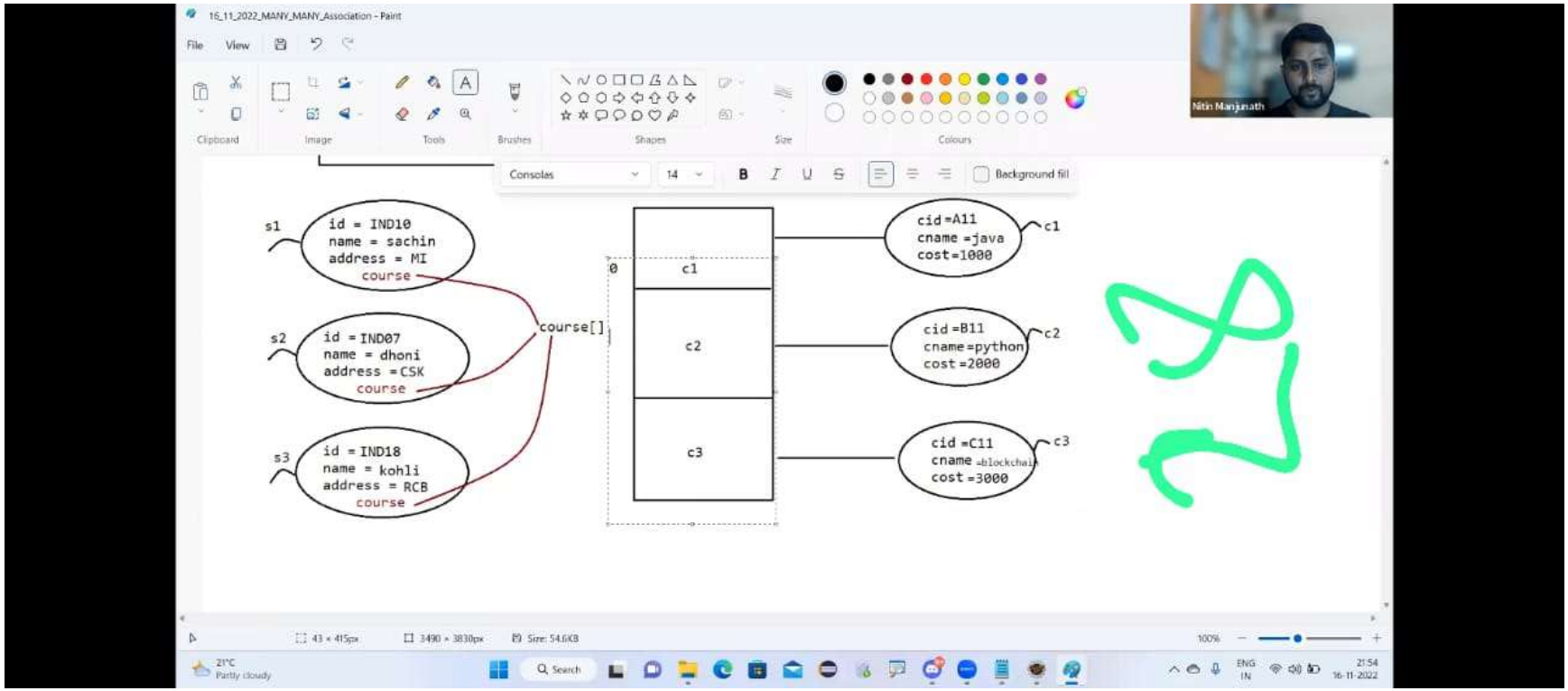

Untitled - Paint
File
View
Clipboard
Image
Tools
Brushes
Shapes
Size
Colours
Nitin Manjunath
Consolas
14
Background fill
s1
sid = 10
sname =sachin
saddr = MI
branch
s2
sid = 18
sname =kohli
saddr = RCB
branch
s3
sid = 7
sname =dhoni
saddr = CSK
branch
bid = IND100
bname = bengaluru
branch
151, 617px
1 × 1px
3490 × 3830px
100%
22°C
Partly cloudy
Search
ENG
IN
21:32
16-11-2022

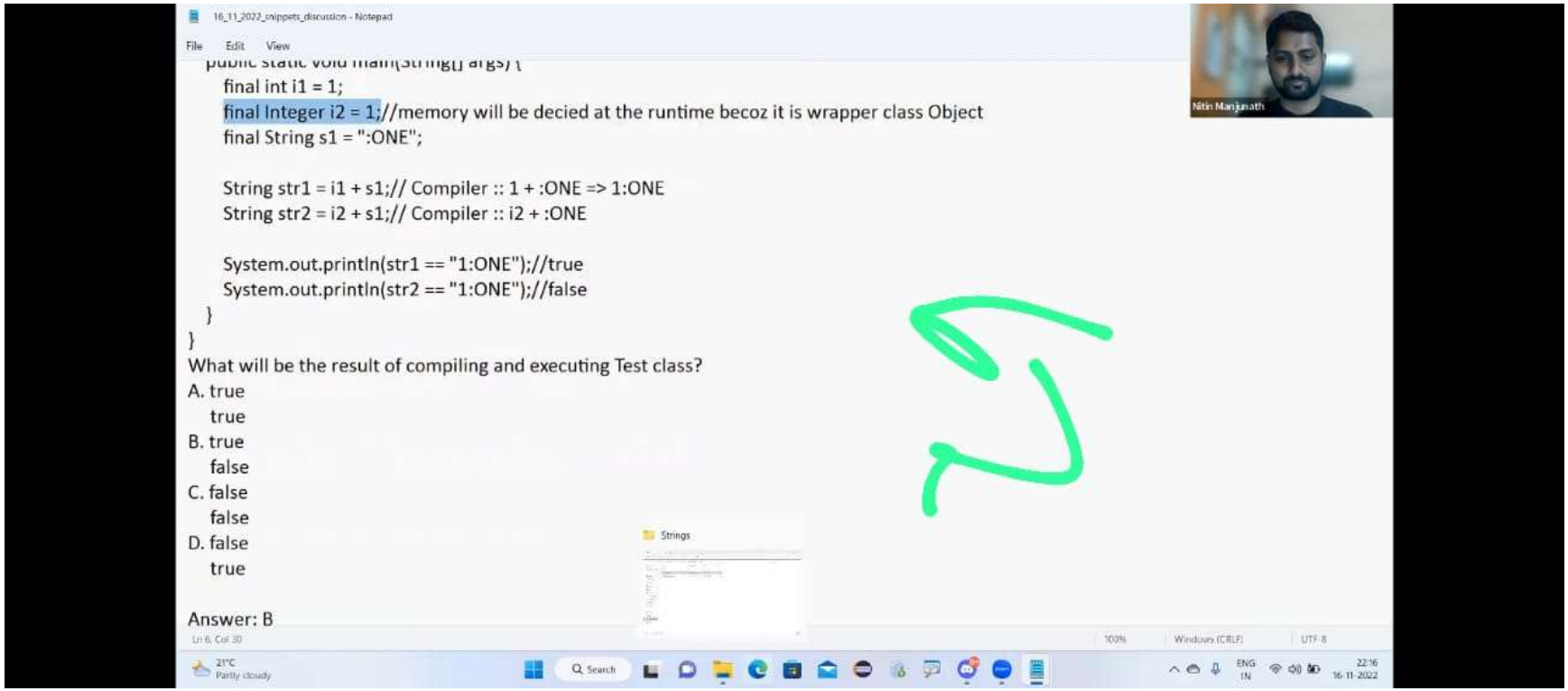

16_11_2022_HAS_A_relationship_classnotes - Notepad
File Edit View
eg: refer:: 02-One-One-Association-mapping
2.One-To-Many Association:
It is a relationship between entity classes, where one instance of an entity should be mapped with multiple instances of another entity.
Example:Single department has multiple employees.
eg: refer:: 03-One-MANY-Association-mapping
Many-To-One Association:
==========
It is a relationship between entities,where multiple instances of an entity should be mapped with exactly one instance of another entity.
EX:Multiple Student have joined with a single branch.
eg: refer:: 04-MANY-ONE-Association-mapping
Nitin Manjunath
Ln 106, Col 24
100%
Windows (CRLF)
UTF-8
21°C
Partly cloudy
Search
ENG
IN
21:35
16-11-2022

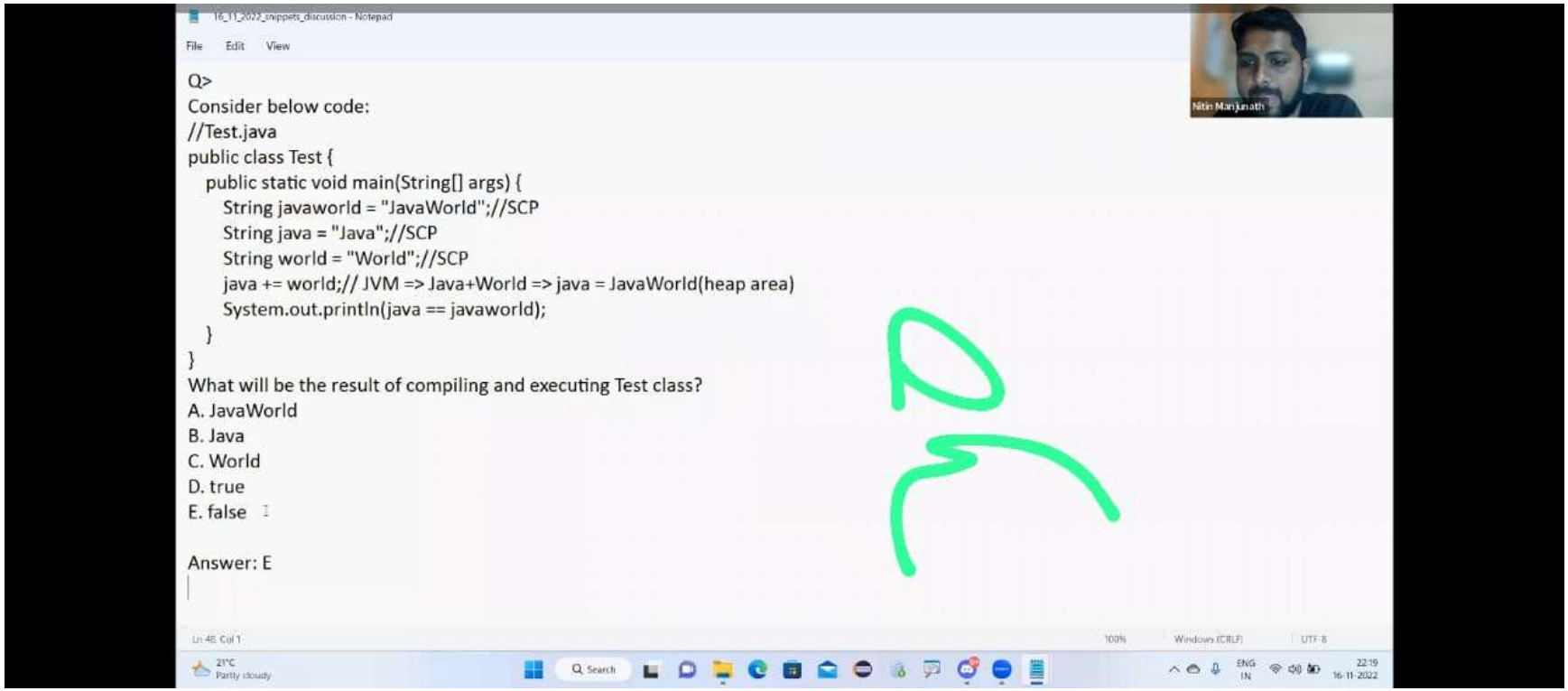

jobguaranteedbatch - 05-MANY-MANY-Association-mapping/src/in/ineuron/bean/Student.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
01-PrimitiveValue-Injection
02-One-One-Association-mapping
03-One-MANY-Association-mapping
04-MANY-ONE-Association-mapping
05-MANY-MANY-Association-mappin
JRE System Library [JavaSE-1.8]
src
in.ineuron.bean
Course.java
Student.java
in.ineuron.main
TestApp.java
DemoApp
gitprojects
LambdaExpressionCode
TestApp
VariableProeject
*Student.java
TestApp.java
Course.java
3 //Target Object
4 public class Student {
5
6 private String sid;
7 private String sname;
8 private String saddr;
9
10 // HAS-A variable
11 private Course[] course;
12
13 // Constructor Injection
14 public Student(String sid, String sname, String saddr, Course[] course) {
15 super();
16 this.sid = sid;
17 this.sname = sname;
18 this.saddr = saddr;
19 this.course = course;
20 }
21
22 public void getStudentDetails() {
23 System.out.println();
24 System.out.println("SID IS :: "+sid);
25 System.out.println("SNAME IS :: "+sname);
26 System.out.println("SADDR IS :: "+saddr);
27 System.out.println("COURSE DETAILS ARE:: ");
28 }
29
Writable
Smart Insert
23 : 28 : 460
Nitin Manjunath
21°C
Partly cloudy
Search
ENG
IN
21:46
16-11-2022


File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
01-PrimitiveValue-Injection
02-One-One-Association-mapping
03-One-MANY-Association-mapping
04-MANY-ONE-Association-mapping
05-MANY-MANY-Association-mappin
JRE System Library [JavaSE-1.8]
src
in.ineuron.bean
Course.java
Student.java
in.ineuron.main
TestApp.java
DemoApp
gitprojects
LambdaExpressionCode
TestApp
VariableProeject
Student.java
TestApp.java
Course.java
1 package in.ineuron.bean;
2
3 //Dependent Object
4 public class Course {
5 String cid;
6 String cname;
7 int ccost;
8
9 public Course(String cid, String cname, int ccost) {
10 super();
11 this.cid = cid;
12 this.cname = cname;
13 this.ccost = ccost;
14 }
15
16 }
17
Nitin Manjunath
Writable
Smart Insert
11 : 24 : 200
21°C
Partly cloudy
Search
ENG
IN
21:40
16-11-2022


Nitin Manjunath
import in.ineuron.bean.Student;
public class TestApp {
public static void main(String[] args) {
Course c1 = new Course("A11", "JAVA", 1000);
Course c2 = new Course("B11", "Python", 2000);
Course c3 = new Course("C11", "Blockchain", 3000);
Course[] course = new Course[3];
course[0] = c1;
course[1] = c2;
course[2] = c3;
Student s1 = new Student("IND10", "sachin", "MI", course);
Student s2 = new Student("IND7", "dhoni", "CSK", course);
Student s3 = new Student("IND18", "kohli", "RCB", course);
s1.getStudentDetails();
s2.getStudentDetails();
s3.getStudentDetails();
}
}


16_11_2022_MANY_MANY_Association - Paint
File
View
Nitin Manjunath
Consolas
14
Background fill
s1
id = IND10
name = sachin
address = MI
course
s2
id = IND07
name = dhoni
address = CSK
course
s3
id = IND18
name = kohli
address = RCB
course
course[]
0
c1
c2
c3
cid =A11
cname =java
cost=1000
c1
cid =B11
cname =python
cost =2000
c2
cid =C11
cname =blockchain
cost =3000
c3
100%
21°C
Partly cloudy
Search
ENG
IN
21:54
16-11-2022


16_11_2022_snippets_discussion - Notepad
File Edit View
public static void main(String[] args) {
final int i1 = 1;
final Integer i2 = 1;//memory will be decied at the runtime becoz it is wrapper class Object
final String s1 = ":ONE";
String str1 = i1 + s1;// Compiler :: 1 + :ONE => 1:ONE
String str2 = i2 + s1;// Compiler :: i2 + :ONE
System.out.println(str1 == "1:ONE");//true
System.out.println(str2 == "1:ONE");//false
}
}
What will be the result of compiling and executing Test class?
A. true
true
B. true
false
C. false
false
D. false
true
Answer: B
Nitin Manjunath
Strings
Ln 6, Col 30
100%
Windows (CRLF)
UTF-8
21°C
Partly cloudy
Search
ENG
IN
22:16
16-11-2022


16_11_2022_snippets_discussion - Notepad
File Edit View
Q>
Consider below code:
//Test.java
public class Test {
public static void main(String[] args) {
String javaworld = "JavaWorld";//SCP
String java = "Java";//SCP
String world = "World";//SCP
java += world;// JVM => Java+World => java = JavaWorld(heap area)
System.out.println(java == javaworld);
}
}
What will be the result of compiling and executing Test class?
A. JavaWorld
B. Java
C. World
D. true
E. false
Answer: E
Nitin Manjunath
Ln 48, Col 1
100%
Windows (CRLF)
UTF-8
21°C
Partly cloudy
Search
ENG
IN
22:19
16-11-2022

16_11_2022_snippets_discussion - Notepad

File Edit View

Q>

```
public class Test {
   public static void main(String[] args) {
      String s1 = "OCP";
      String s2 = "ocp";
      System.out.println(/*INSERT*/);
   }
}
```

Which of the following options, if used to replace /*INSERT*/, will compile successfully and on execution will print true on to the console?
Select 2 options.

A. s1.equals(s2)
B. s1.equals(s2.toUpper())
C. s1.equals(s1.toLower())
D. s1.length()==s2.length()
E. s1.equalsIgnoreCase(s2)

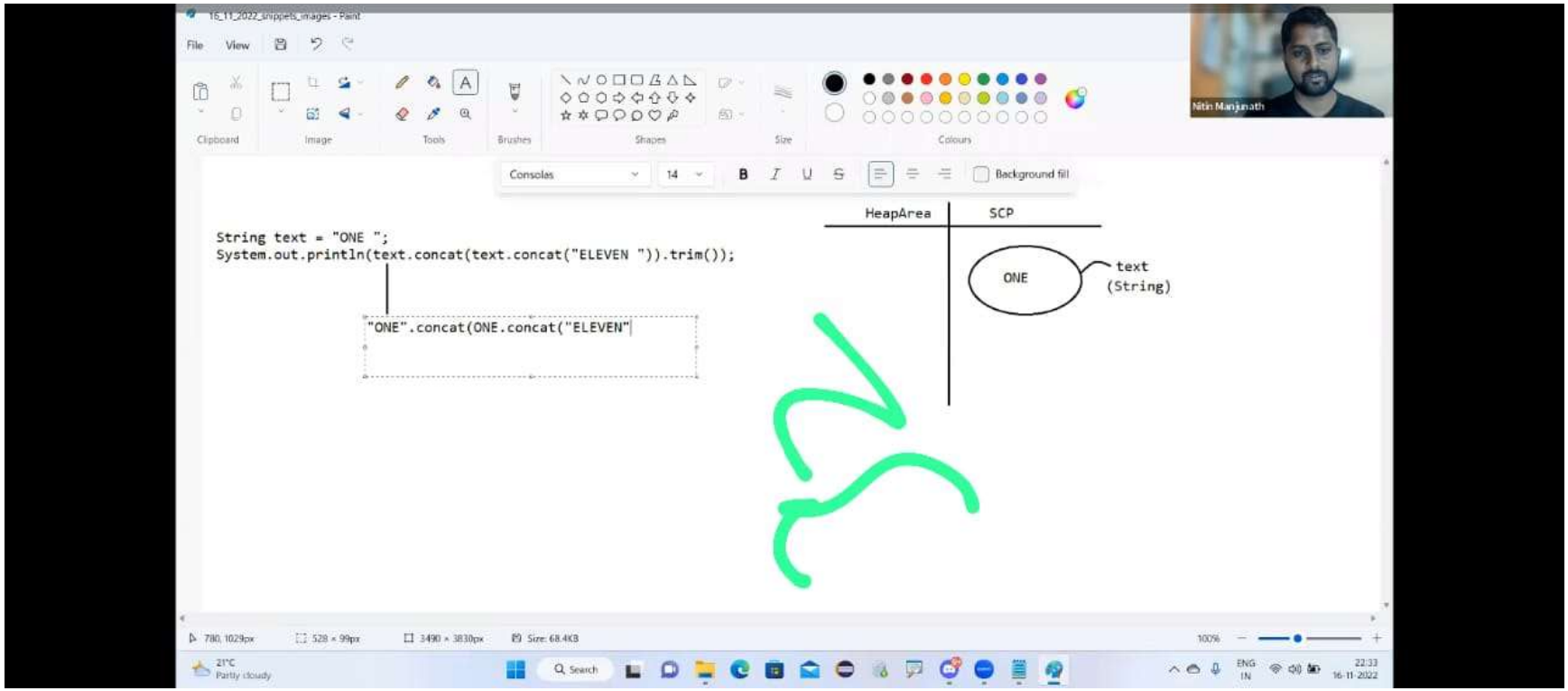
Answer: D,E

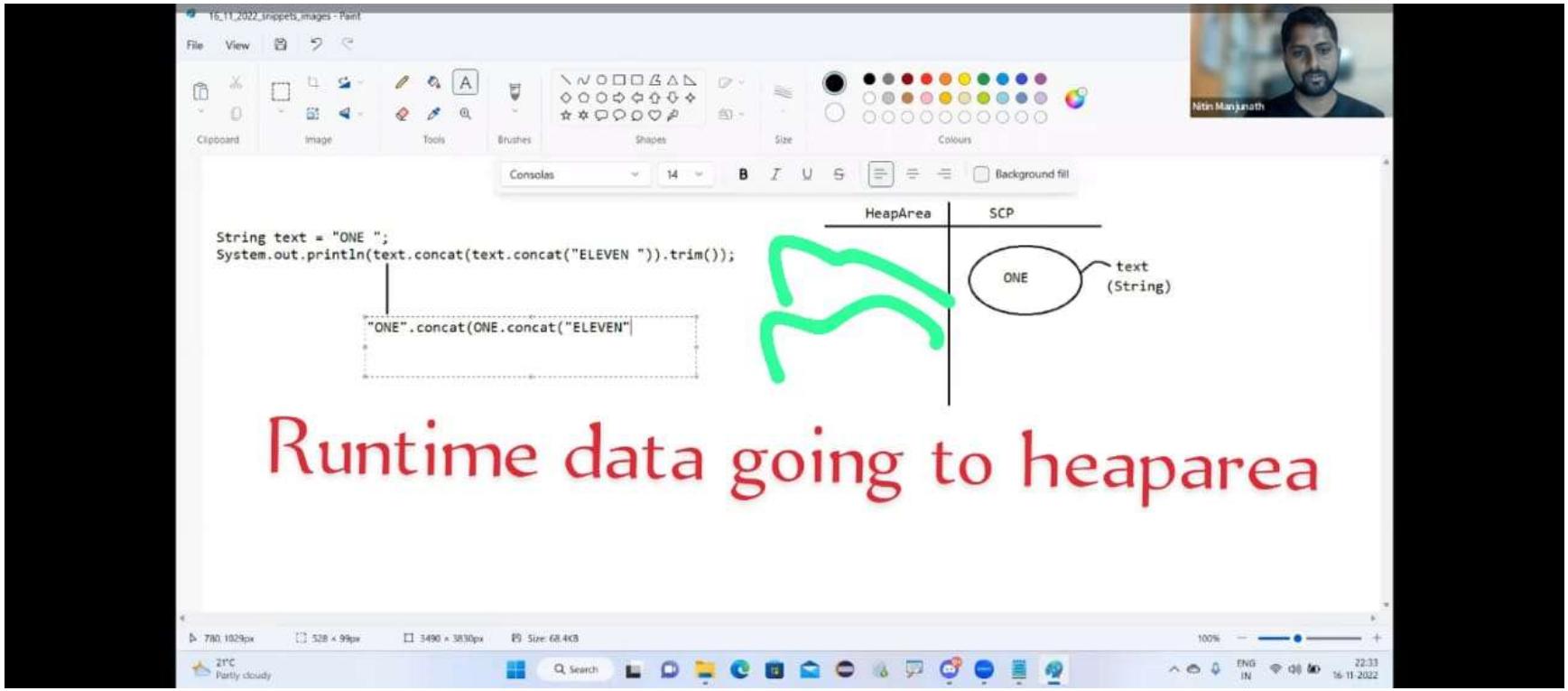
Ln 77, Col 27 100% Windows (CRLF) UTF-8

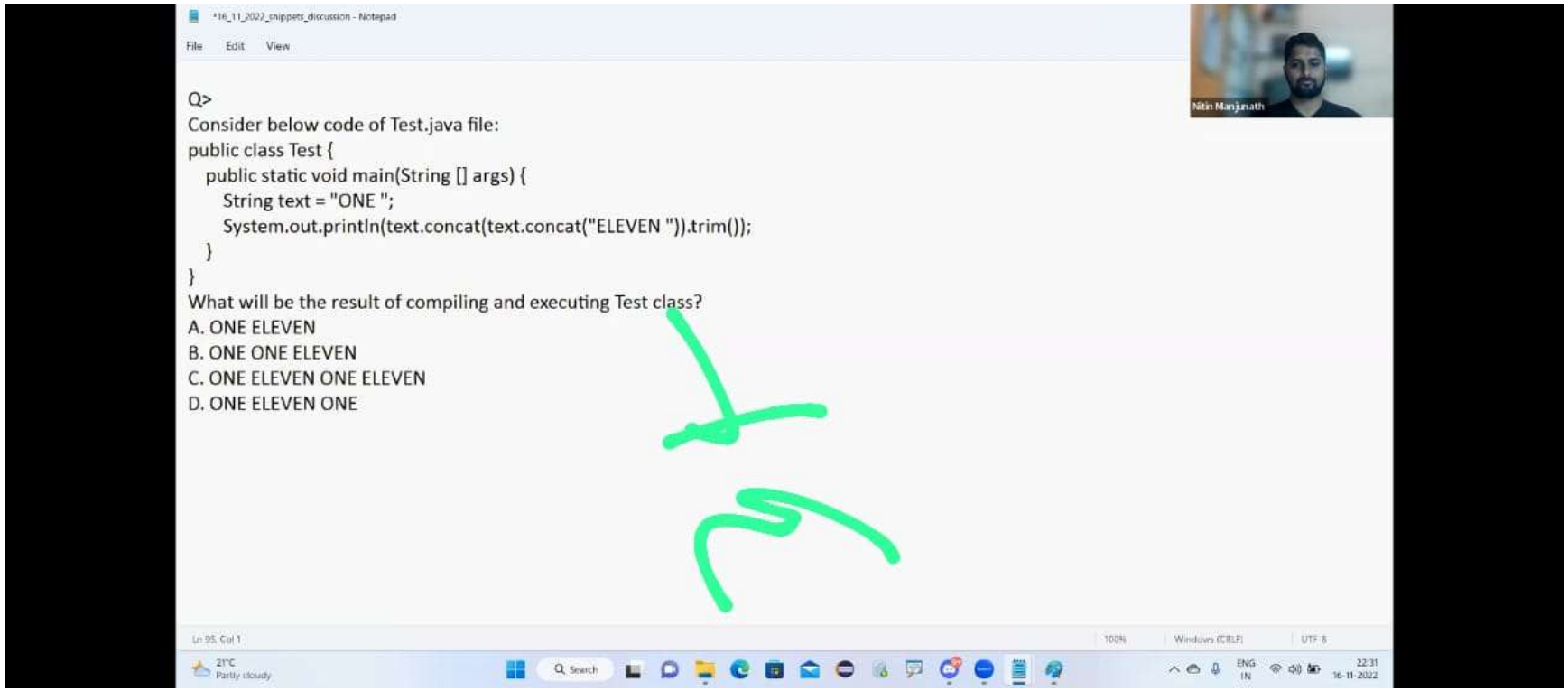
21°C Partly cloudy

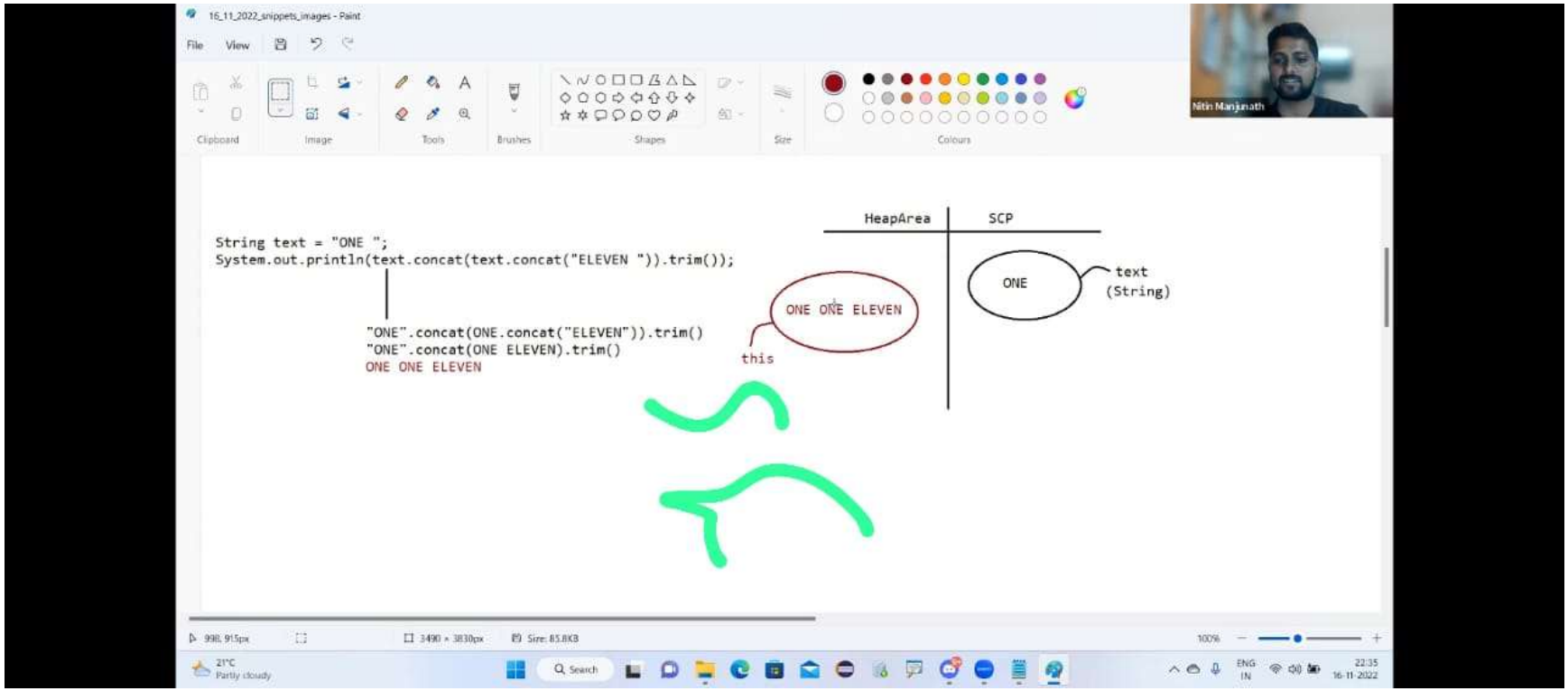
Search

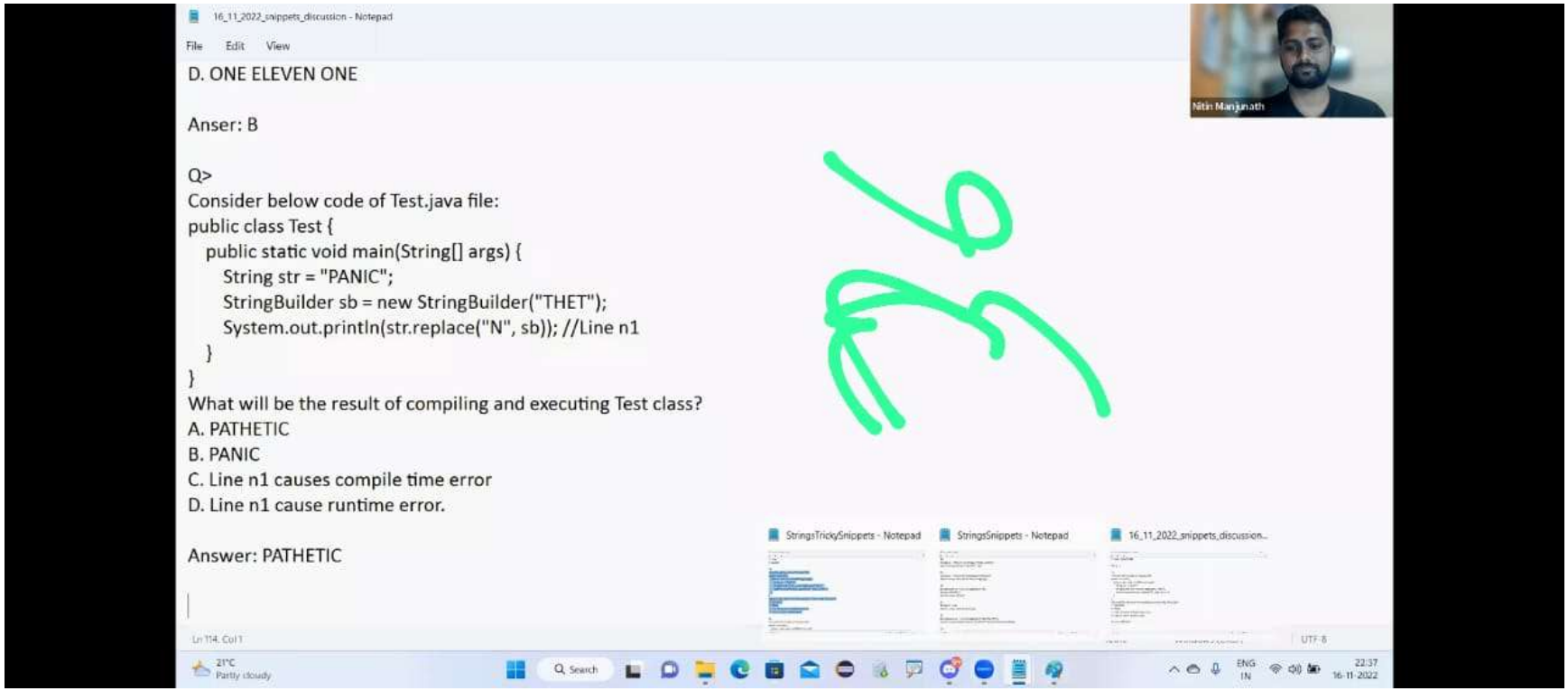
ENG IN 22:29 16-11-2022

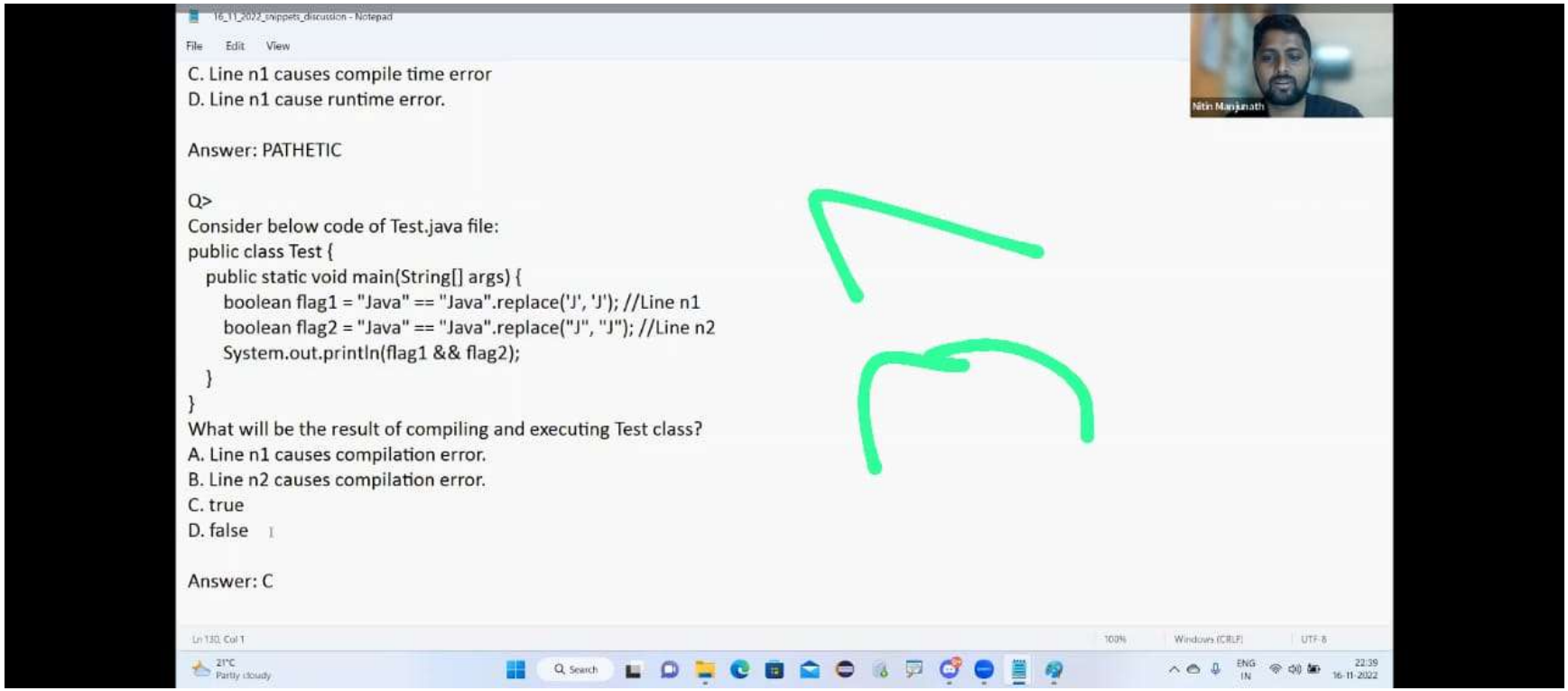

16_11_2022_snippets_images - Paint
File
View
Clipboard
Image
Tools
Brushes
Shapes
Size
Colours
Nitin Manjunath
Consolas
14
Background fill
String text = "ONE ";
System.out.println(text.concat(text.concat("ELEVEN ")).trim());
"ONE".concat(ONE.concat("ELEVEN"
HeapArea
SCP
ONE
text
(String)
780, 1029px
528 × 99px
3490 × 3830px
Size: 68.4KB
100%
21°C
Partly cloudy
Search
ENG
IN
22:33
16-11-2022

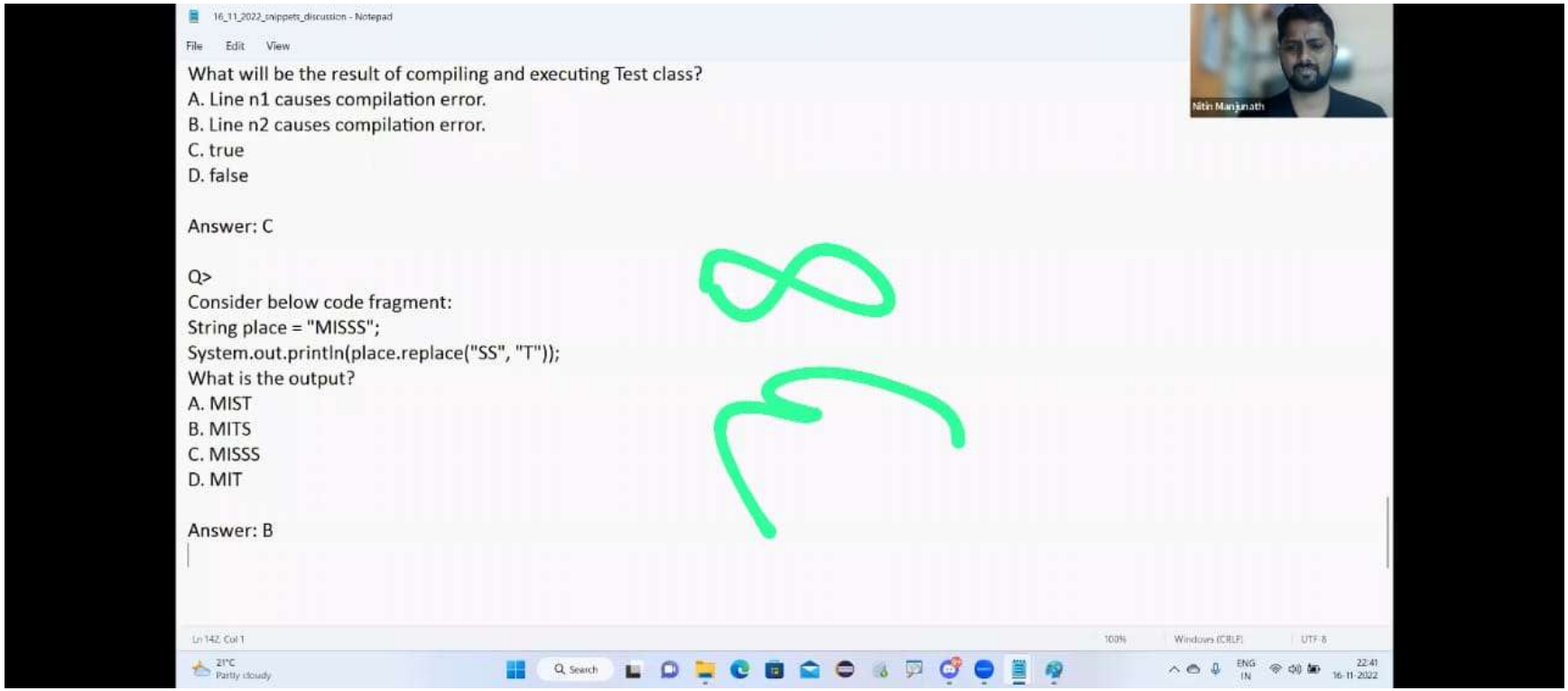

Nitin Manjunath
File
View
Consolas
14
Background fill
String text = "ONE ";
System.out.println(text.concat(text.concat("ELEVEN ")).trim());
"ONE".concat(ONE.concat("ELEVEN"
HeapArea
SCP
ONE
text
(String)
Runtime data going to heaparea

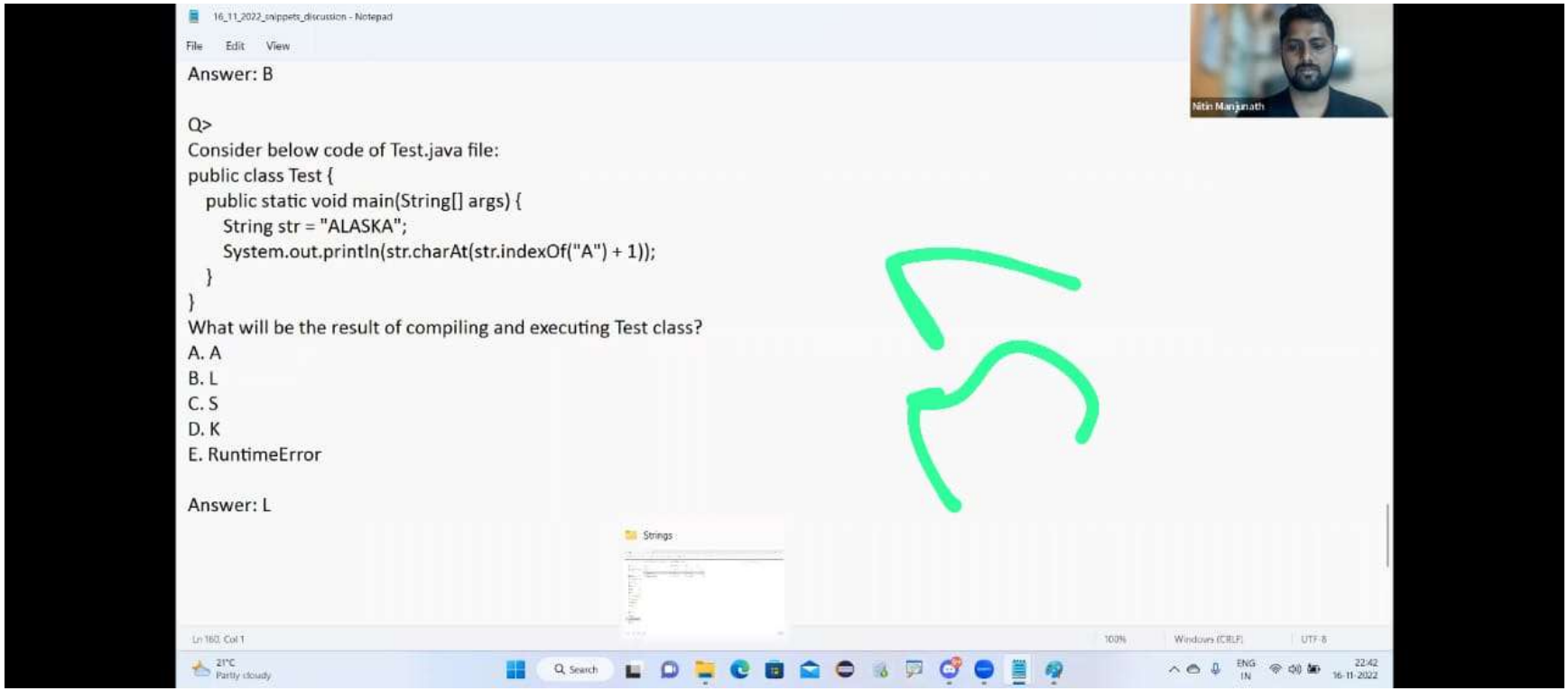

*16_11_2022_snippets_discussion - Notepad
File Edit View
Nitin Manjunath
Q>
Consider below code of Test.java file:
public class Test {
public static void main(String [] args) {
String text = "ONE ";
System.out.println(text.concat(text.concat("ELEVEN ")).trim());
}
}
What will be the result of compiling and executing Test class?
A. ONE ELEVEN
B. ONE ONE ELEVEN
C. ONE ELEVEN ONE ELEVEN
D. ONE ELEVEN ONE
Ln 95, Col 1
100%
Windows (CRLF)
UTF-8
21°C
Partly cloudy
Search
ENG
IN
22:31
16-11-2022

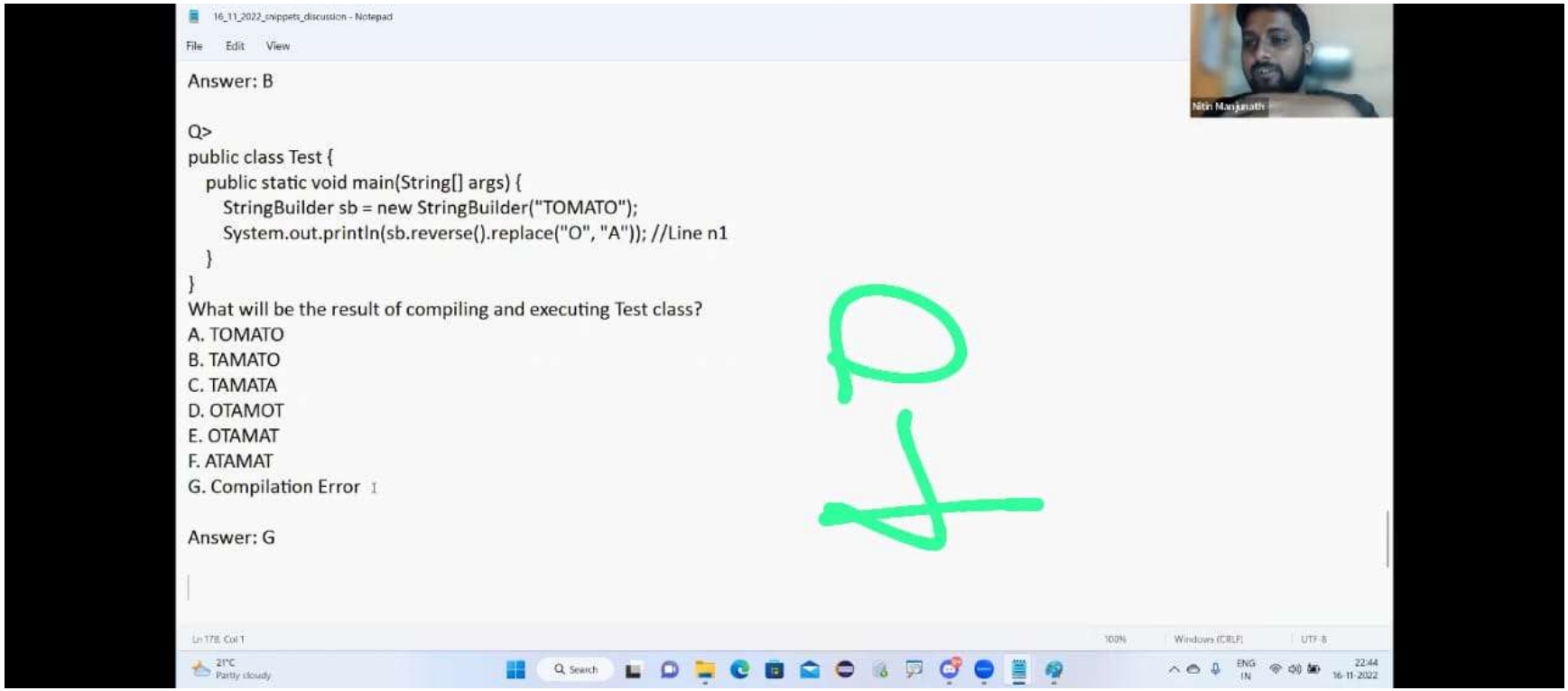

16_11_2022_snippets_images - Paint
File
View
Clipboard
Image
Tools
Brushes
Shapes
Size
Colours
Nitin Manjunath
String text = "ONE ";
System.out.println(text.concat(text.concat("ELEVEN ")).trim());
"ONE".concat(ONE.concat("ELEVEN")).trim()
"ONE".concat(ONE ELEVEN).trim()
ONE ONE ELEVEN
HeapArea
SCP
ONE
text
(String)
ONE ONE ELEVEN
this
998, 915px
3490 × 3830px
Size: 85.8KB
100%
21°C
Partly cloudy
Search
ENG
IN
22:35
16-11-2022

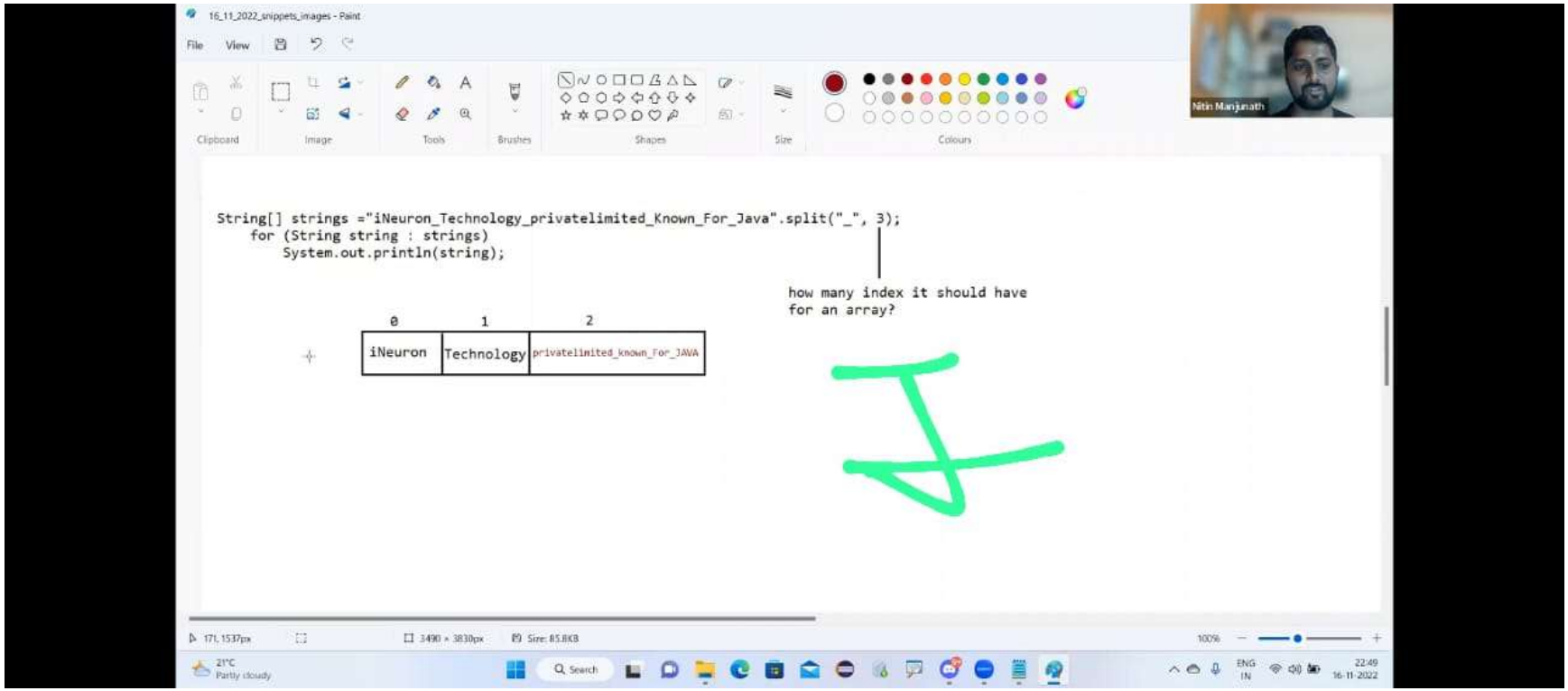

16_11_2022_snippets_discussion - Notepad
File Edit View
D. ONE ELEVEN ONE
Anser: B
Q>
Consider below code of Test.java file:
public class Test {
public static void main(String[] args) {
String str = "PANIC";
StringBuilder sb = new StringBuilder("THET");
System.out.println(str.replace("N", sb)); //Line n1
}
}
What will be the result of compiling and executing Test class?
A. PATHETIC
B. PANIC
C. Line n1 causes compile time error
D. Line n1 cause runtime error.
Answer: PATHETIC
Nitin Manjunath
StringsTrickySnippets - Notepad
StringsSnippets - Notepad
16_11_2022_snippets_discussion...
Ln 114, Col 1
UTF-8
21°C
Partly cloudy
Search
ENG
IN
22:37
16-11-2022


16_11_2022_snippets_discussion - Notepad
File Edit View
C. Line n1 causes compile time error
D. Line n1 cause runtime error.
Answer: PATHETIC
Q>
Consider below code of Test.java file:
public class Test {
public static void main(String[] args) {
boolean flag1 = "Java" == "Java".replace('J', 'J'); //Line n1
boolean flag2 = "Java" == "Java".replace("J", "J"); //Line n2
System.out.println(flag1 && flag2);
}
}
What will be the result of compiling and executing Test class?
A. Line n1 causes compilation error.
B. Line n2 causes compilation error.
C. true
D. false
Answer: C
Nitin Manjunath
Ln 130, Col 1
100%
Windows (CRLF)
UTF-8
21°C
Partly cloudy
Search
ENG
IN
22:39
16-11-2022


16_11_2022_snippets_discussion - Notepad
File Edit View
What will be the result of compiling and executing Test class?
A. Line n1 causes compilation error.
B. Line n2 causes compilation error.
C. true
D. false
Answer: C
Q>
Consider below code fragment:
String place = "MISSS";
System.out.println(place.replace("SS", "T"));
What is the output?
A. MIST
B. MITS
C. MISSS
D. MIT
Answer: B
Nitin Manjunath
Ln 142, Col 1
100%
Windows (CRLF)
UTF-8
21°C
Partly cloudy
Search
ENG
IN
22:41
16-11-2022


16_11_2022_snippets_discussion - Notepad
File Edit View
Answer: B
Q>
Consider below code of Test.java file:
public class Test {
public static void main(String[] args) {
String str = "ALASKA";
System.out.println(str.charAt(str.indexOf("A") + 1));
}
}
What will be the result of compiling and executing Test class?
A. A
B. L
C. S
D. K
E. RuntimeError
Answer: L
Nitin Manjunath
Strings
Ln 160, Col 1
100%
Windows (CRLF)
UTF-8
21°C
Partly cloudy
Search
ENG
IN
22:42
16-11-2022


16_11_2022_snippets_discussion - Notepad
File Edit View
Nitin Manjunath
Answer: B
Q>
public class Test {
public static void main(String[] args) {
StringBuilder sb = new StringBuilder("TOMATO");
System.out.println(sb.reverse().replace("O", "A")); //Line n1
}
}
What will be the result of compiling and executing Test class?
A. TOMATO
B. TAMATO
C. TAMATA
D. OTAMOT
E. OTAMAT
F. ATAMAT
G. Compilation Error
Answer: G
Ln 178, Col 1
100%
Windows (CRLF)
UTF-8
21°C
Partly cloudy
Search
ENG
IN
22:44
16-11-2022


16_11_2022_snippets_images - Paint
File
View
Clipboard
Image
Tools
Brushes
Shapes
Size
Colours
Nitin Manjunath
String[] strings ="iNeuron_Technology_privatelimited_Known_For_Java".split("_", 3);
for (String string : strings)
System.out.println(string);
how many index it should have
for an array?
0
1
2
iNeuron
Technology
privatelimited_known_For_JAVA
171, 1537px
3490 × 3830px
Size: 85.8KB
100%
21°C
Partly cloudy
Search
ENG
IN
22:49
16-11-2022

16_11_2022_snippets_discussion - Notepad

File Edit View

All of the above

51.

```
String[] strings = "iNeuron_Technology_privatelimited_Known_For_Java".split("_", 3);
   for (String string : strings)
      System.out.println(string);
```

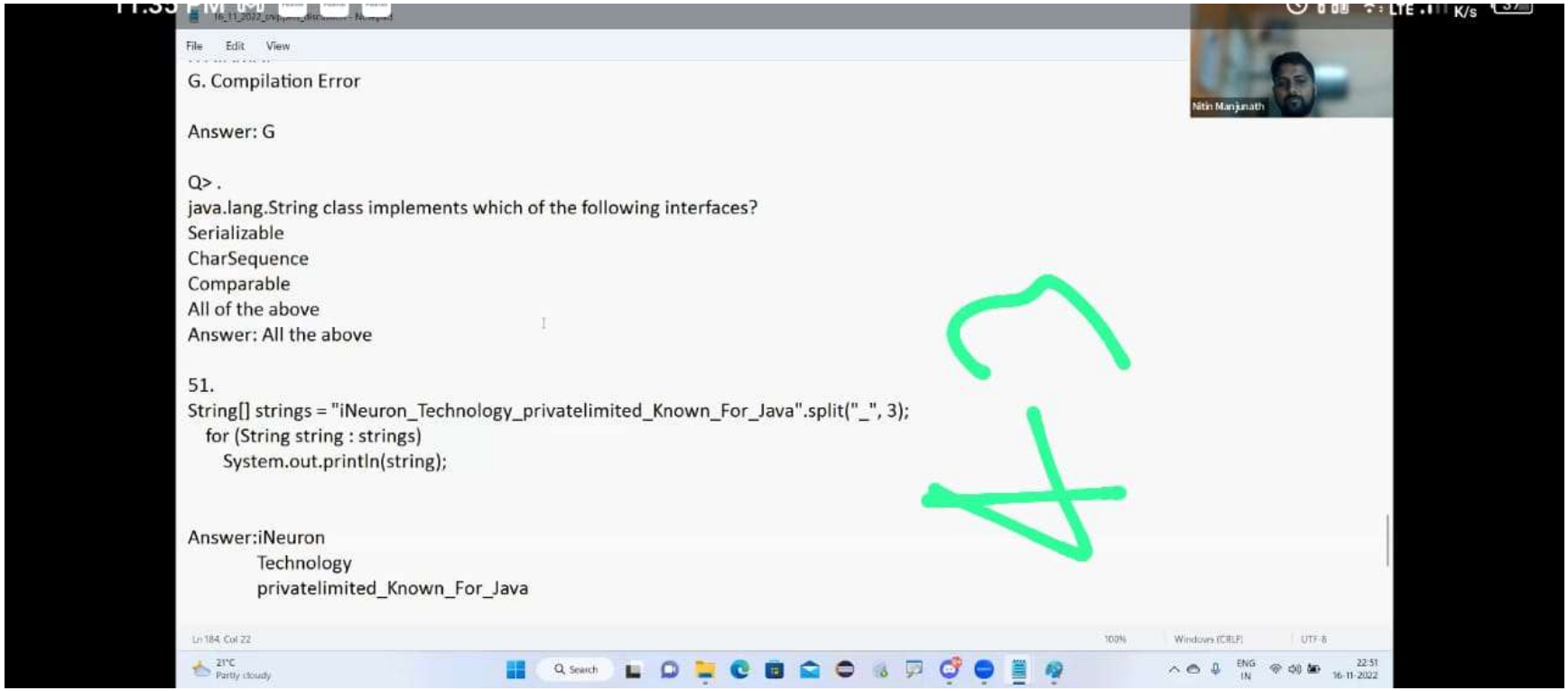
Answer:iNeuron
Technology
privatelimited_Known_For_Java

Ln 193, Col 9 100% Windows (CRLF) UTF-8

21°C Partly cloudy

Search

ENG IN 22:50 16-11-2022


File Edit View
G. Compilation Error
Answer: G
Q> .
java.lang.String class implements which of the following interfaces?
Serializable
CharSequence
Comparable
All of the above
Answer: All the above
51.
String[] strings = "iNeuron_Technology_privatelimited_Known_For_Java".split("_", 3);
for (String string : strings)
System.out.println(string);
Answer:iNeuron
Technology
privatelimited_Known_For_Java
Nitin Manjunath