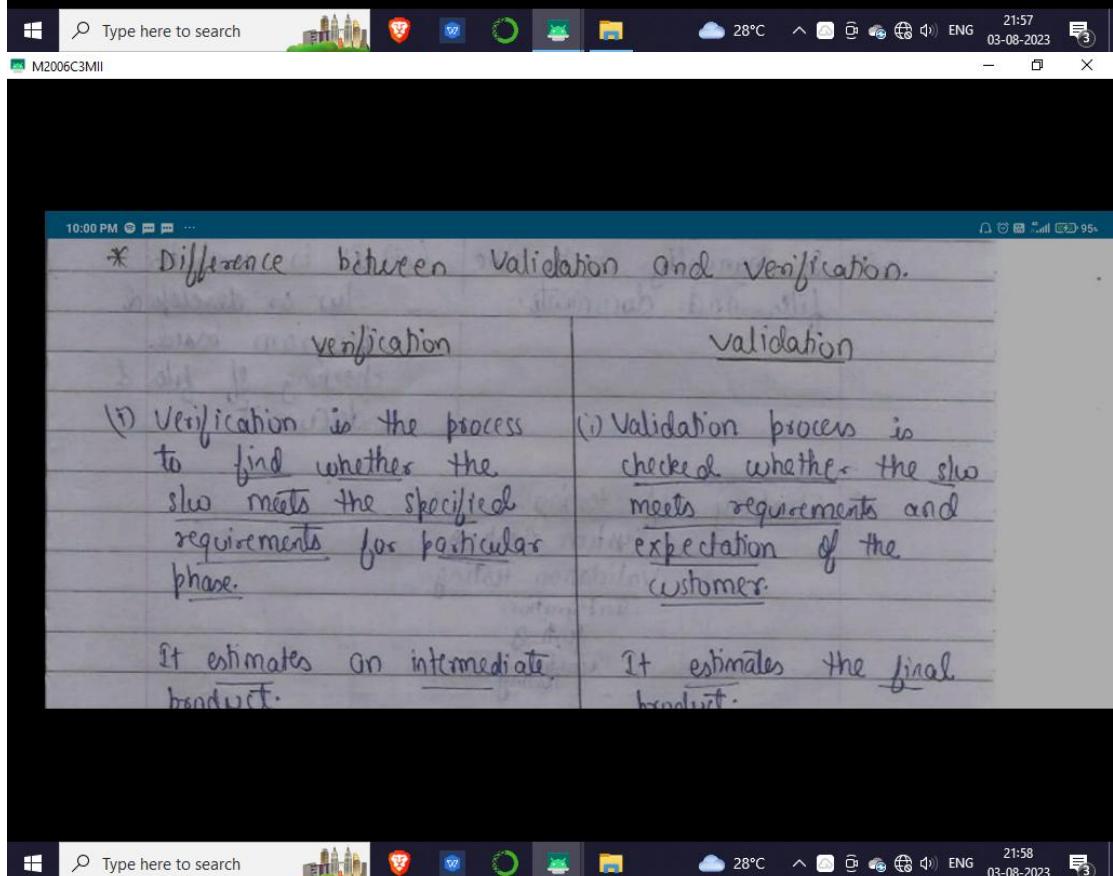
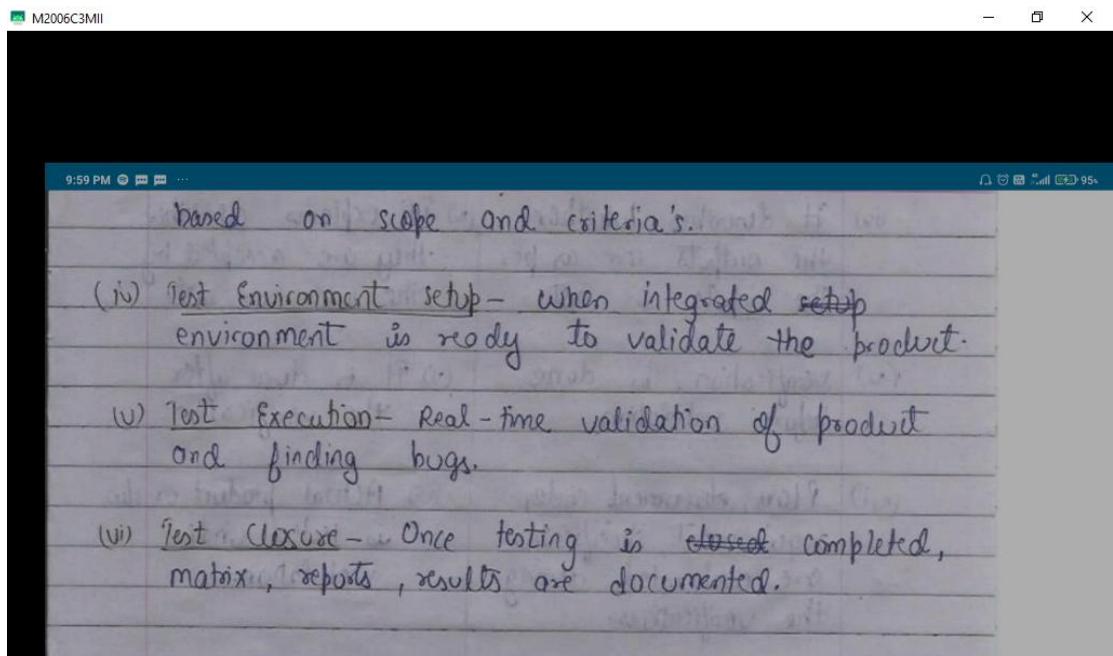
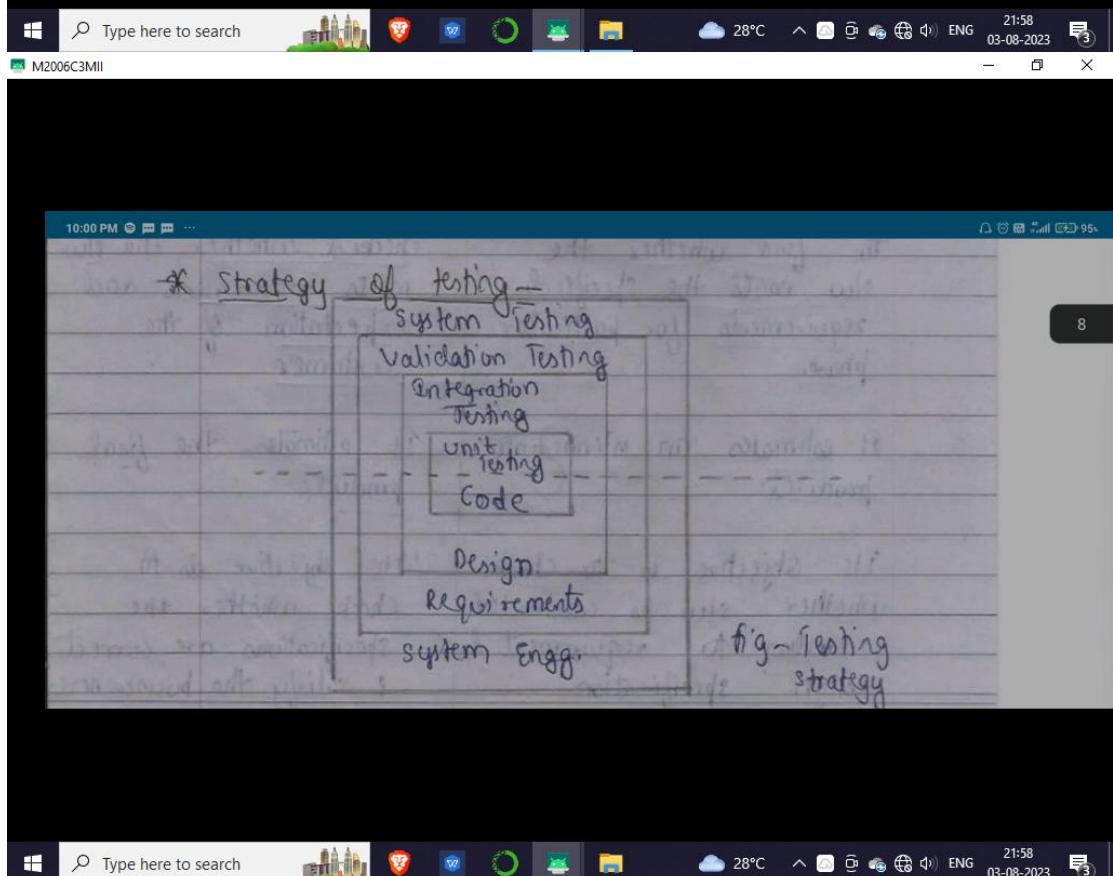
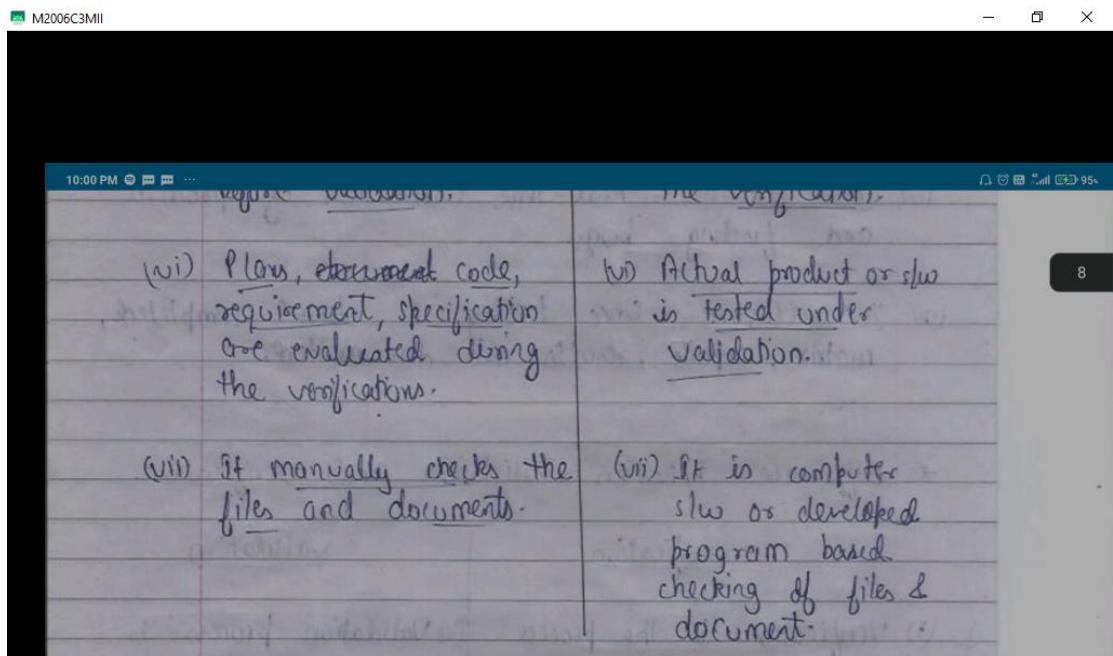


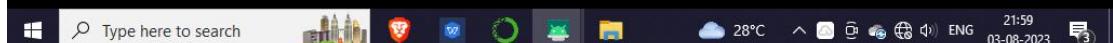
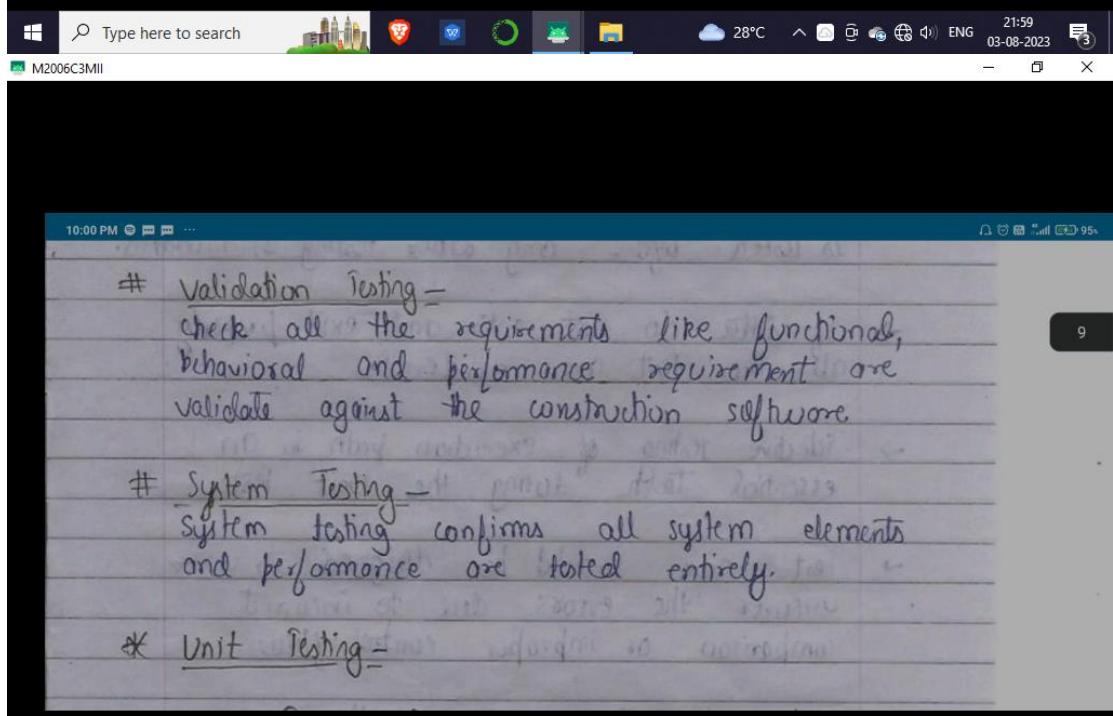
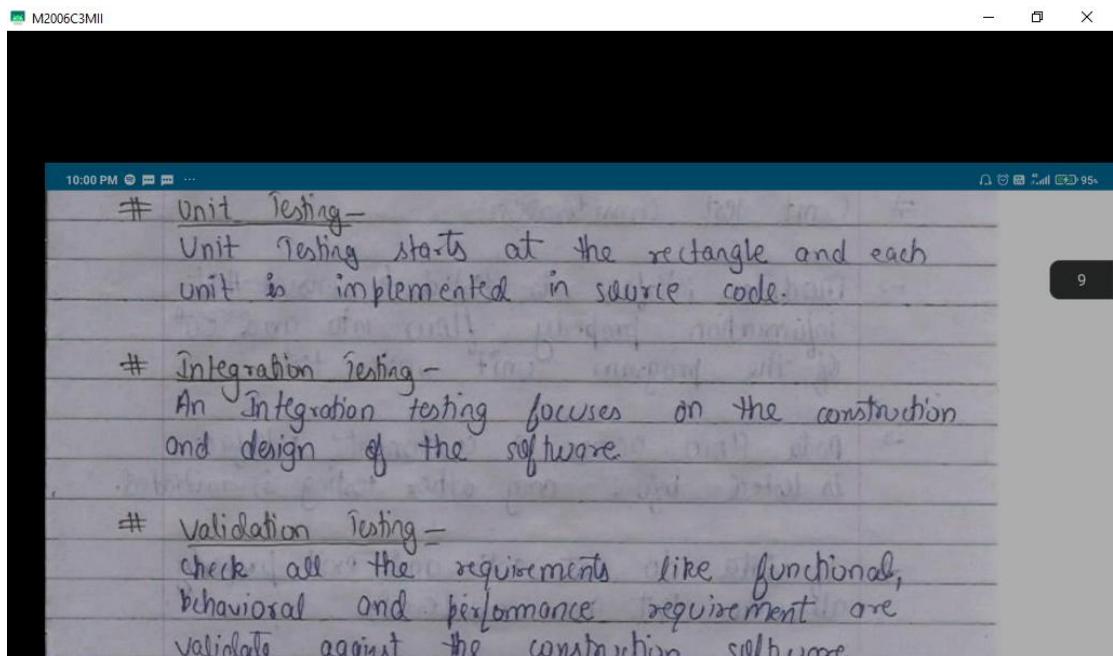
- (i) Requirement Analysis - when the software requirements are ready and shared with the stakeholders, the testing team starts high level analysis concerning the Application under test.
- (ii) Test Planning - Test team plans the strategy and approach.
- (iii) Test Case Designing - Develop the test cases

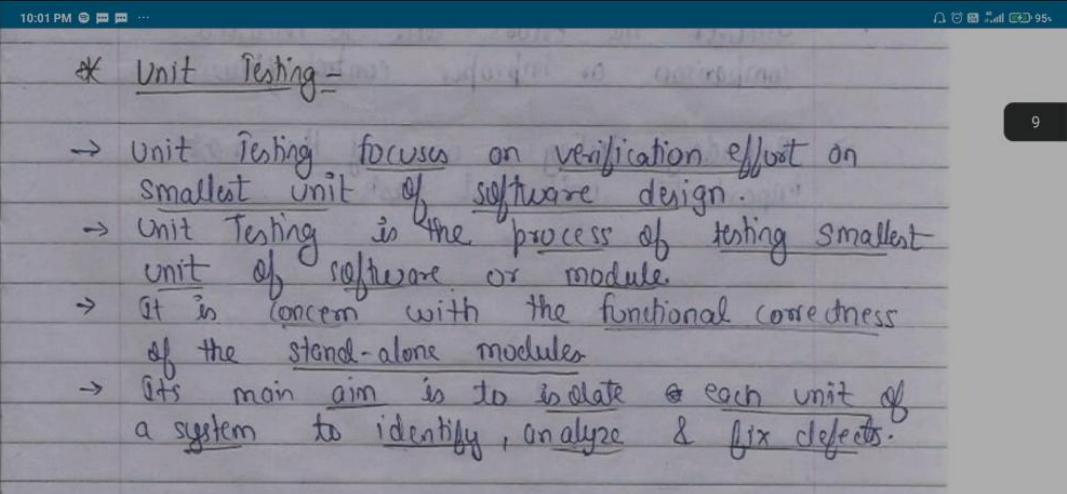


It estimates an intermediate product.	It estimates the final product.
Its objective is to check whether s/w is constructed according to requirement & design specification.	Its objective is to check whether the specifications are correct & satisfy the business need.

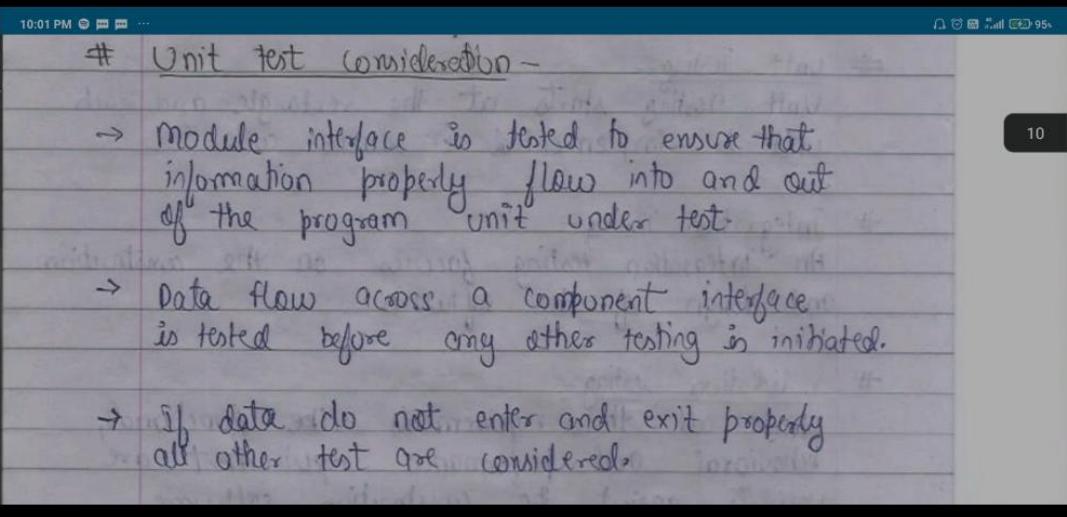
(iv) It describes whether the outputs are as per the inputs or not.	(iv) It explains whether they are accepted by the user or not.
(v) verification is done before validation.	(v) It is done after the verification.
(vi) Plans, environment code, requirement, specification are evaluated during the verifications.	(vi) Actual product or s/w is tested under validation.





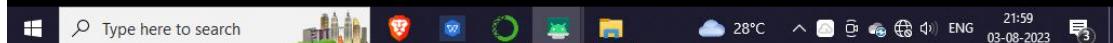
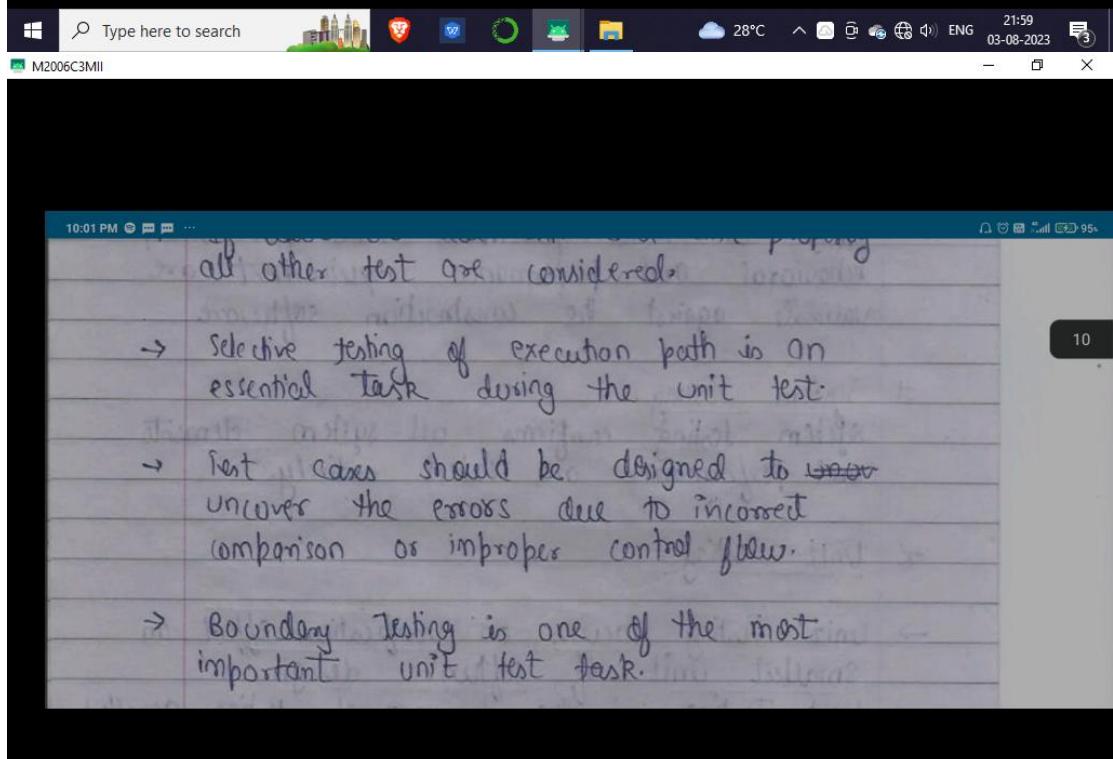
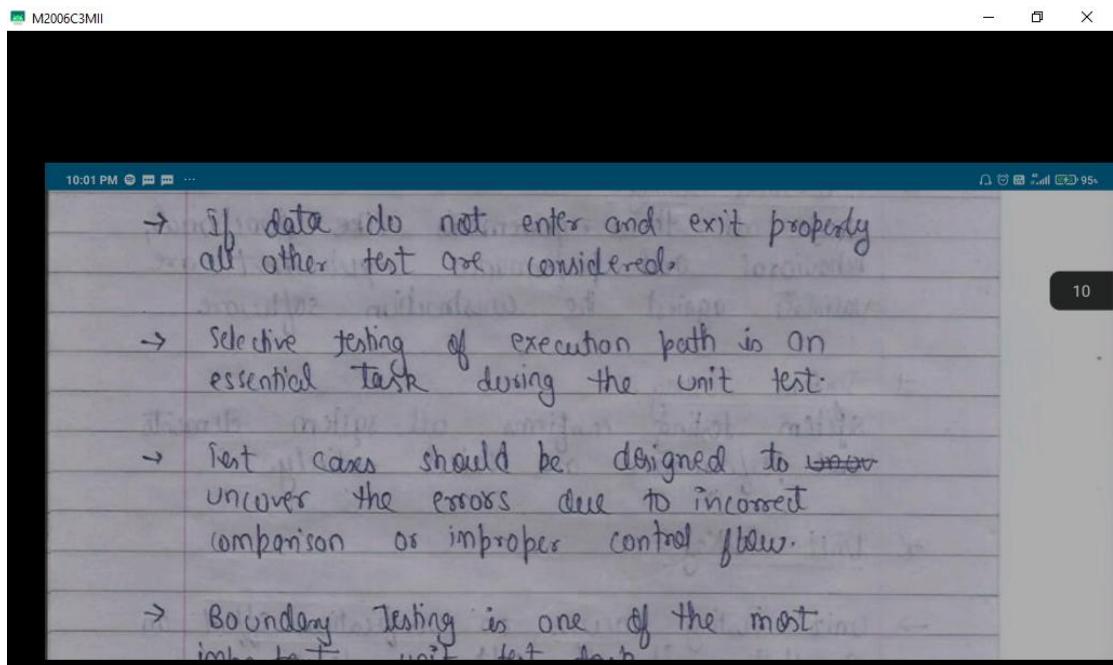


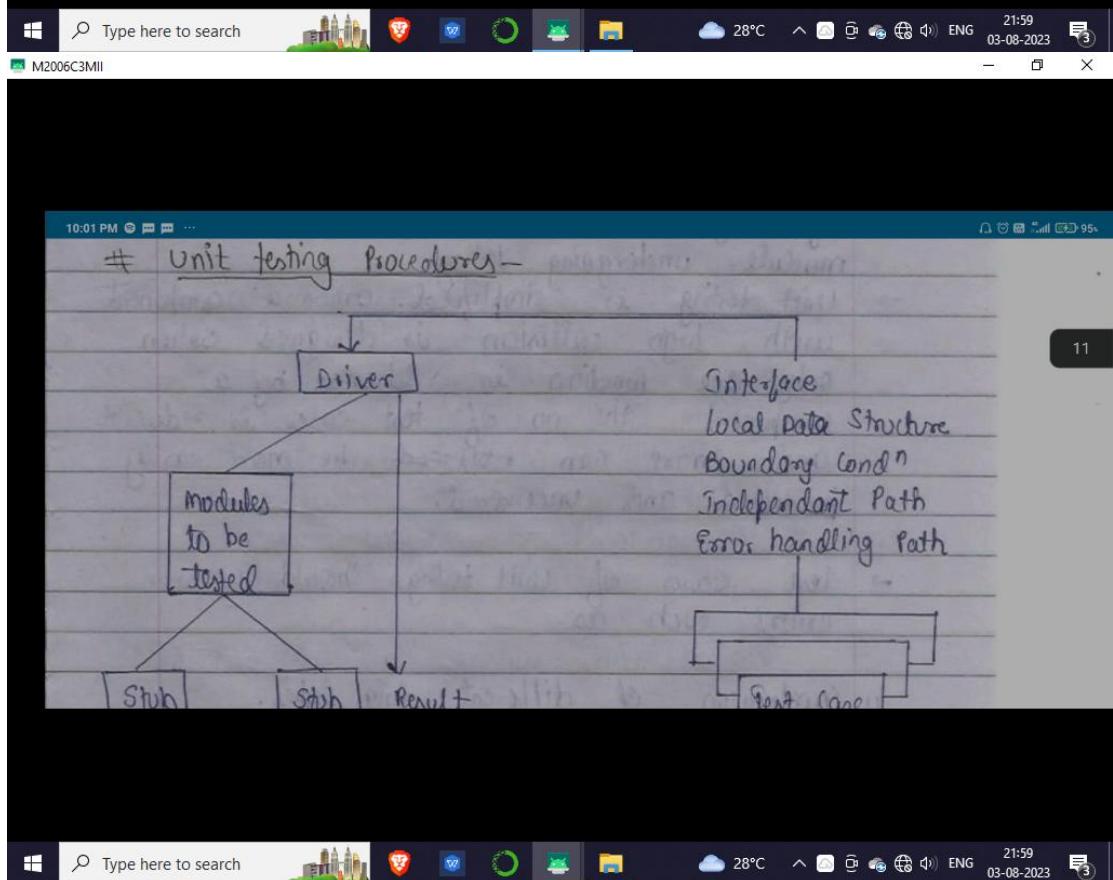
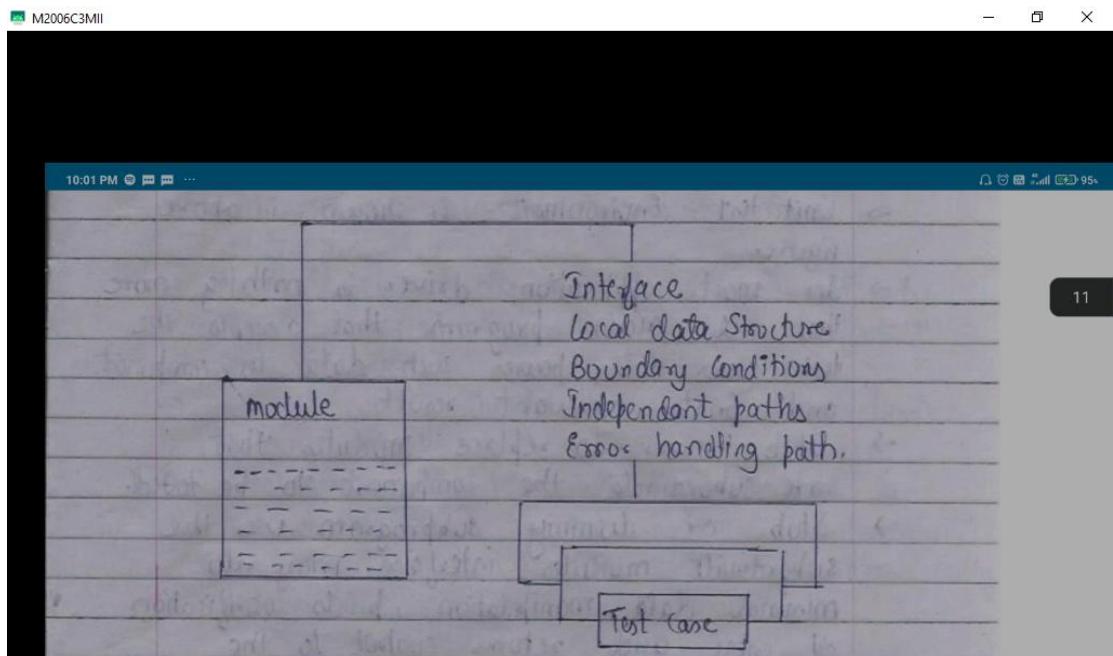
9

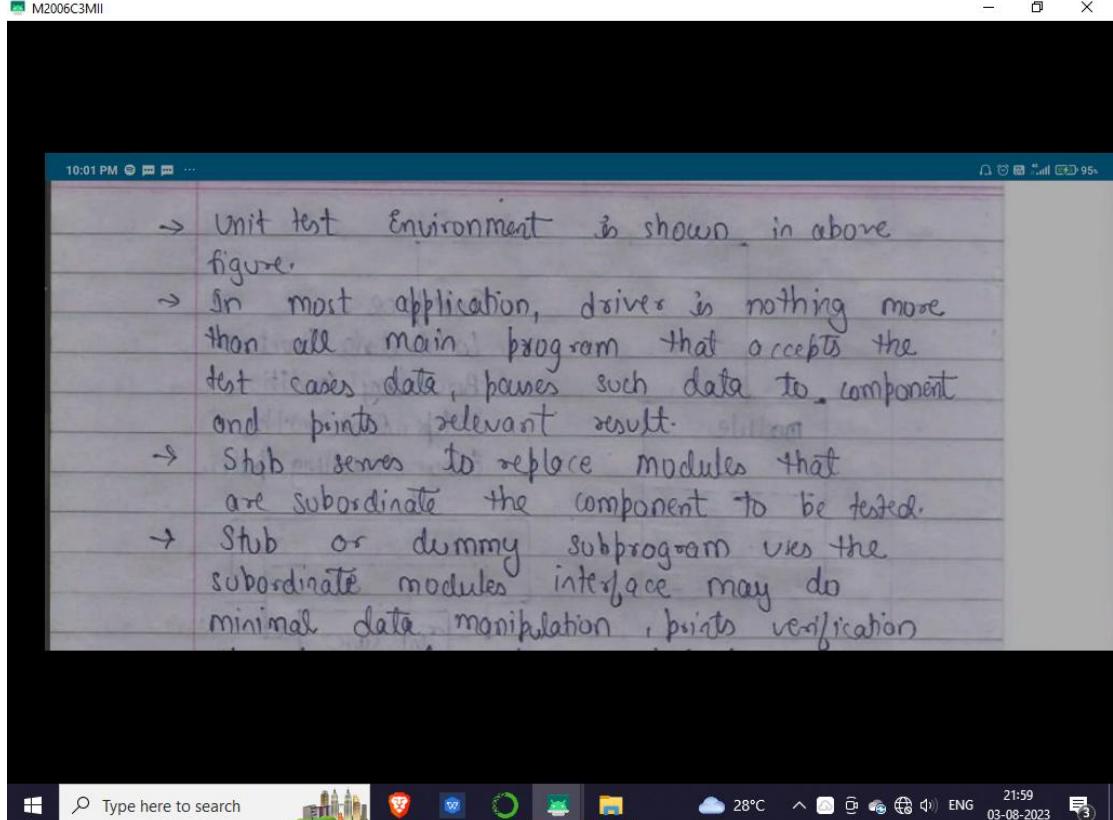
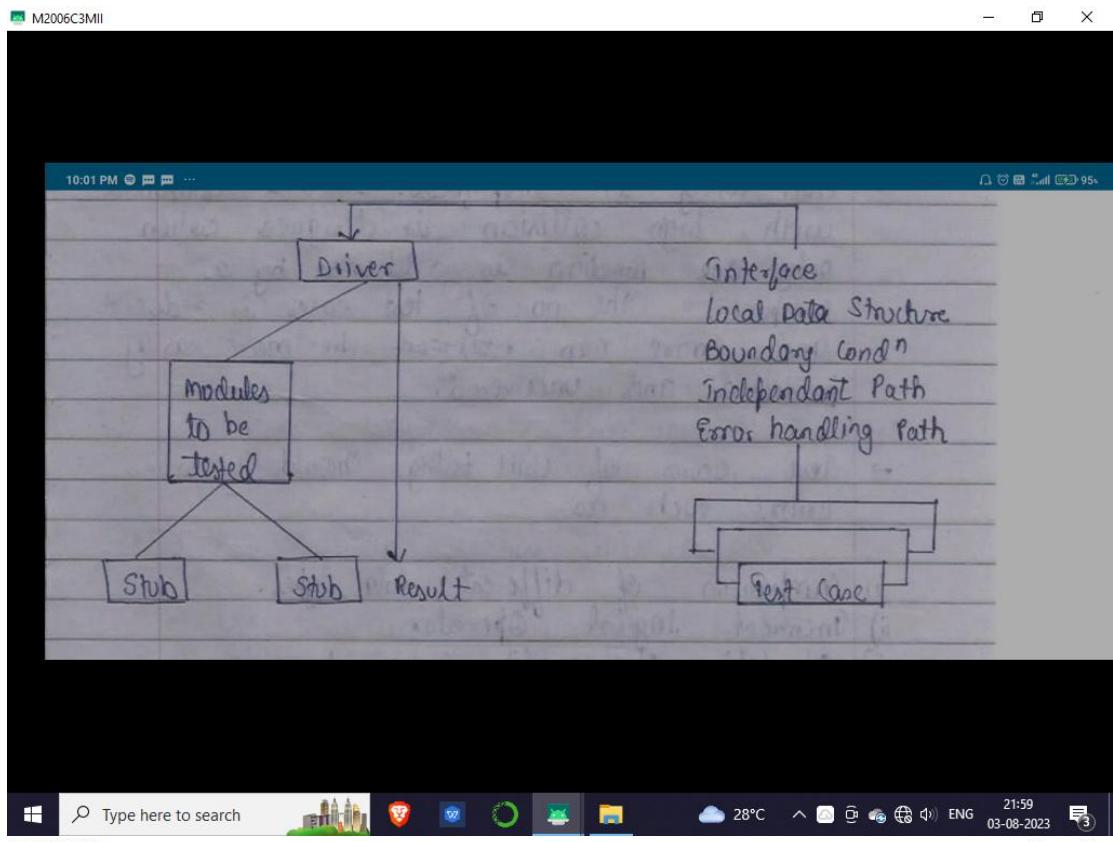


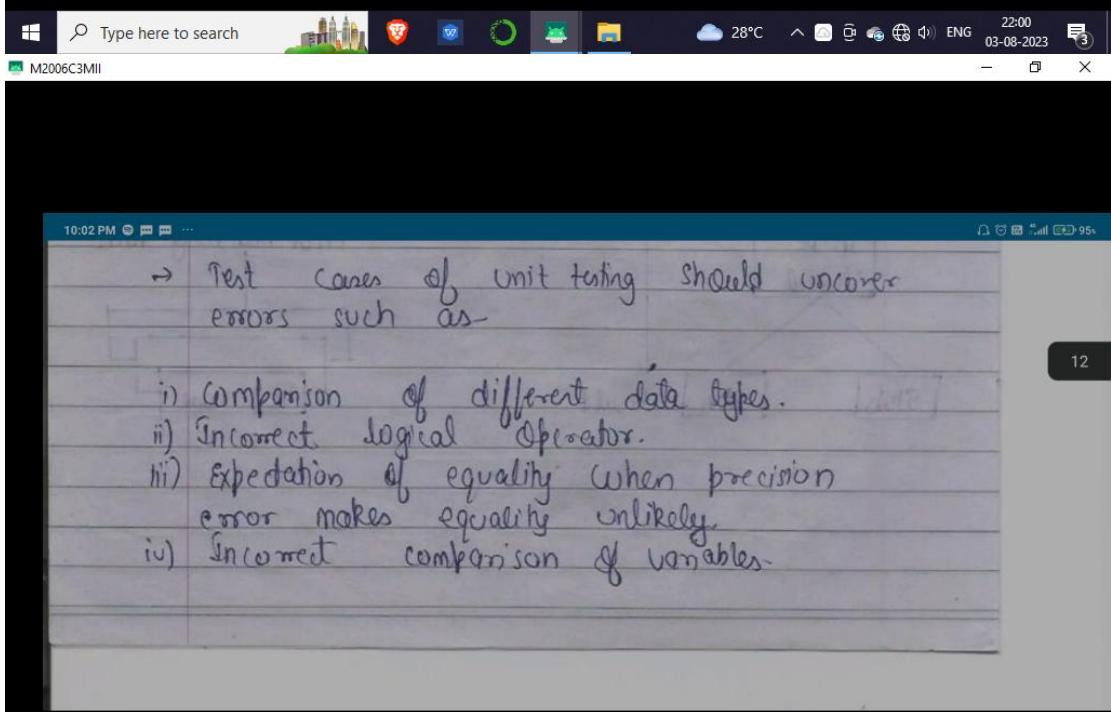
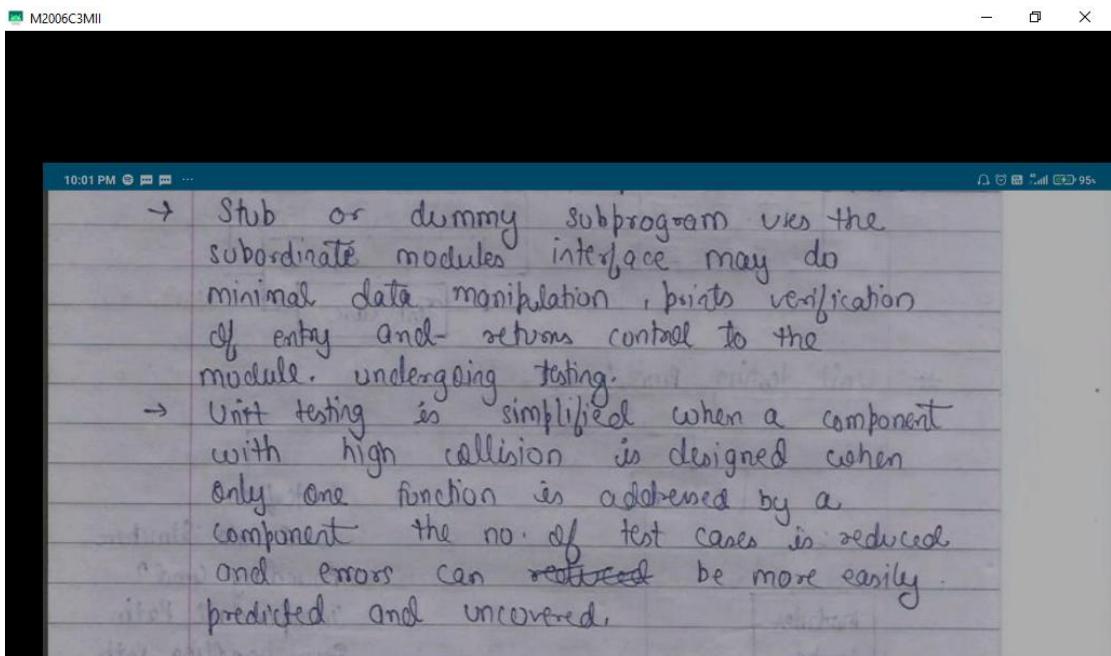
10

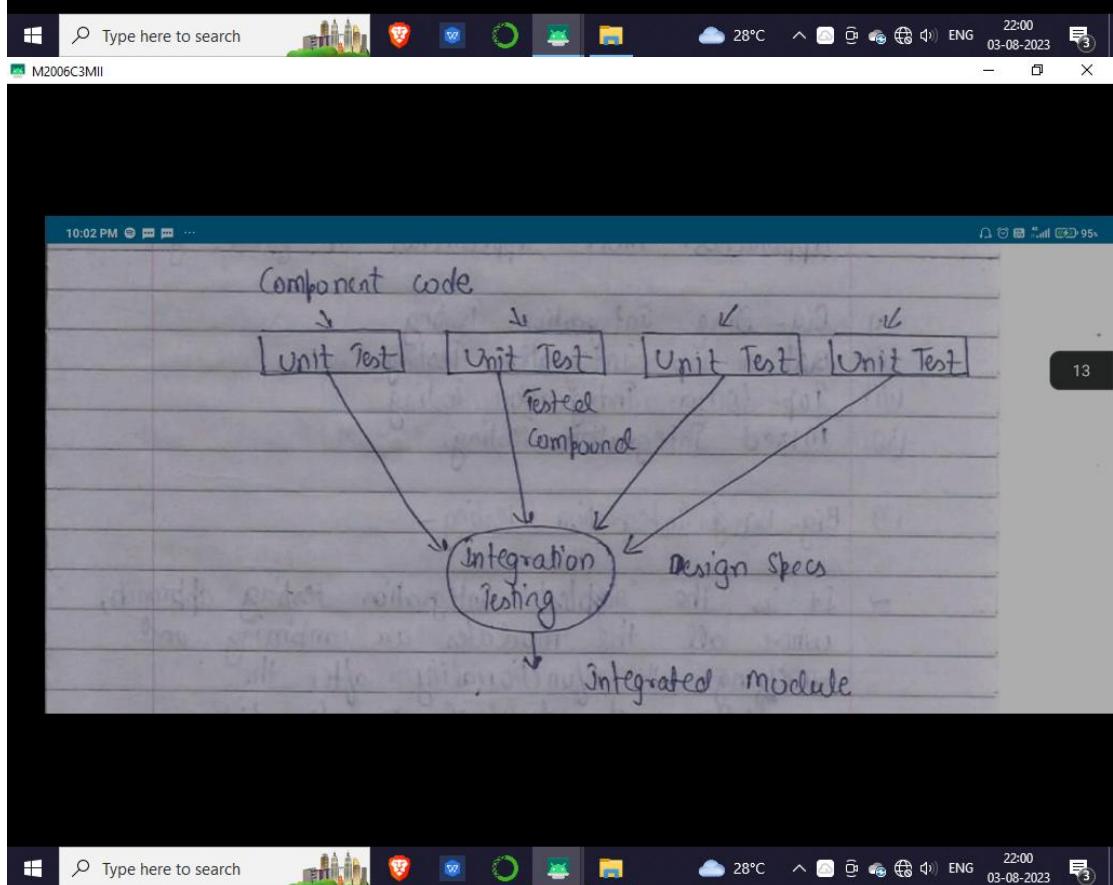
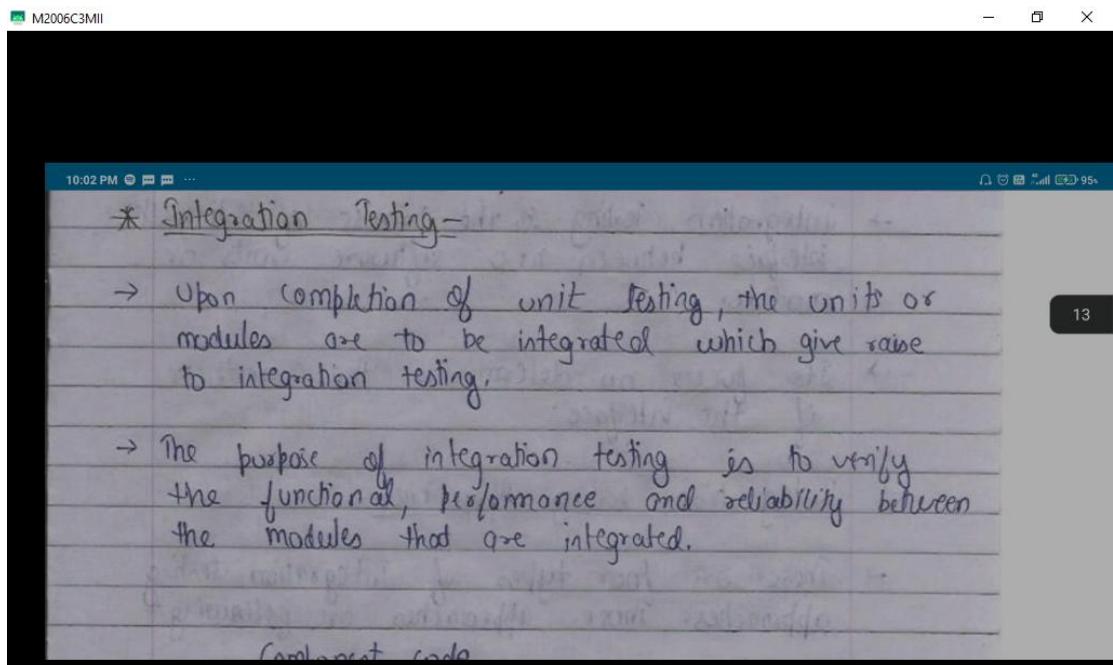


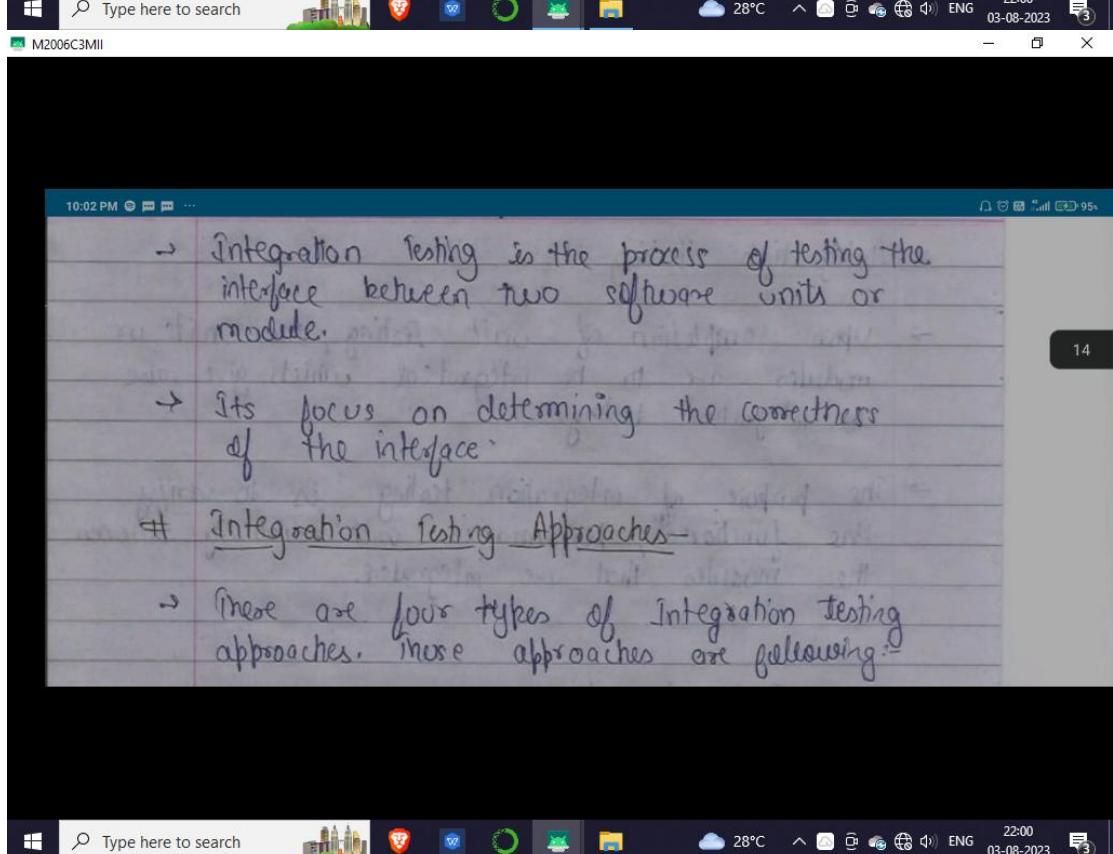
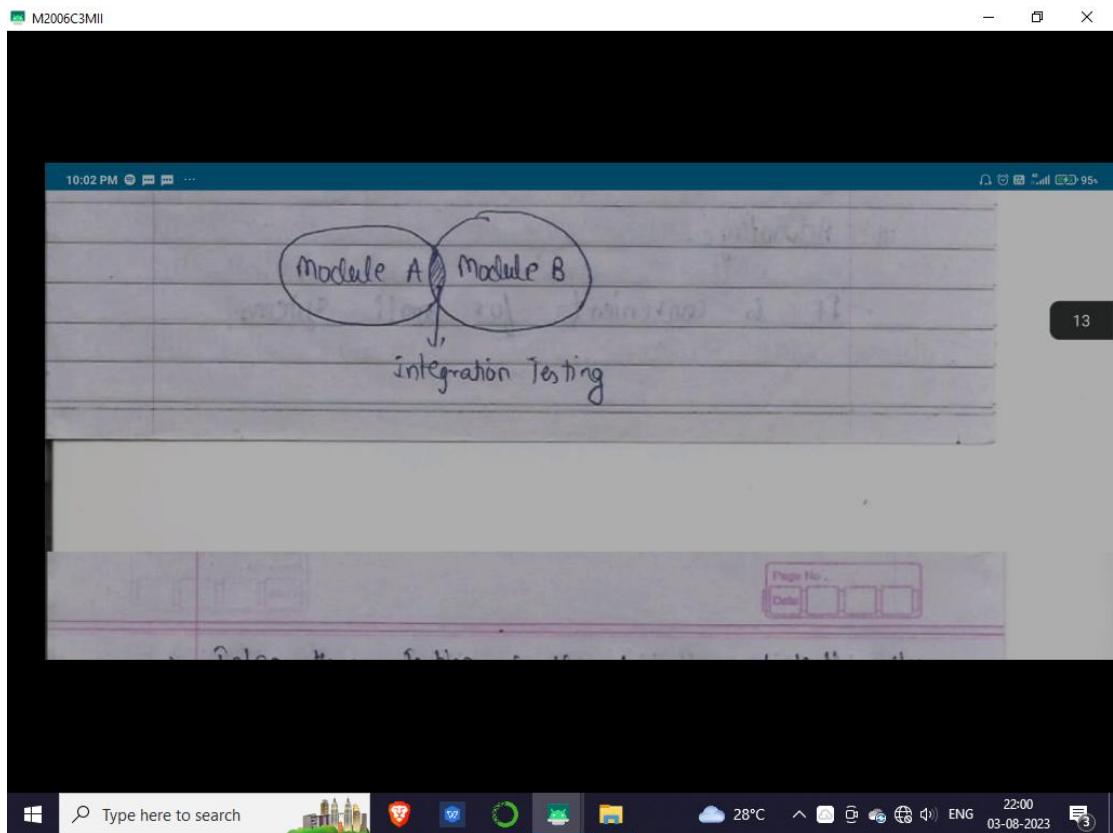


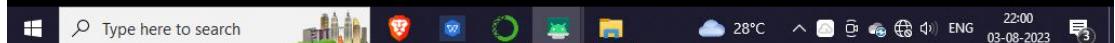
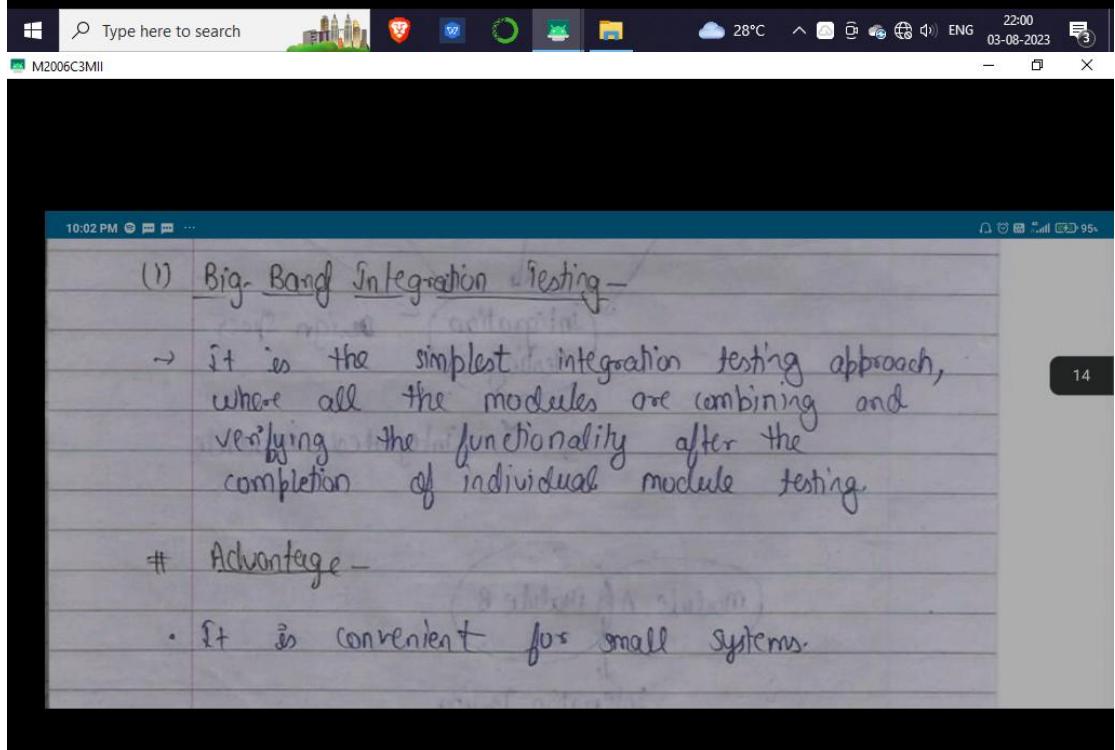
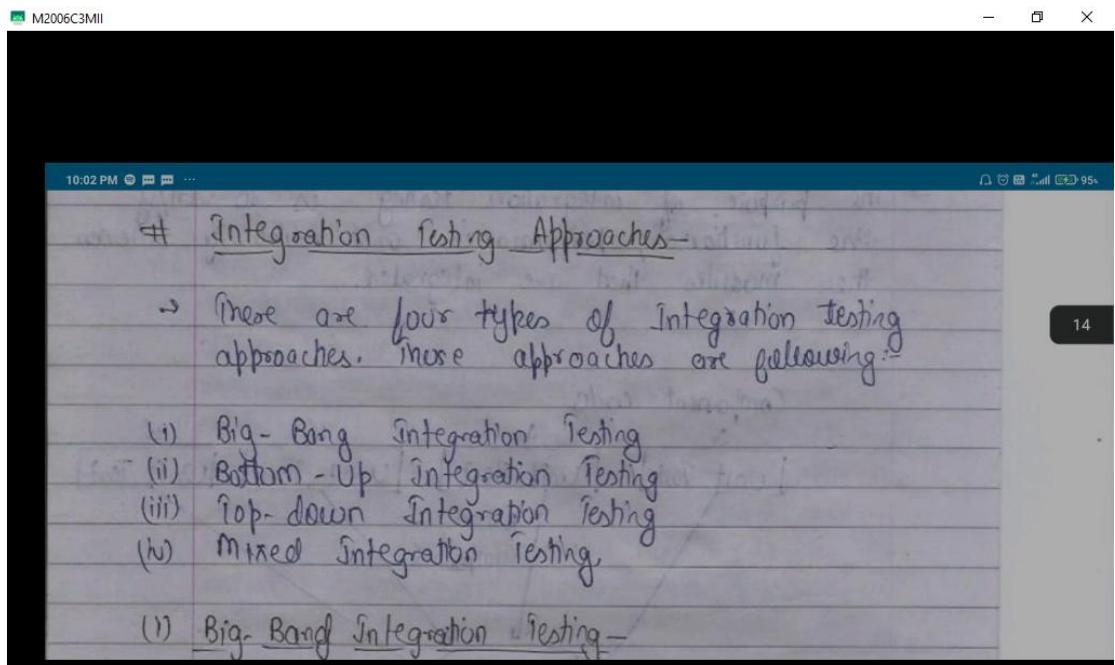


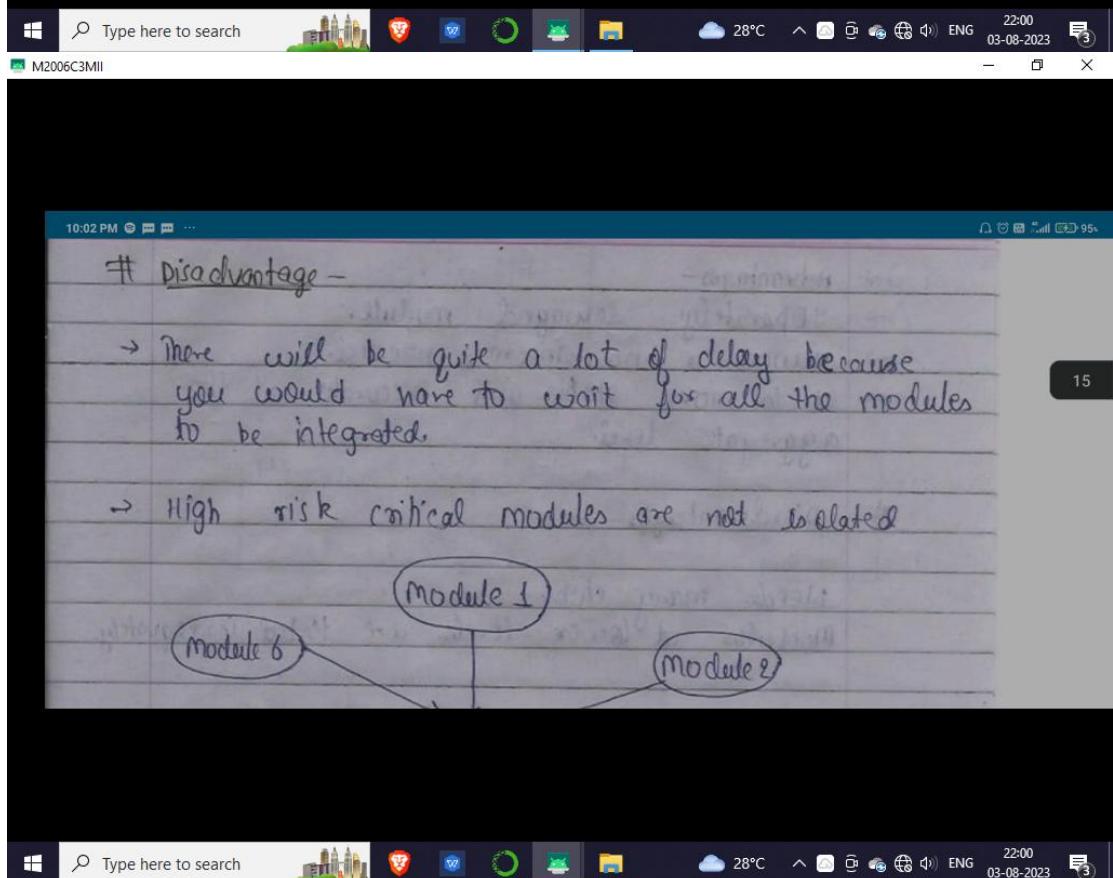
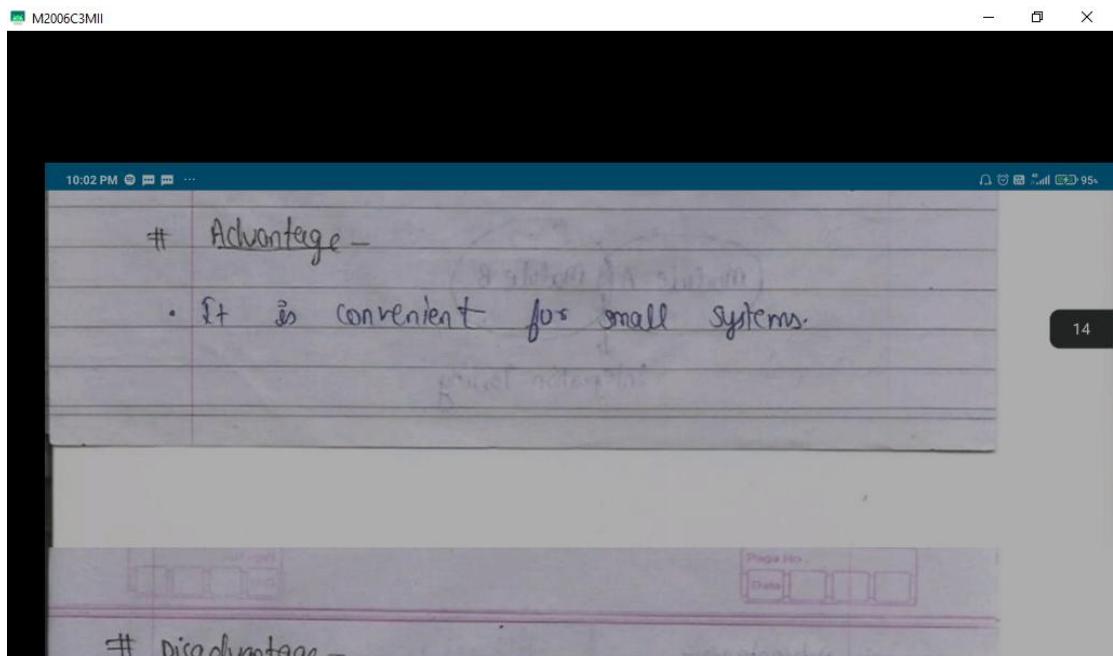


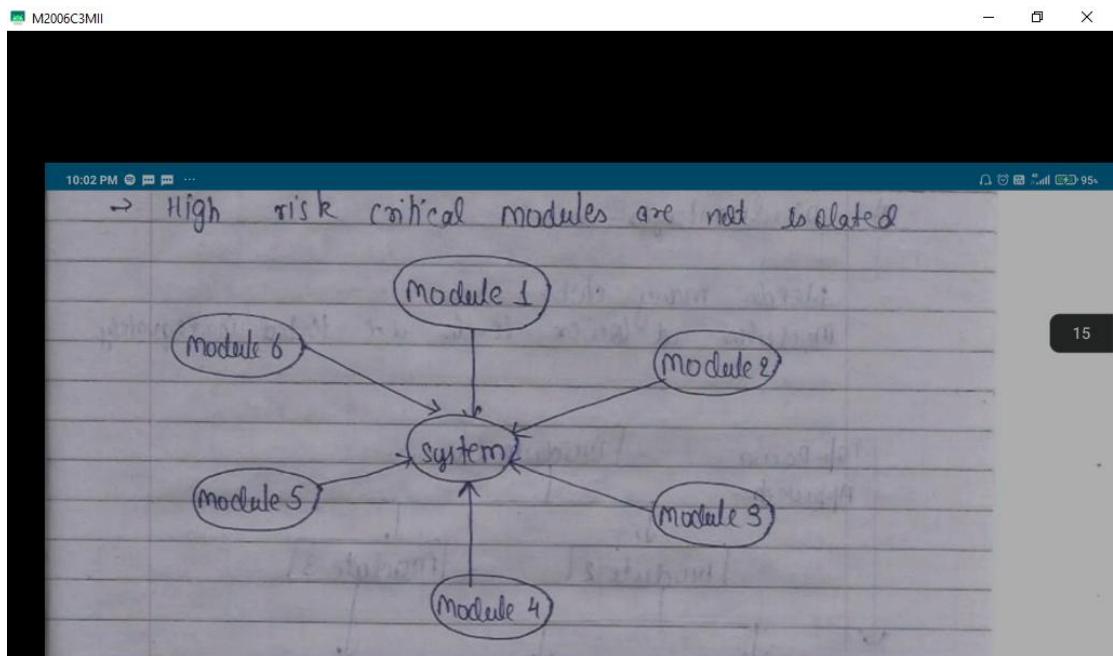




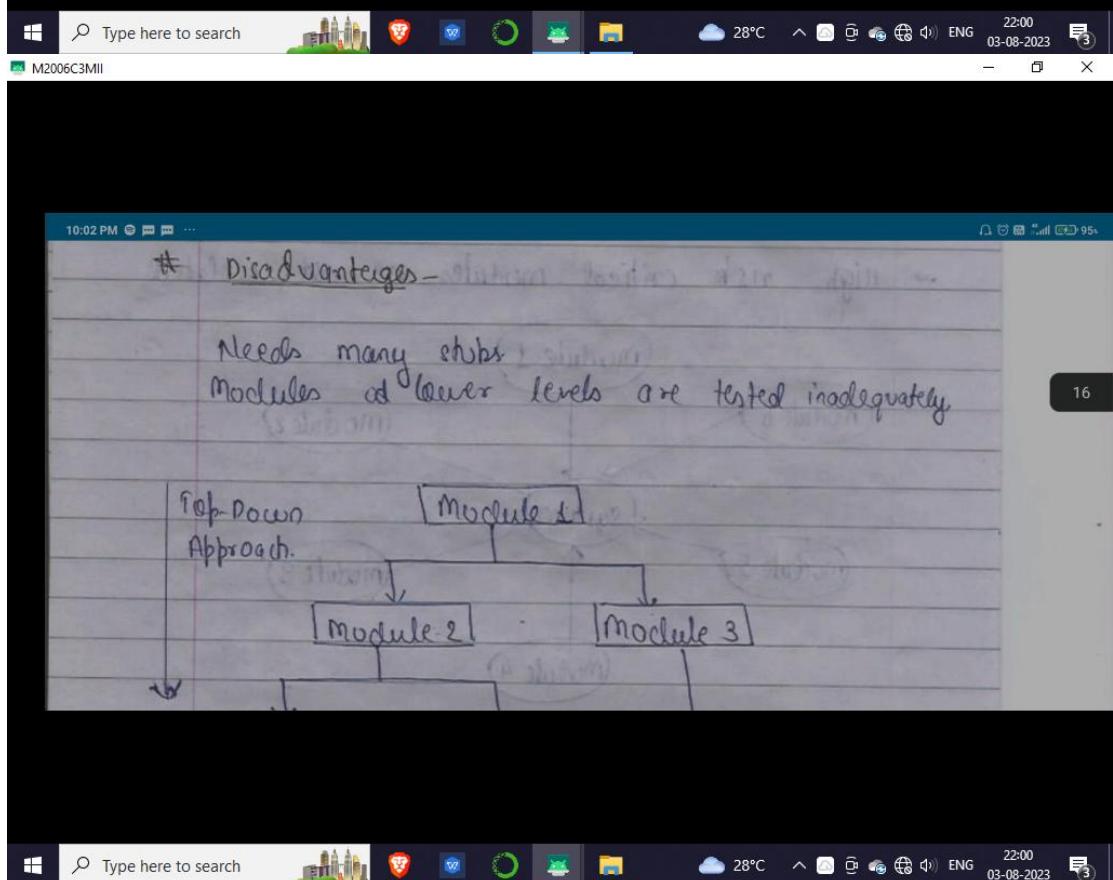
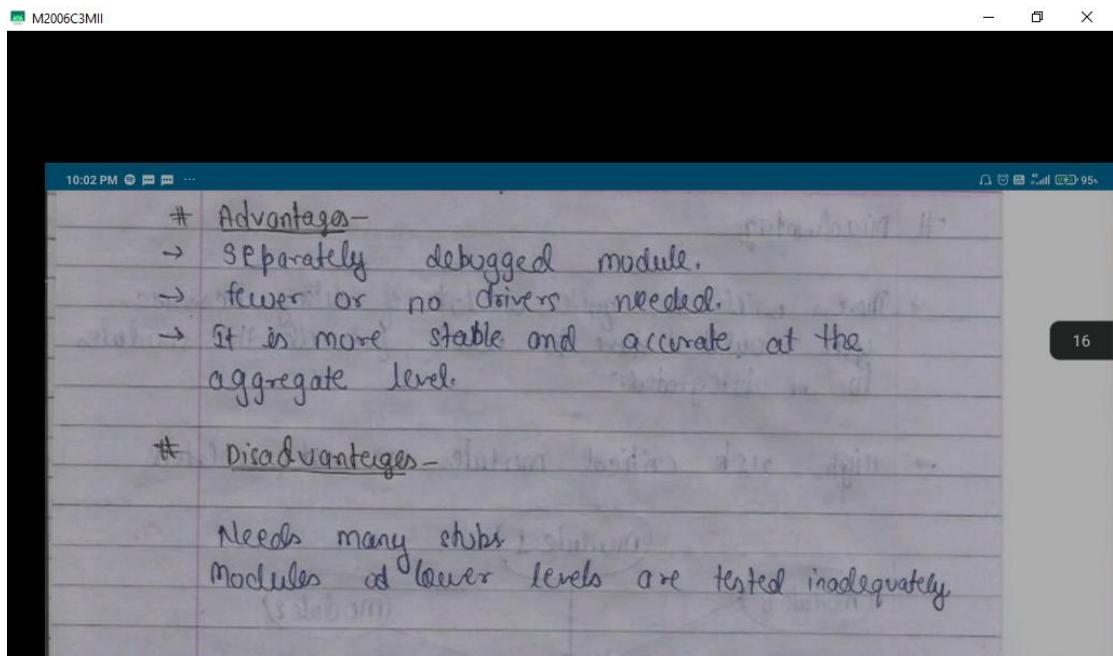


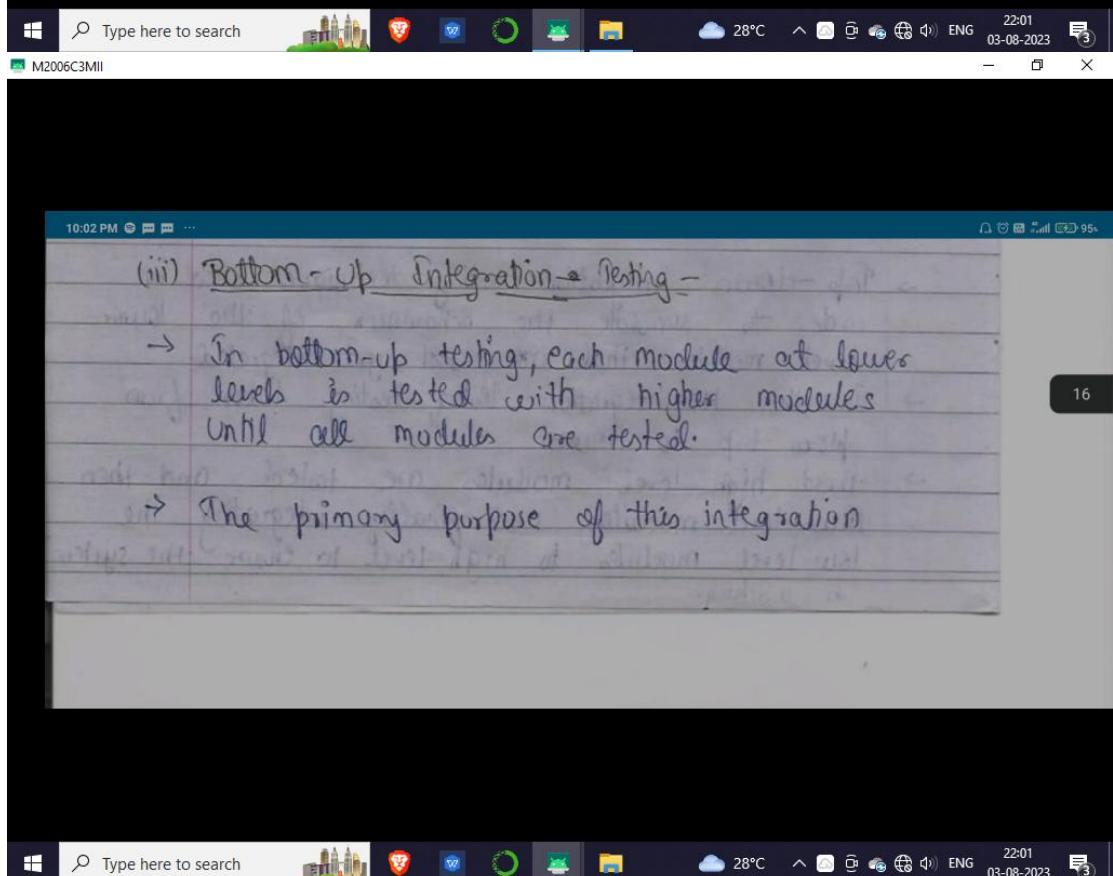
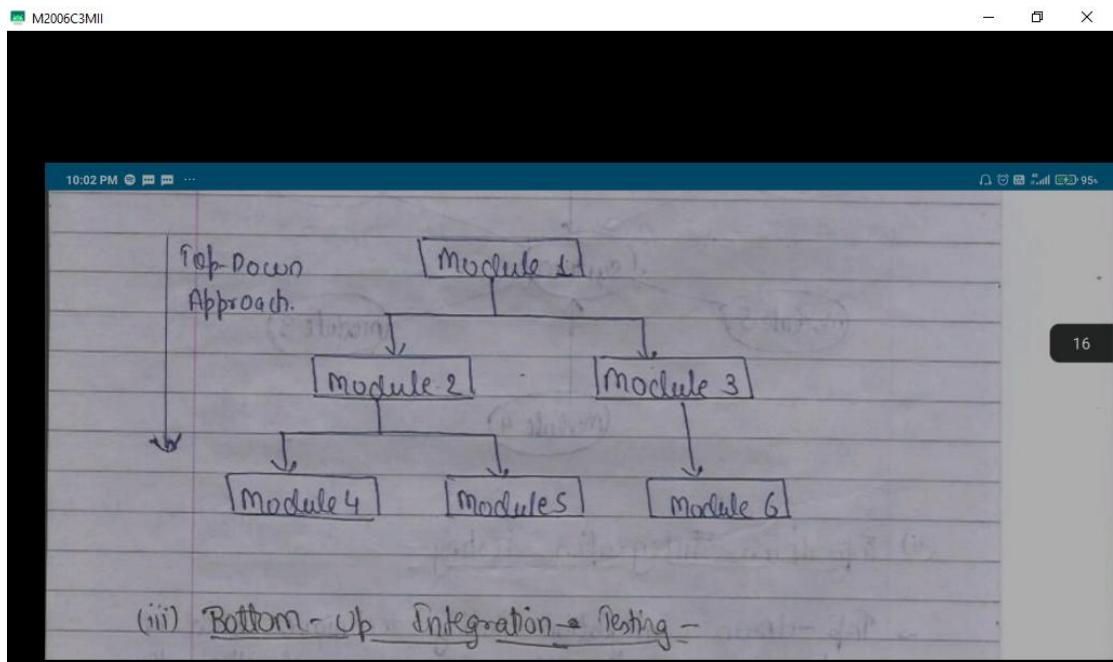


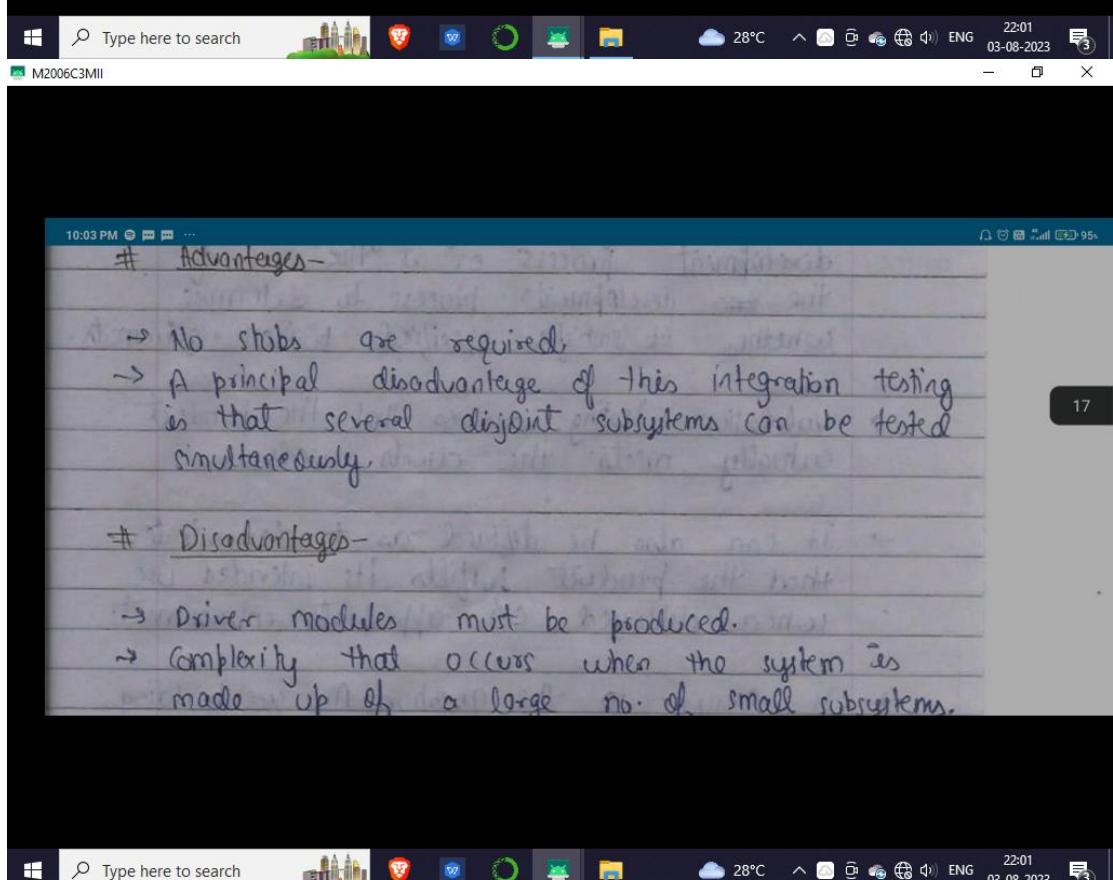
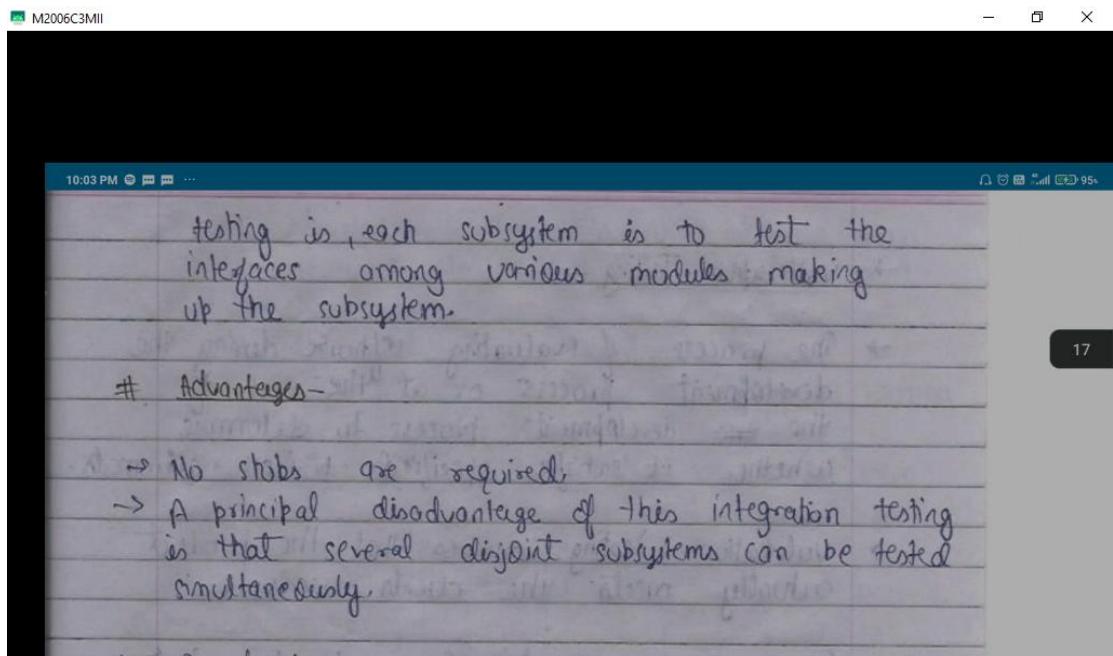


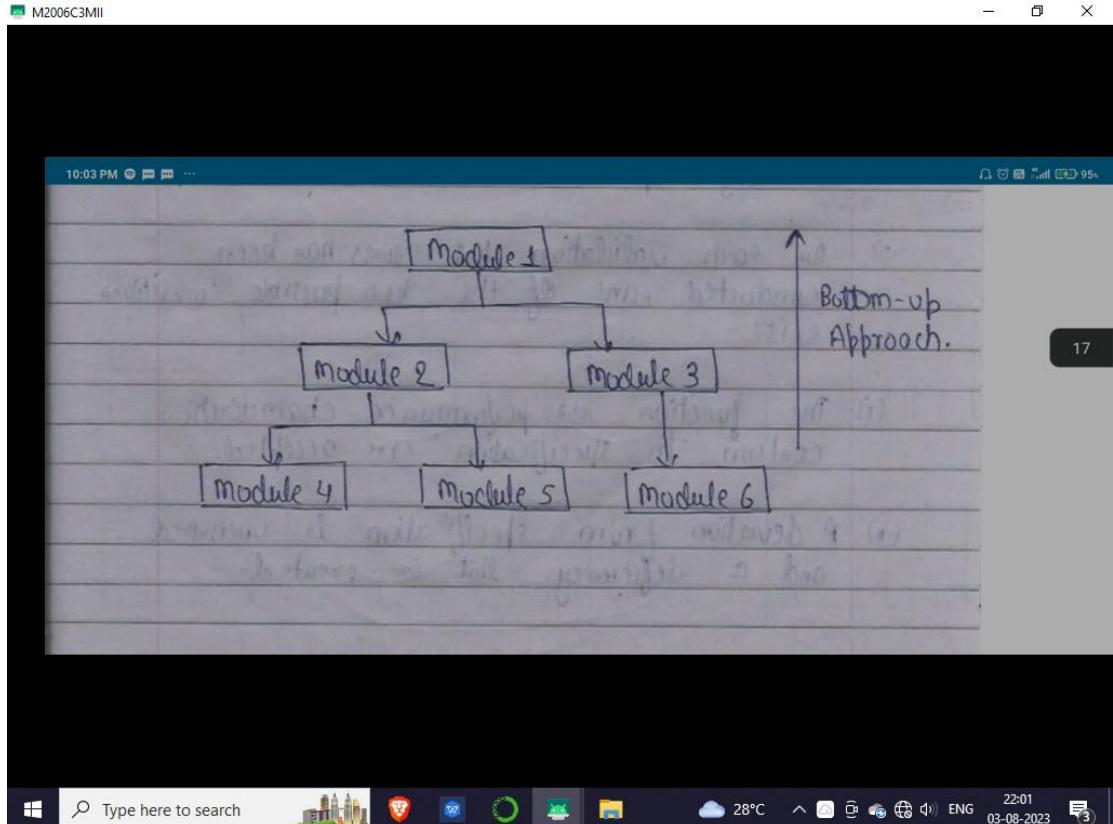
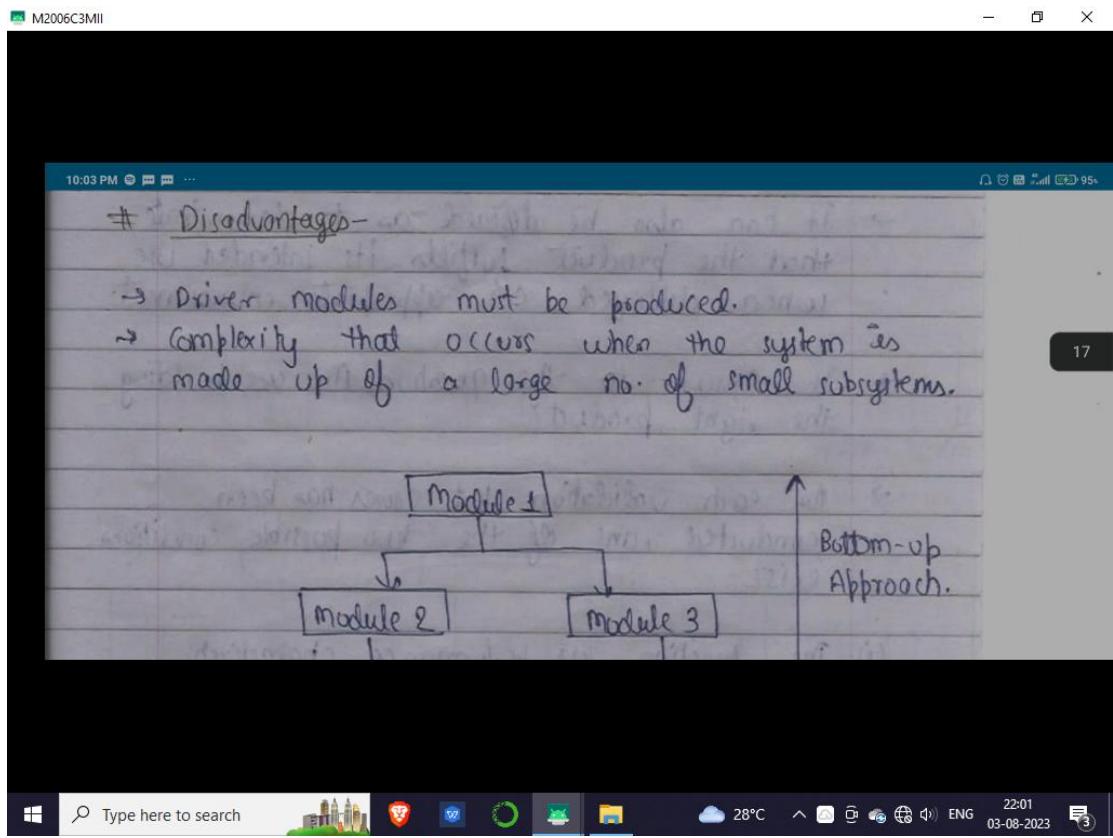


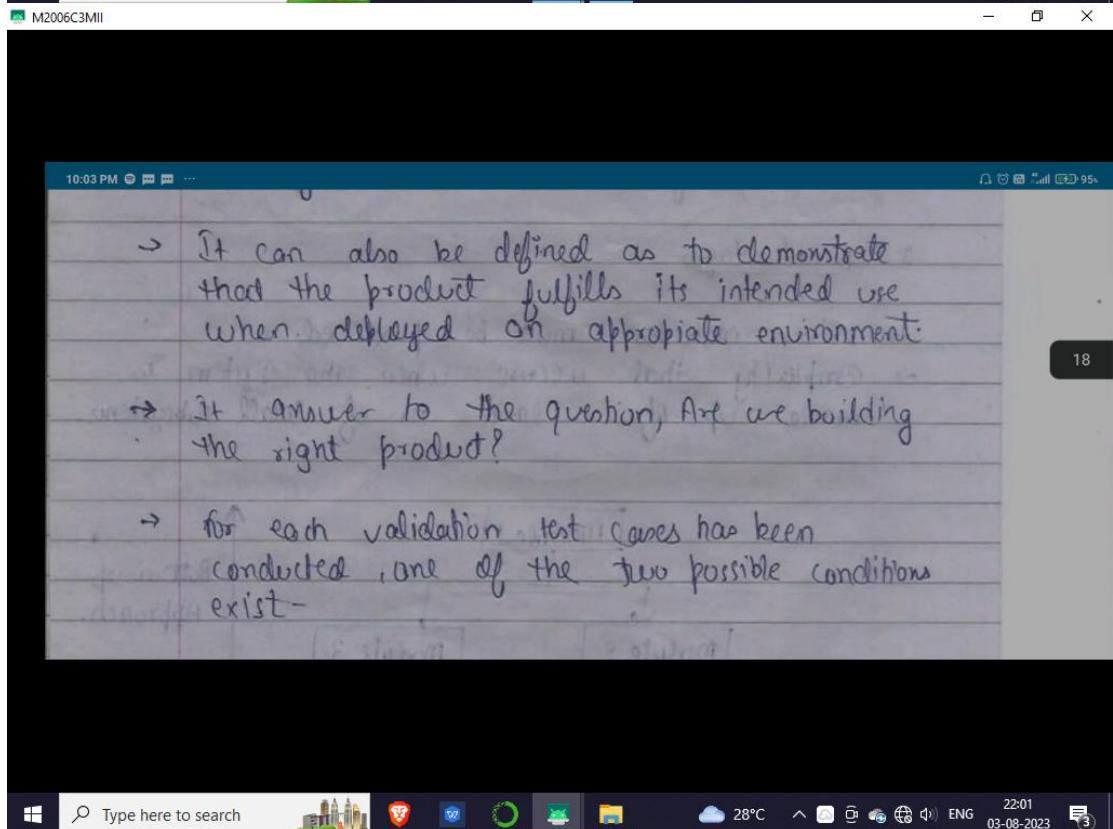
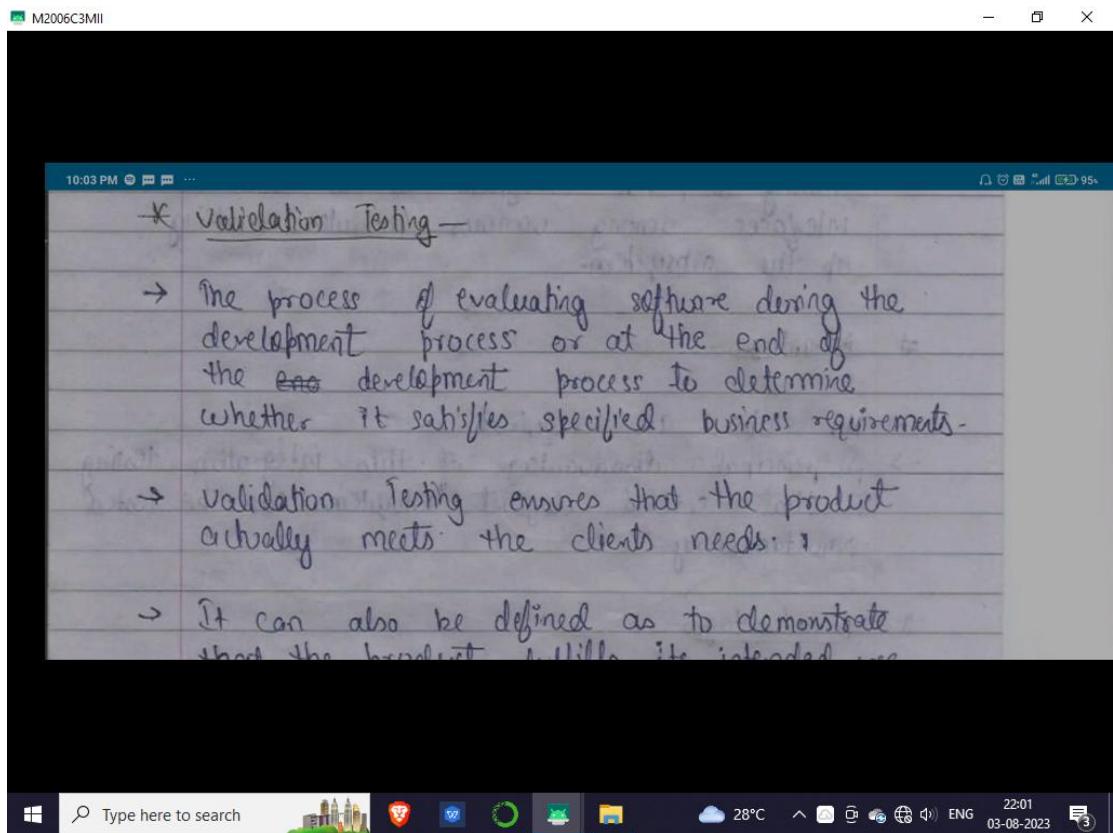
- (ii) Top-down Integration testing =
- Top-down integration testing technique used in order to simulate the behaviour of the lower-level modules that are not yet integrated.
 - In this ~~to~~ integration testing, testing takes from ~~here~~ top to down.
 - first high-level modules are tested and then low-level modules and finally integrating the low-level modules to high-level to ensure the system is working.

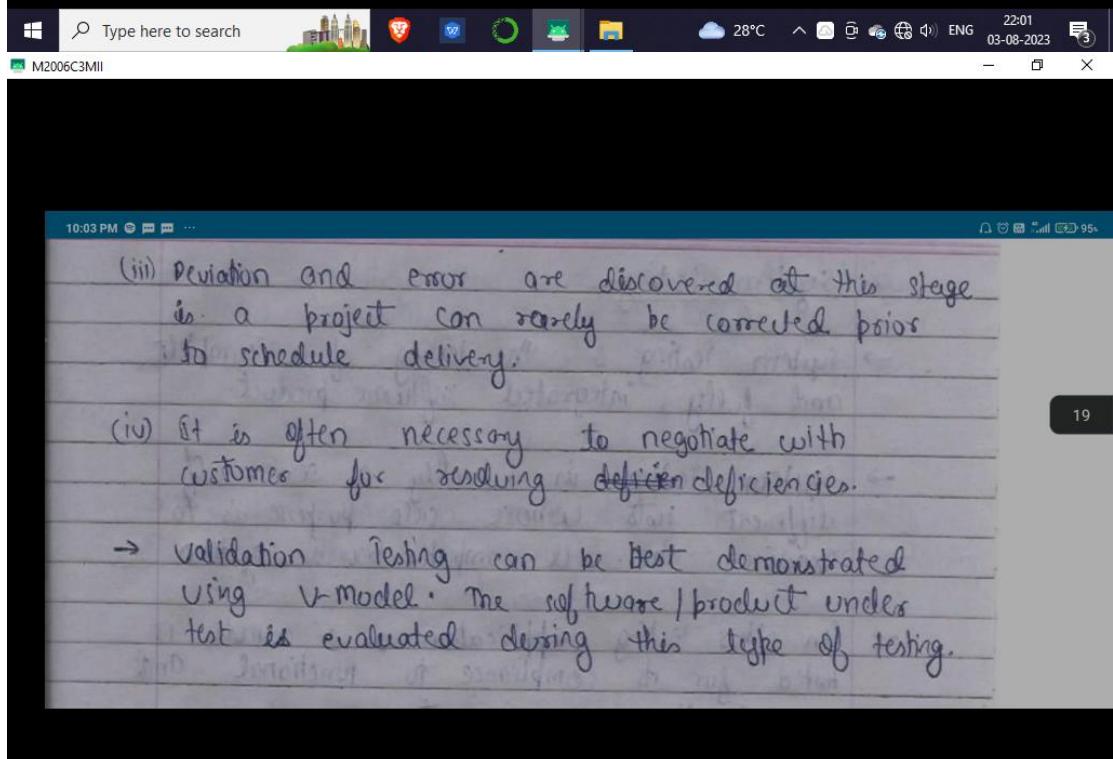
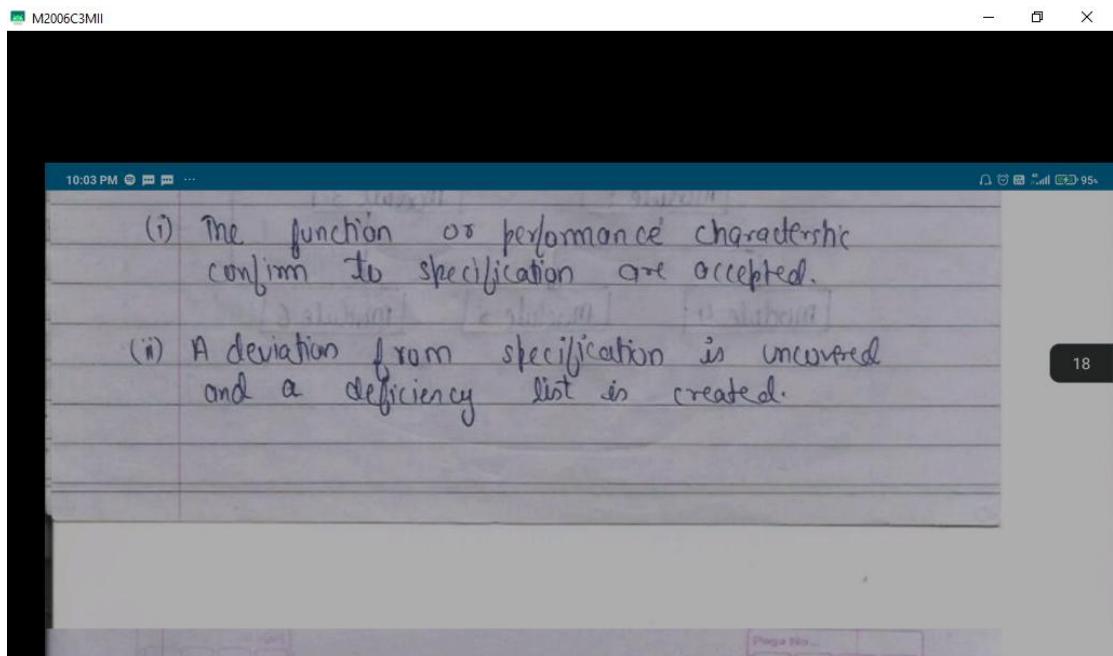


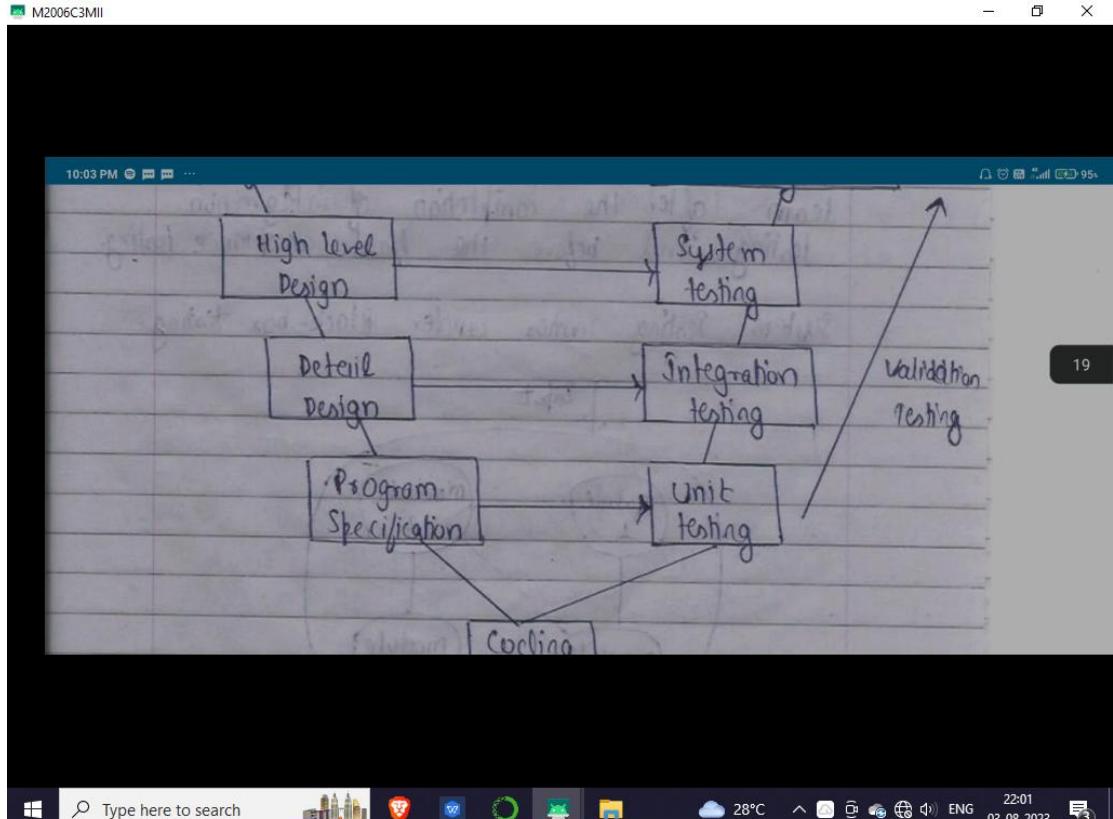
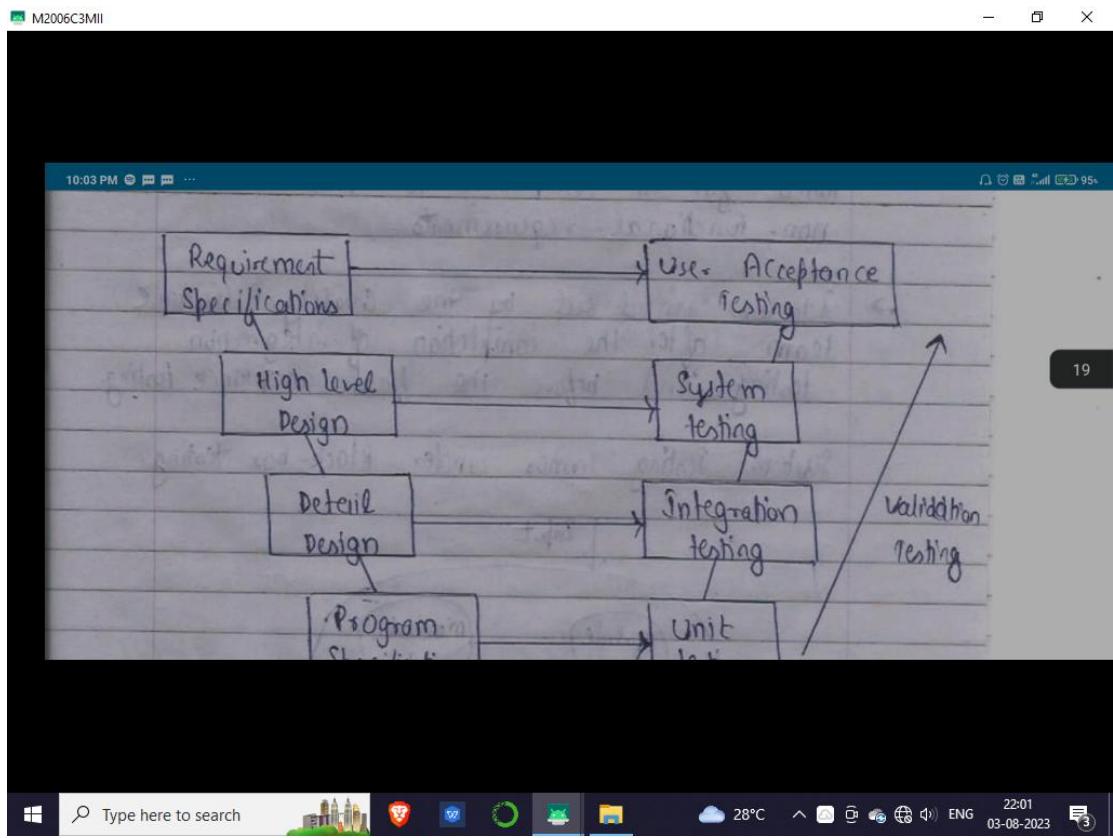


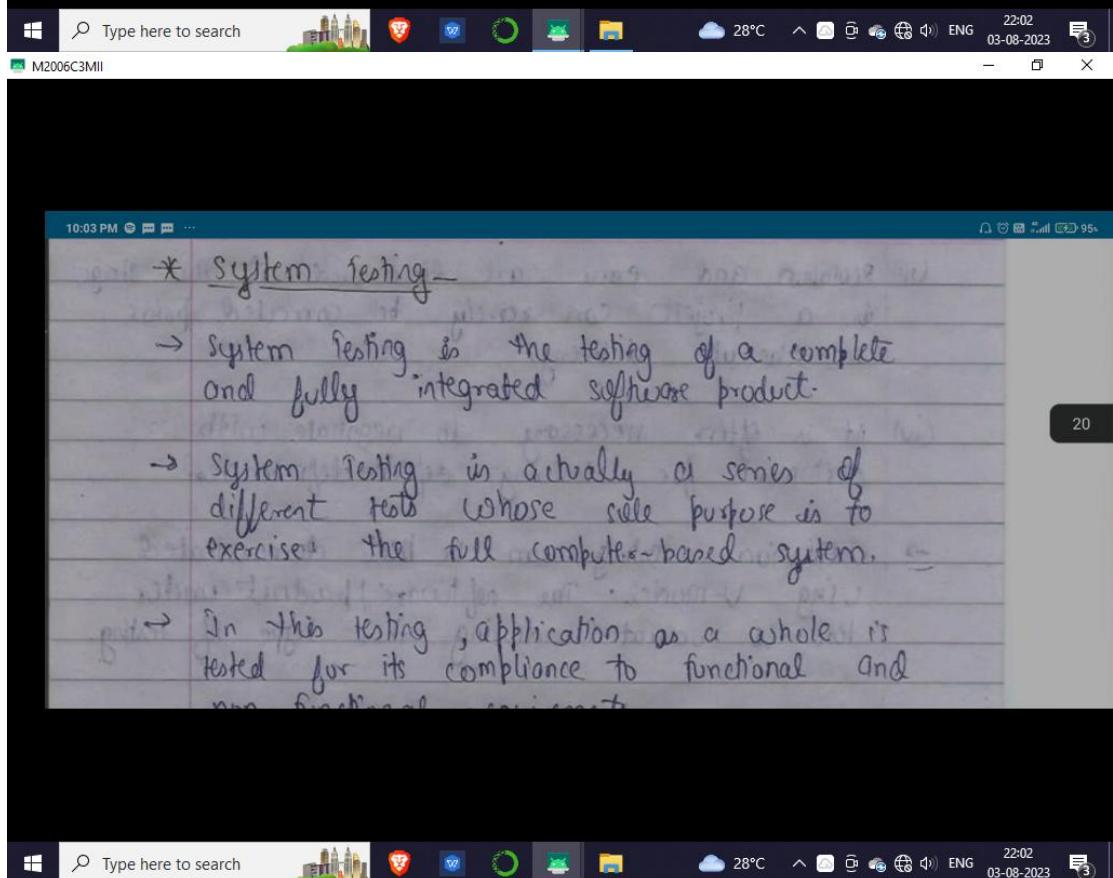
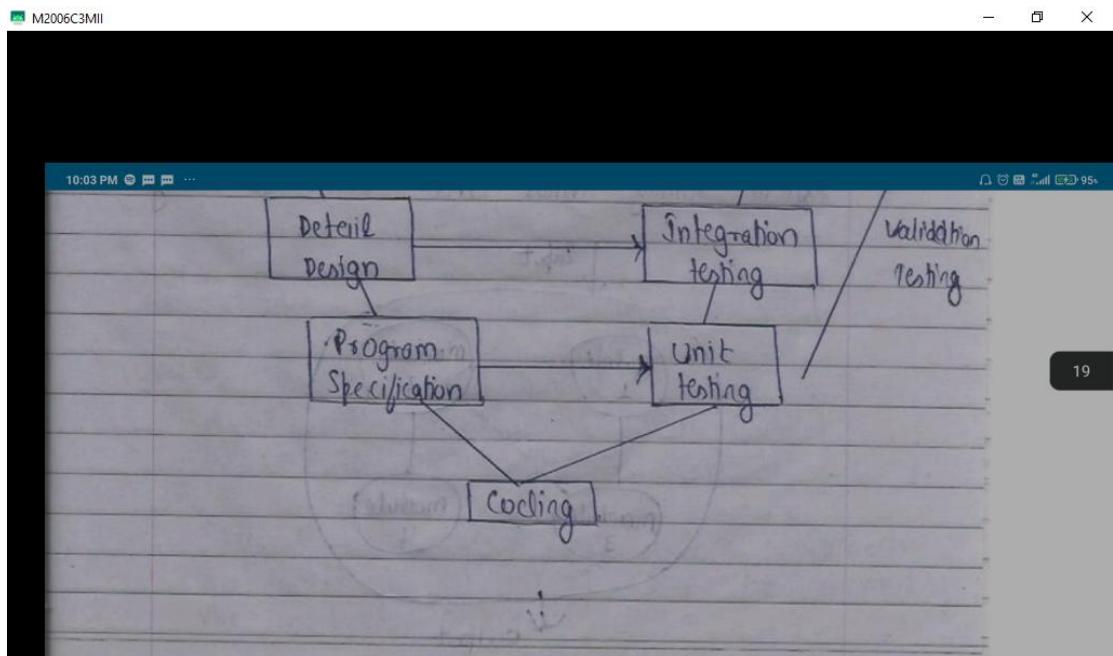


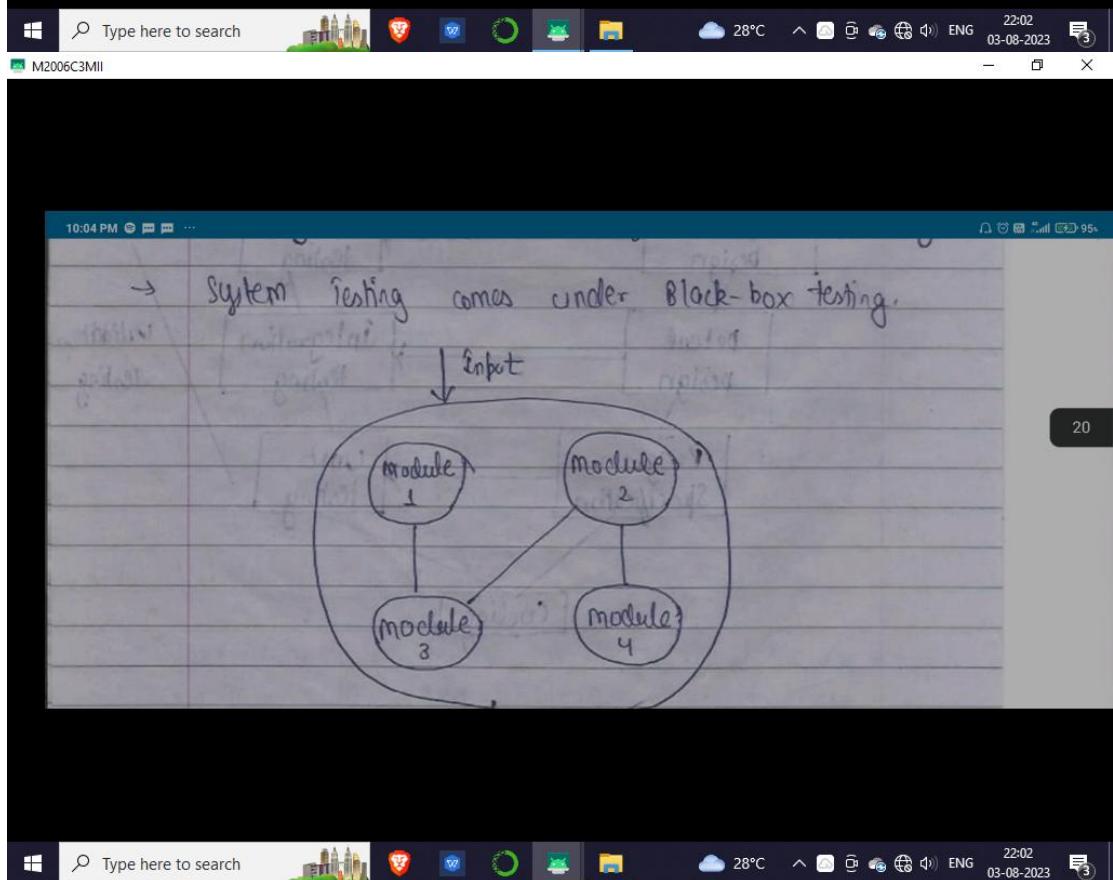
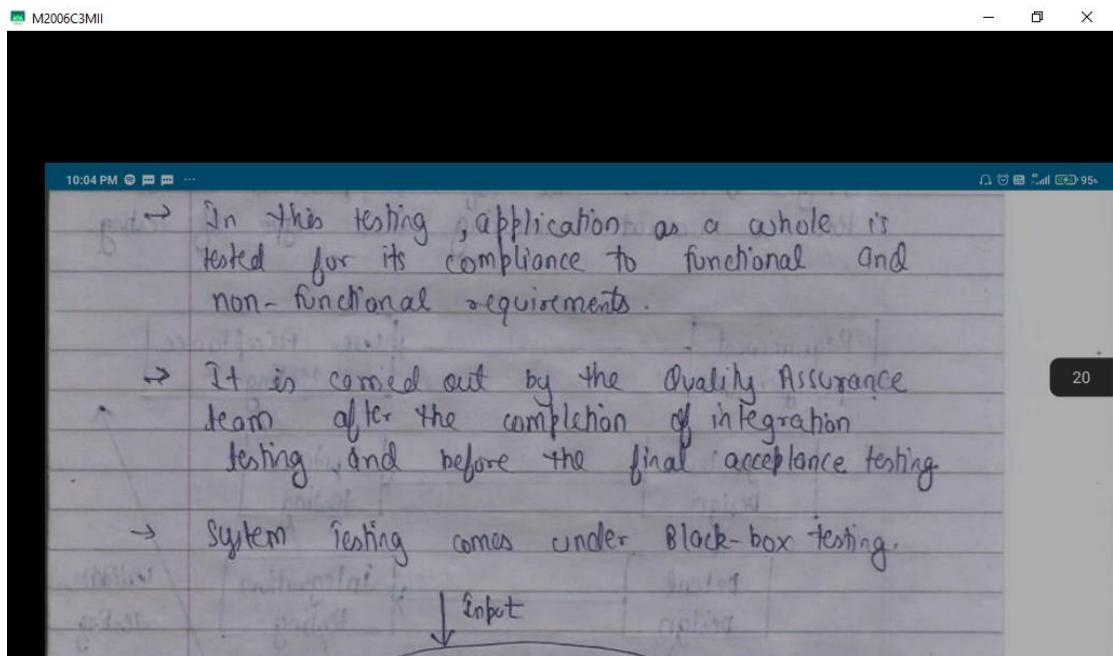


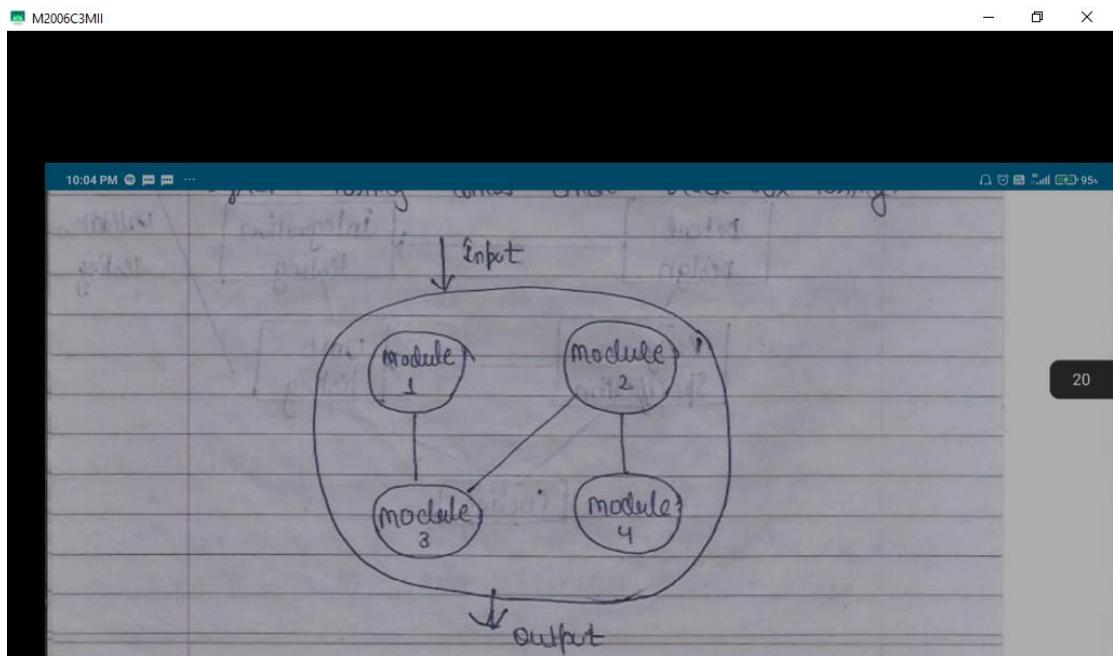




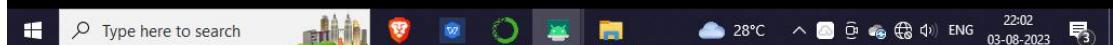
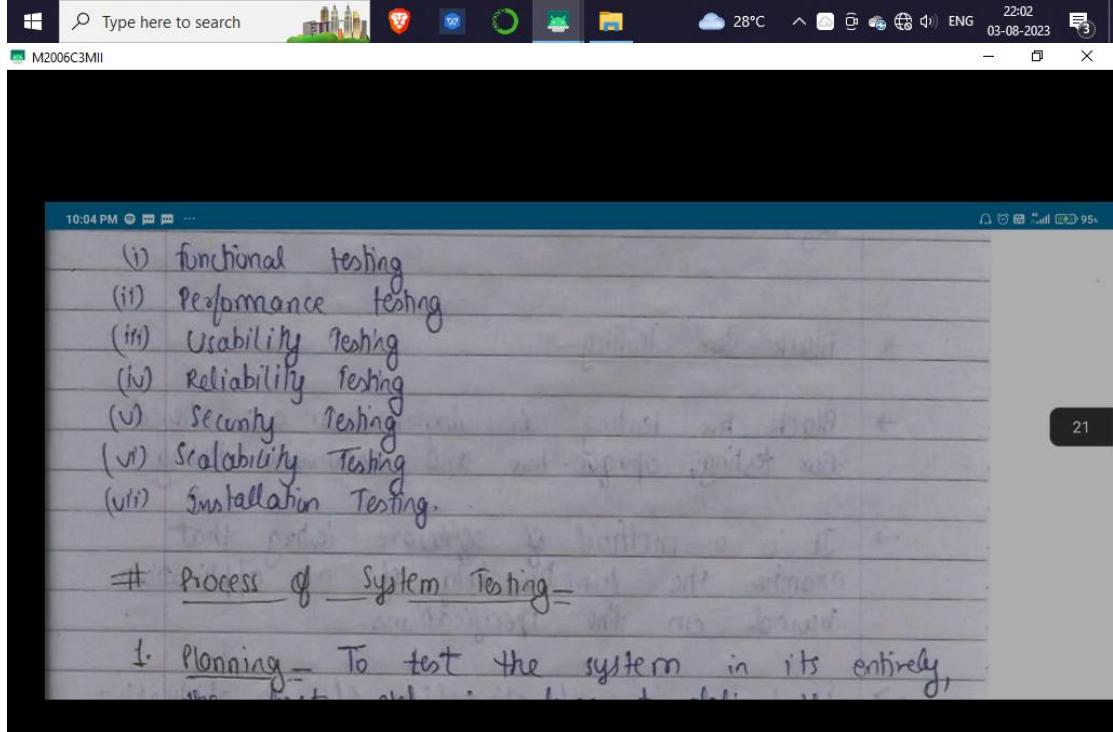
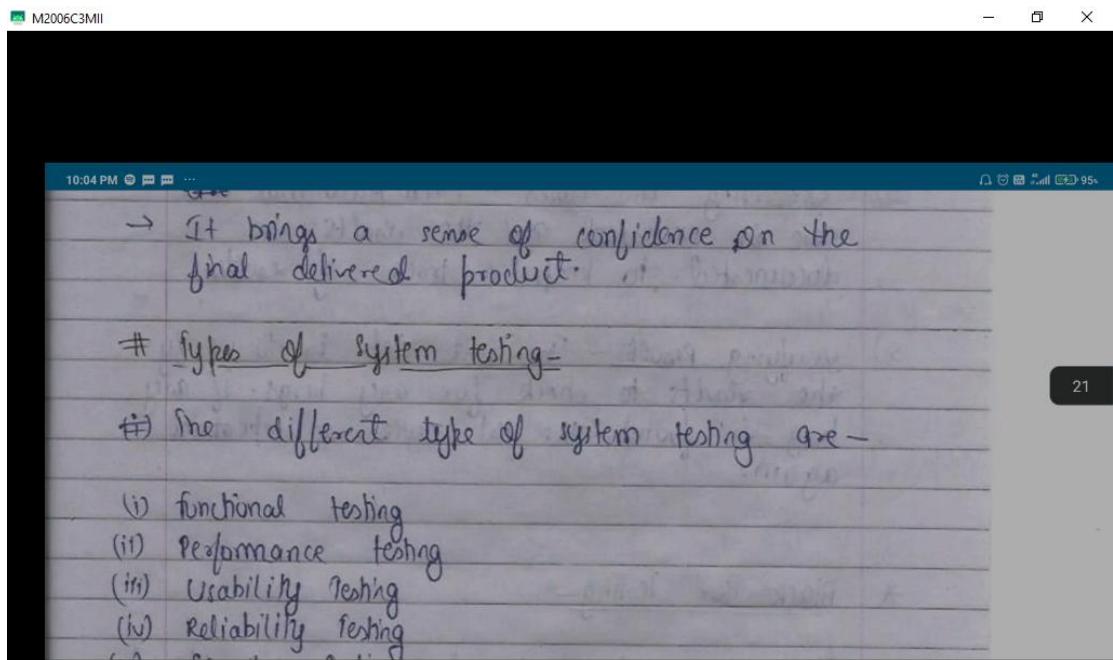


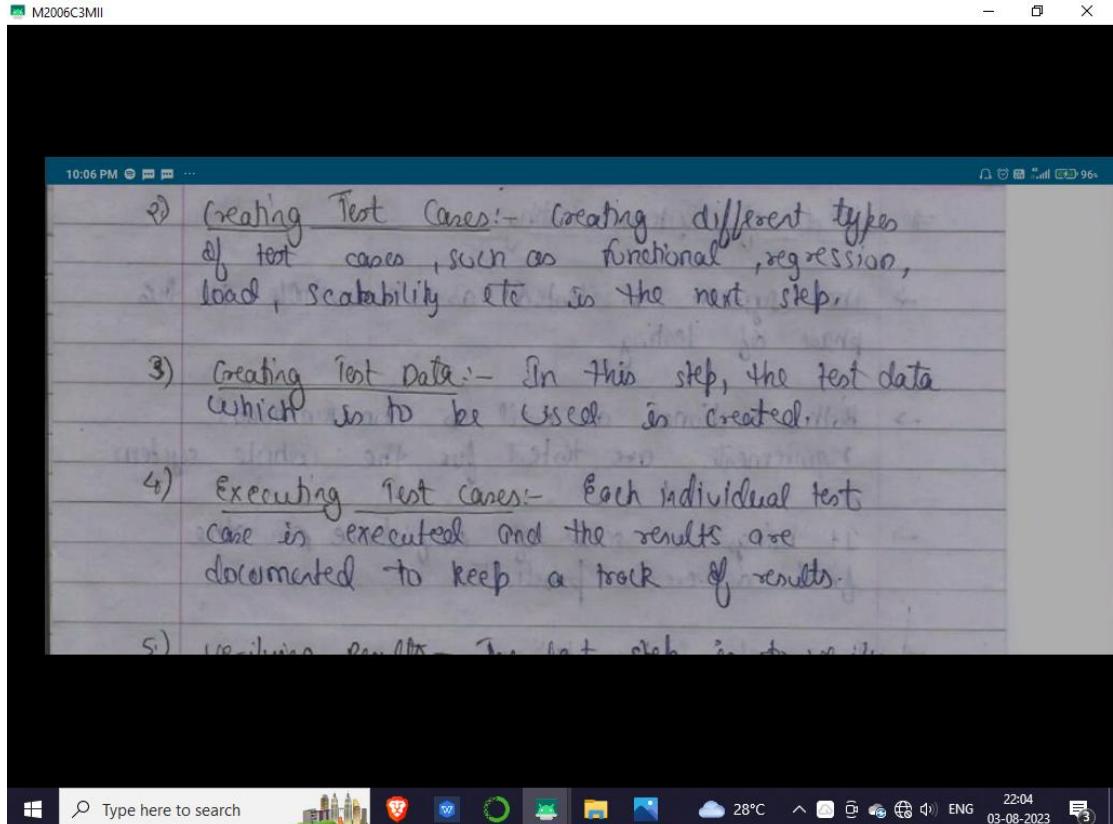
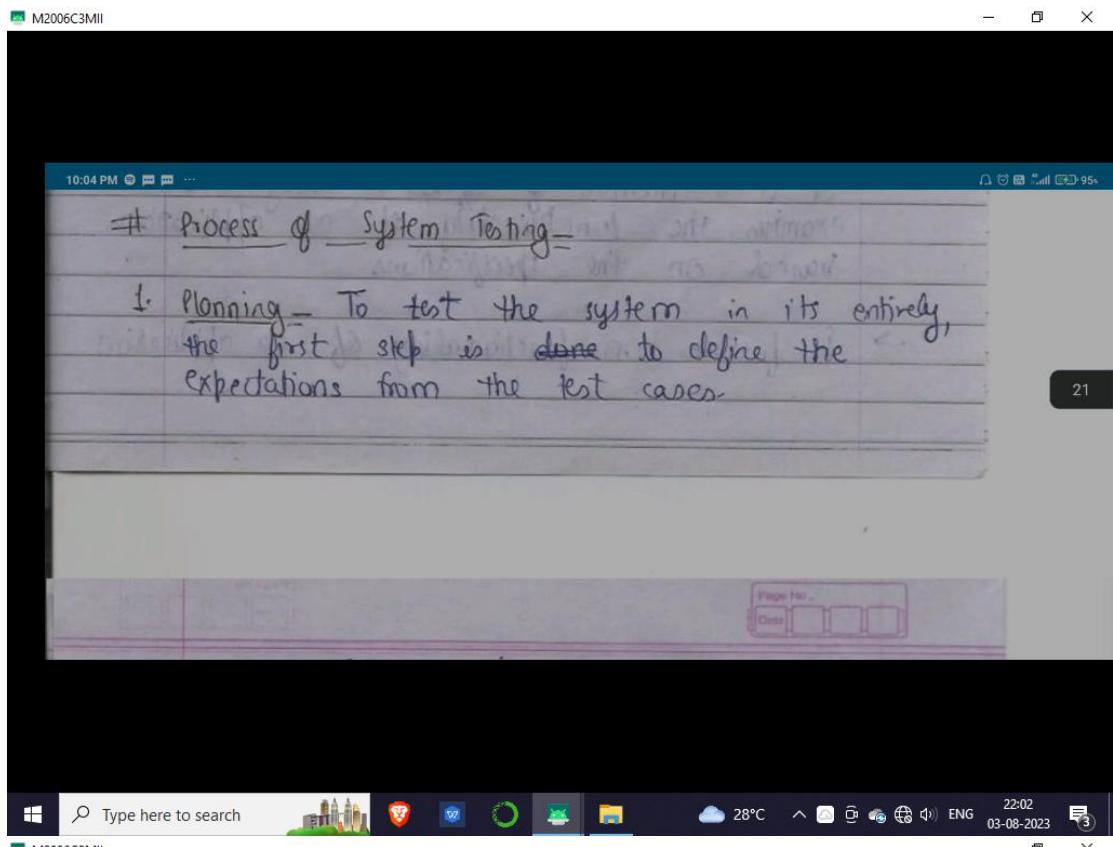


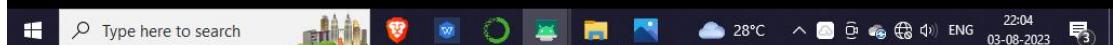
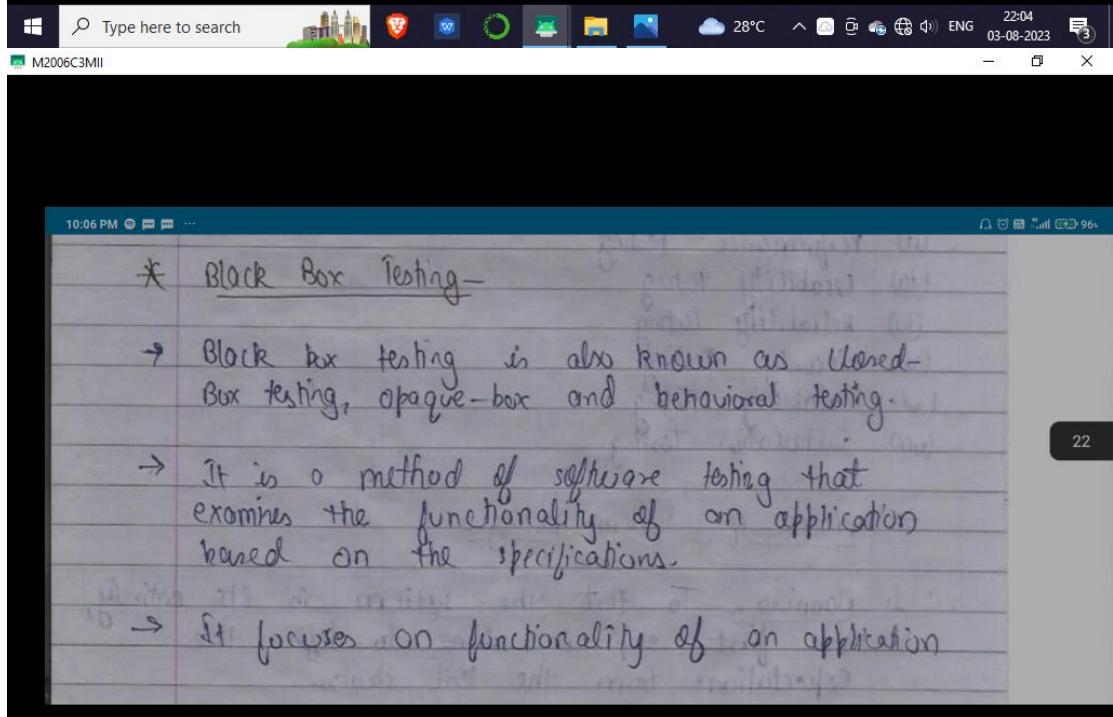
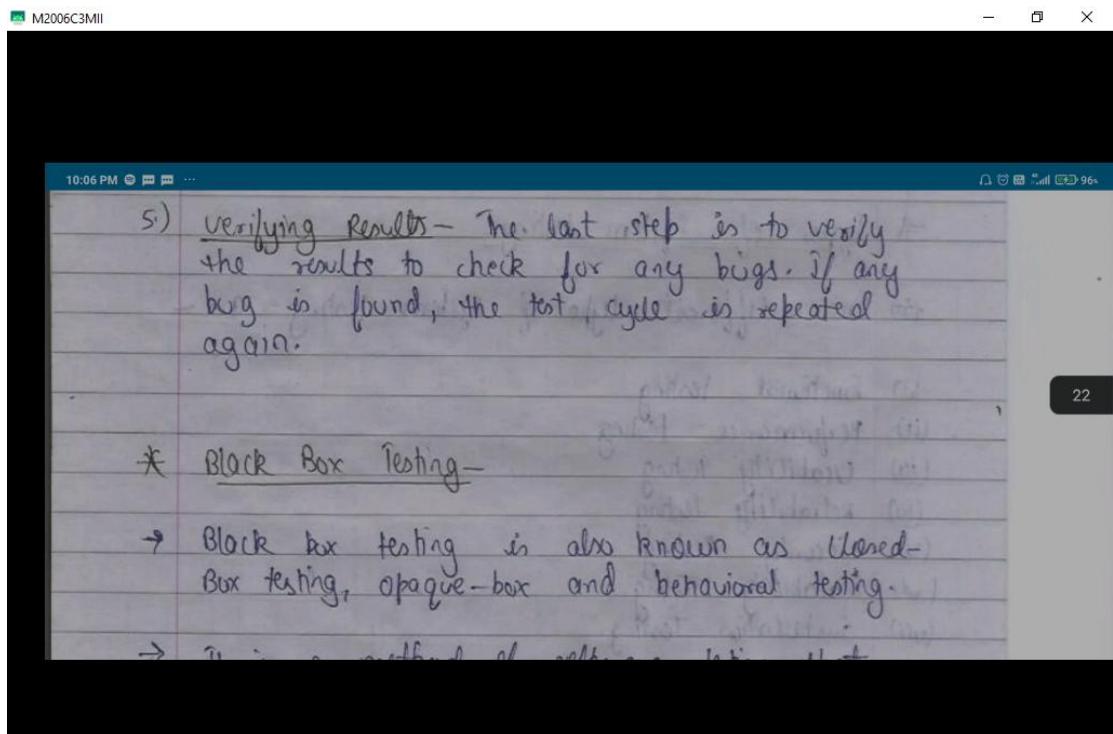


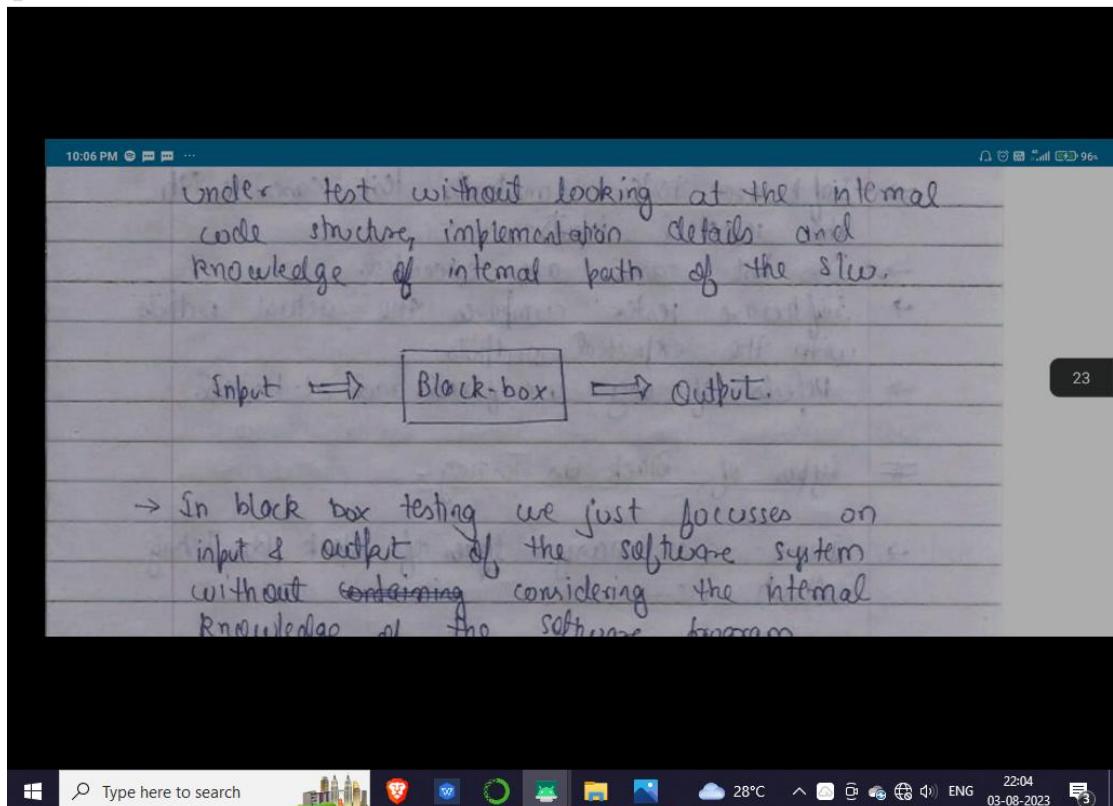
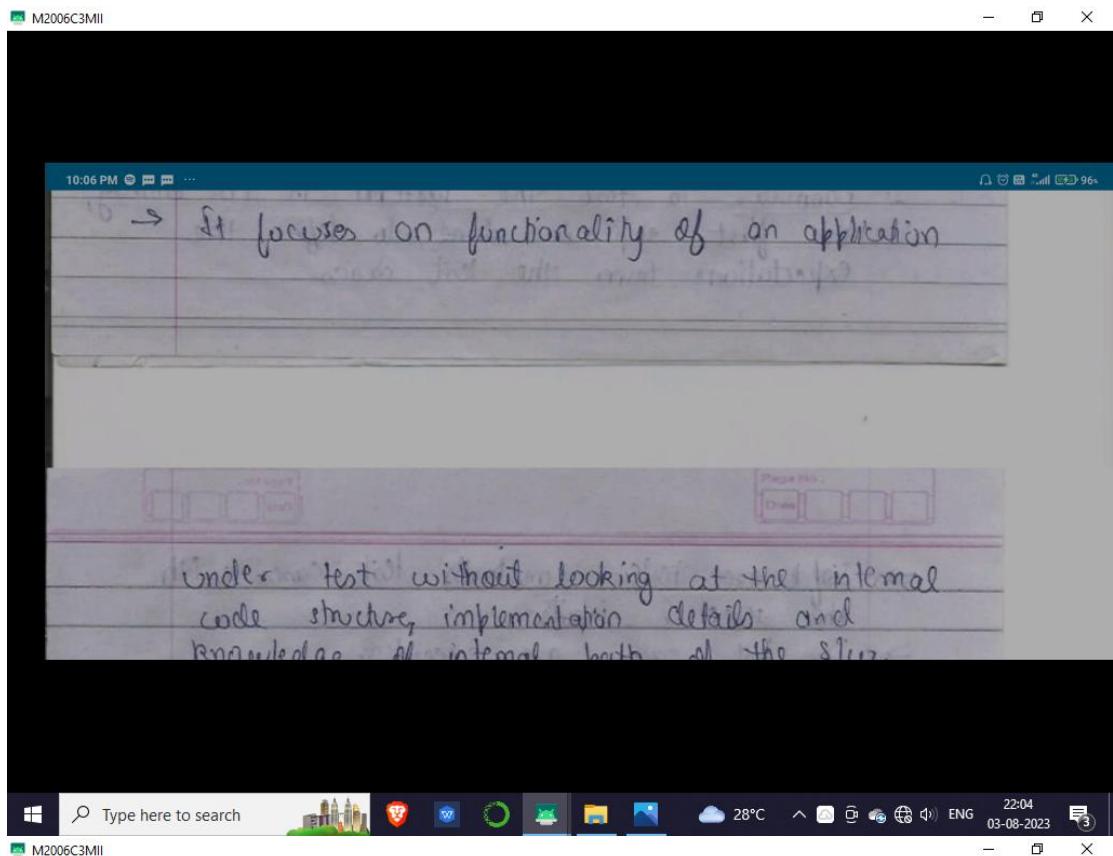


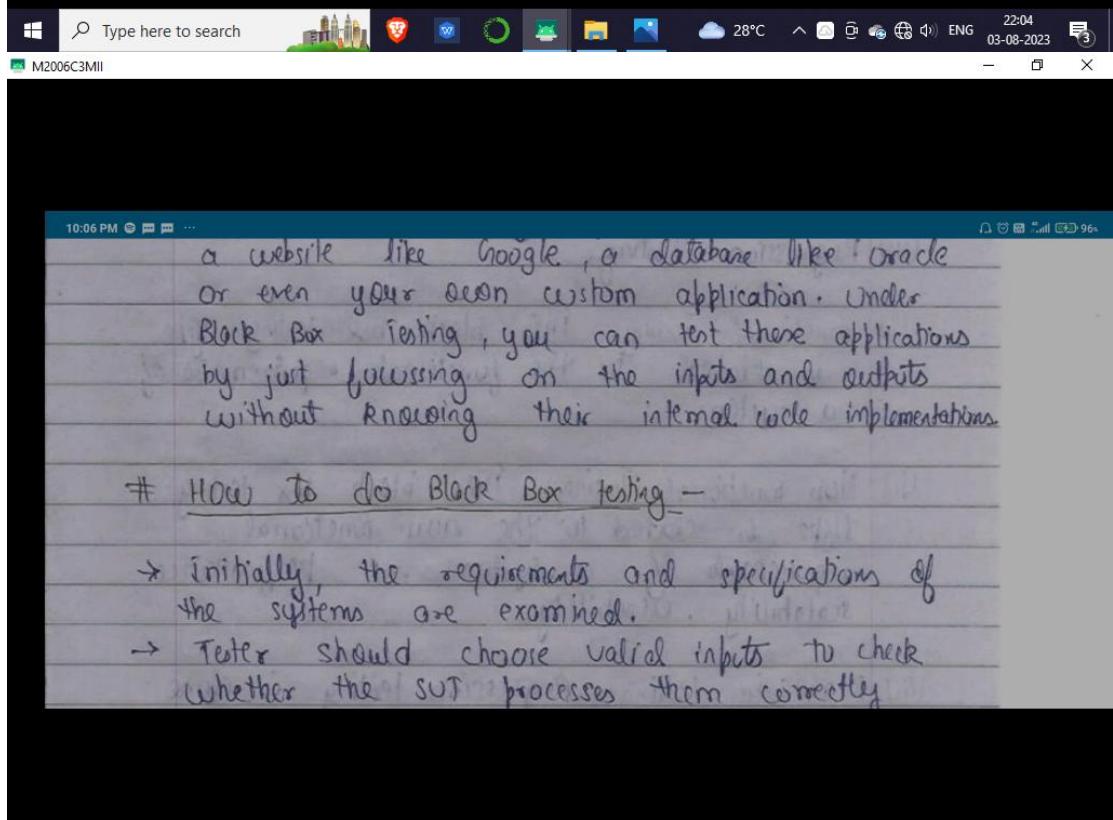
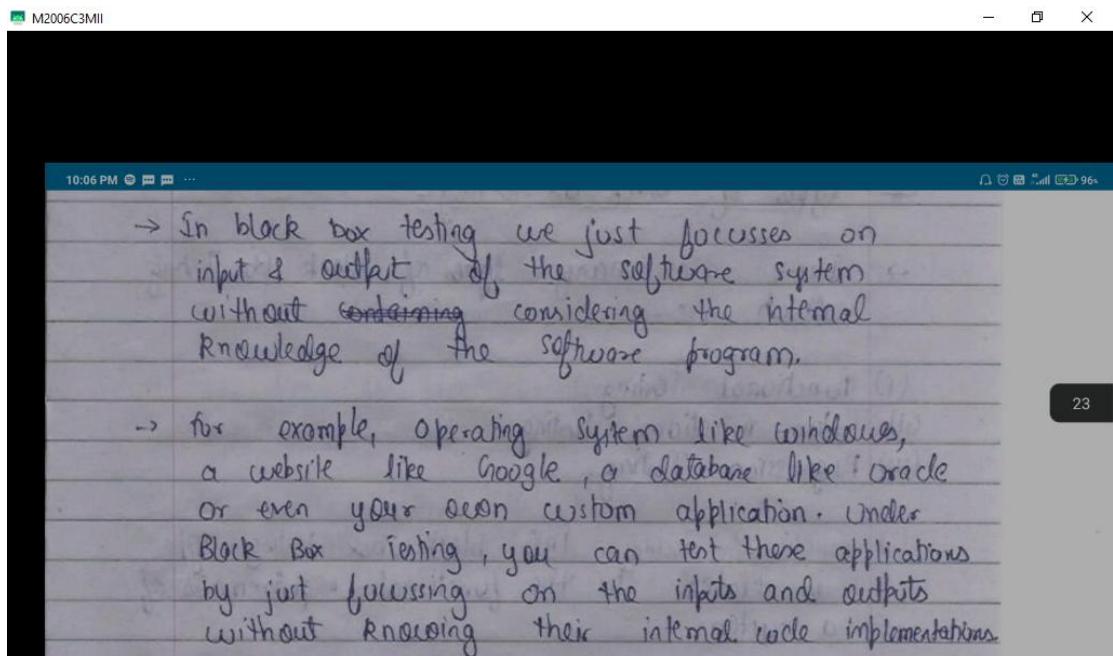
- 21
- # Need of System Testing =
- The system is tested as a whole in this phase of testing.
 - Both functional as well as non-functional requirements are tested for the whole system
 - It brings a sense of confidence on the final delivered product.
- # future of System Test

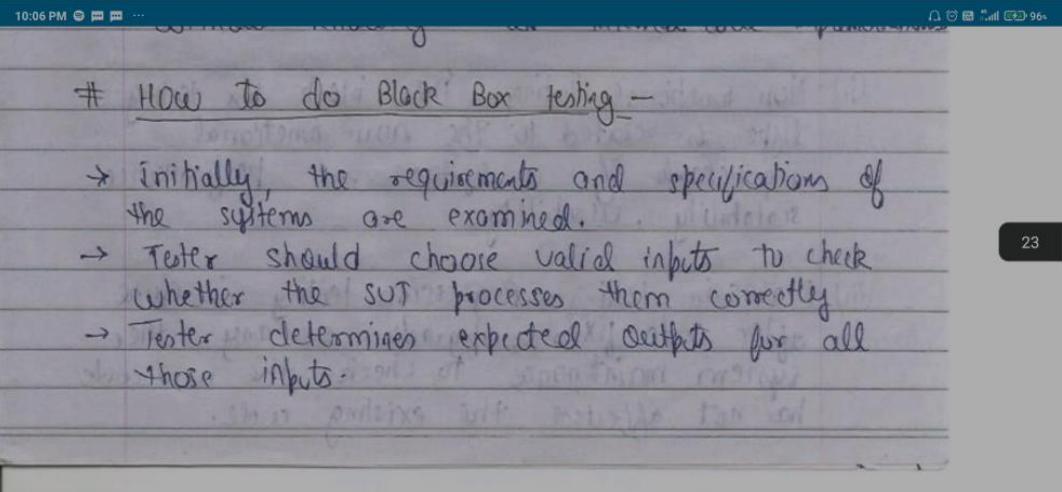




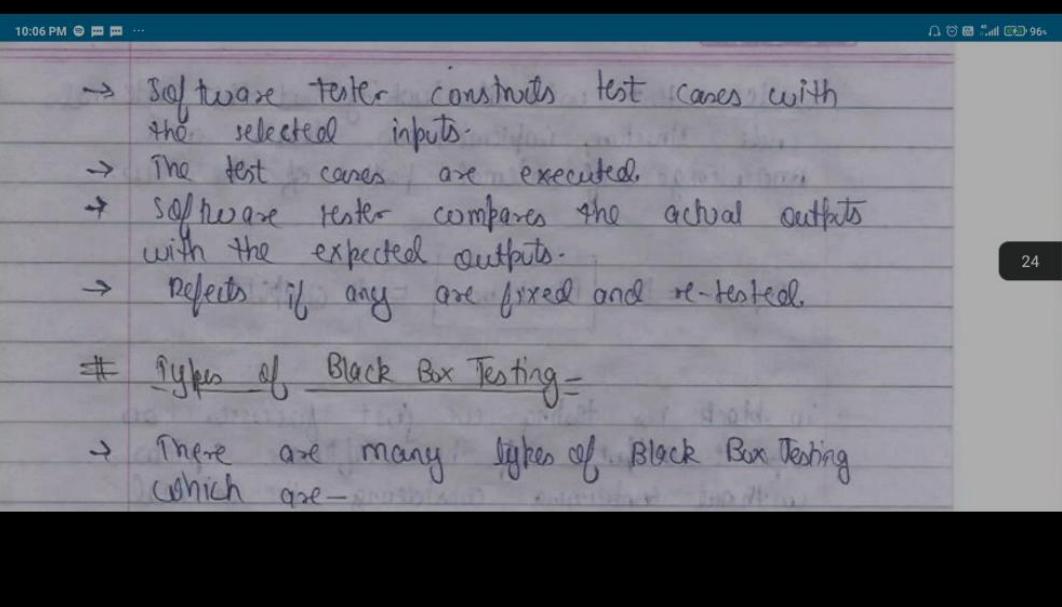




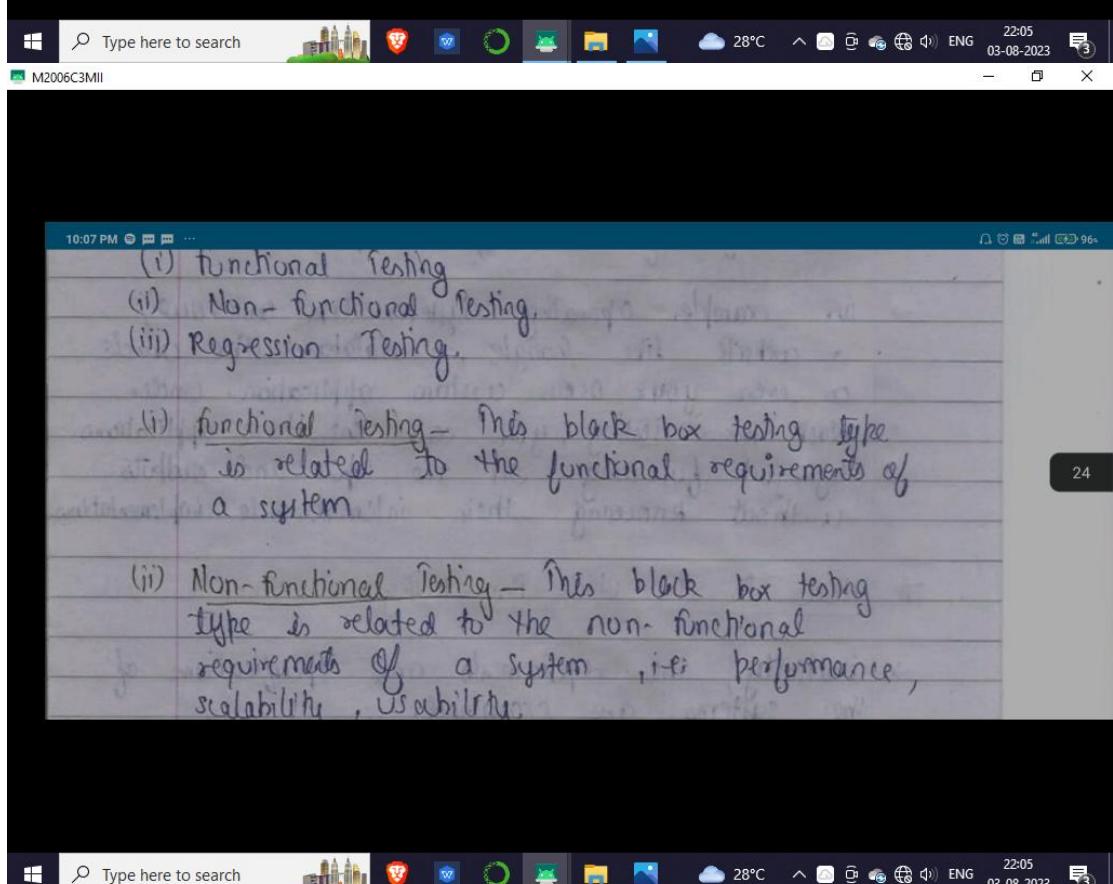
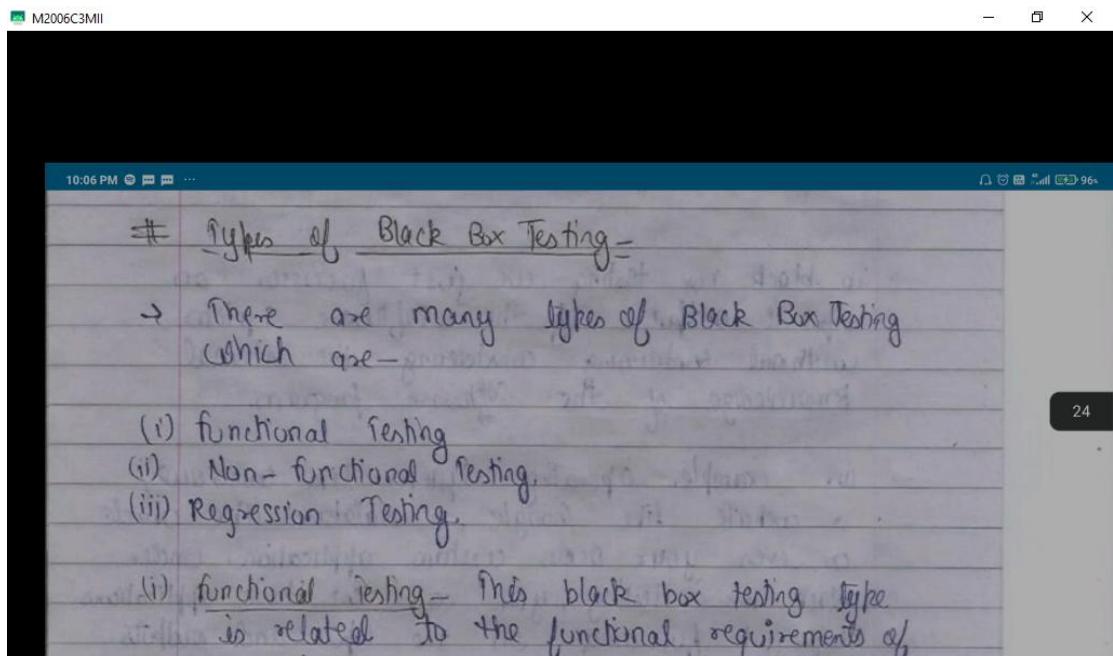


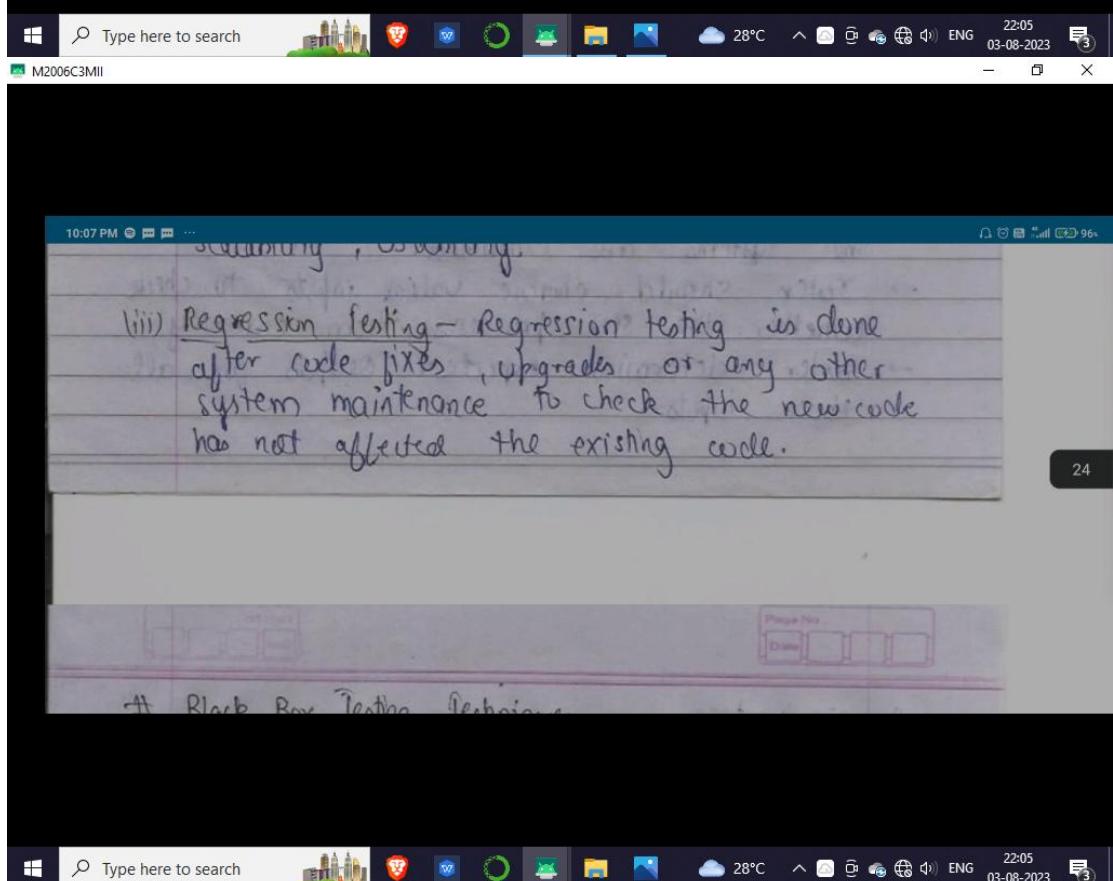
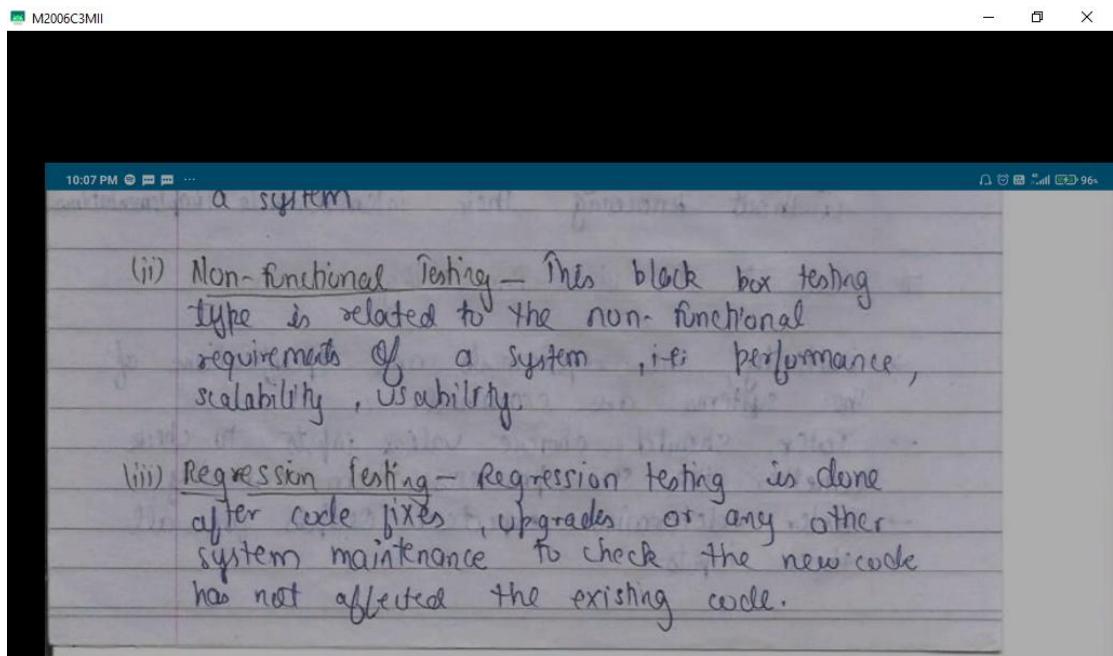


23



24





M2006C3MII

10:07 PM 22:05 96%

Black Box Testing Techniques

(i) Equivalence class Testing - It is used to minimize the no. of possible test cases to an optimal value level while maintains reasonable test coverage

(ii) Boundary value Testing - Boundary value is testing is focused on the values of boundaries. This technique determines whether a certain range of values are acceptable by the system or not.

25

M2006C3MII

10:07 PM 22:05 96%

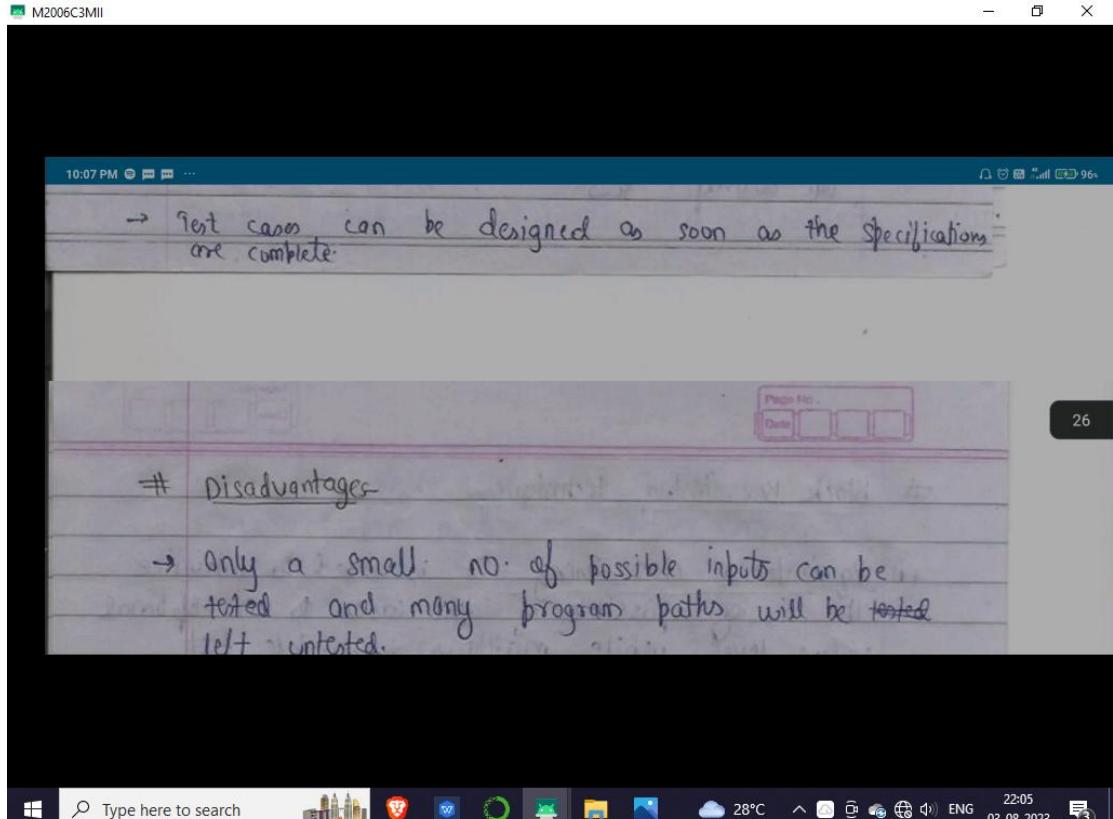
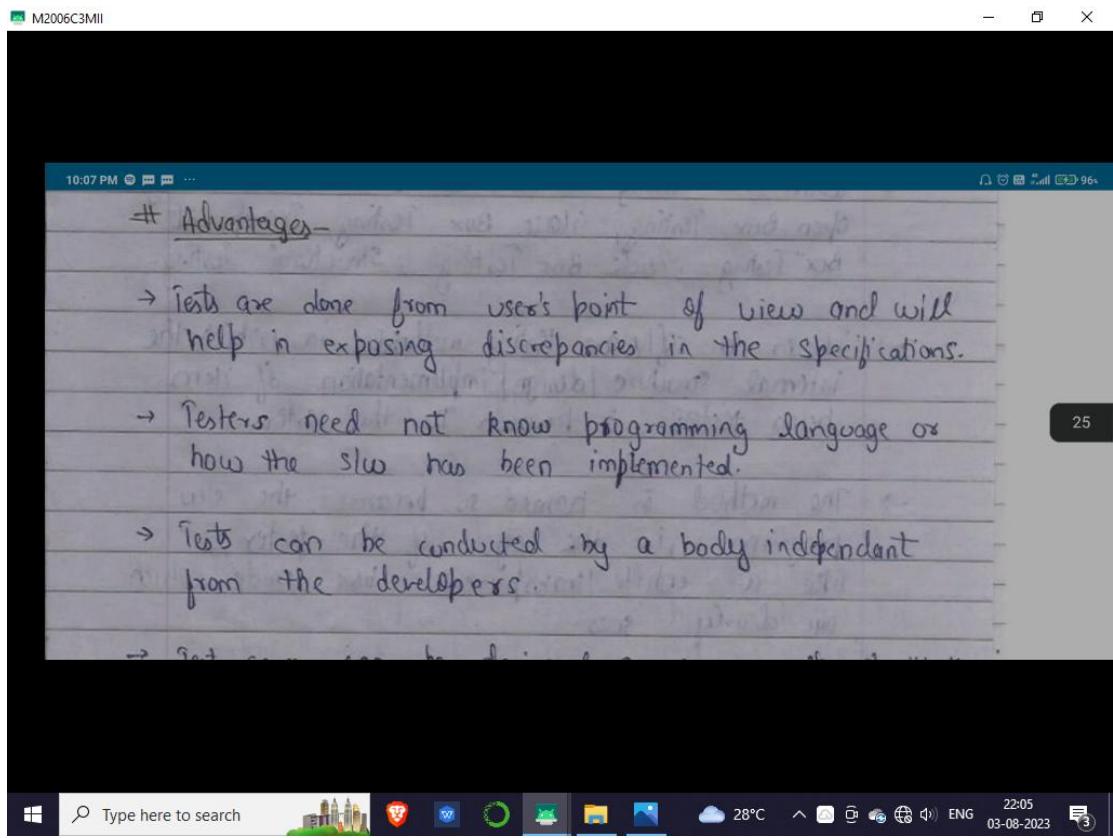
(ii) Boundary value Testing - Boundary value is testing is focused on the values of boundaries. This technique determines whether a certain range of values are acceptable by the system or not.

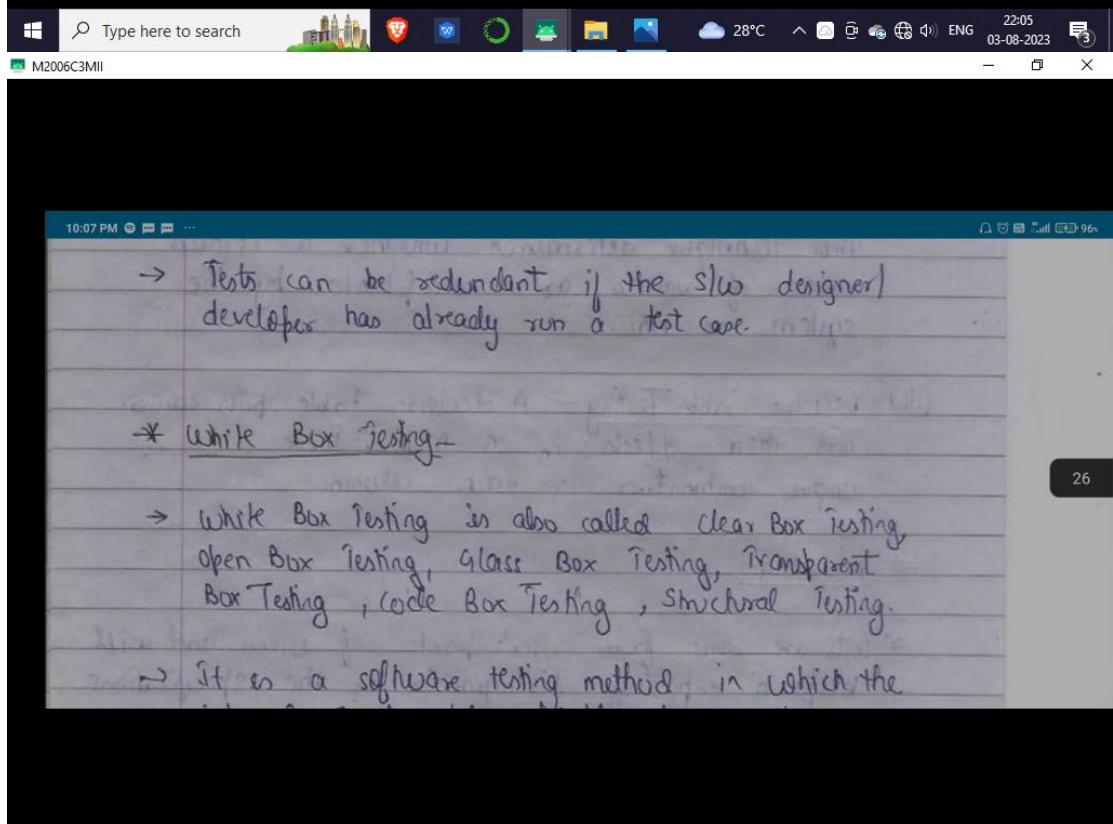
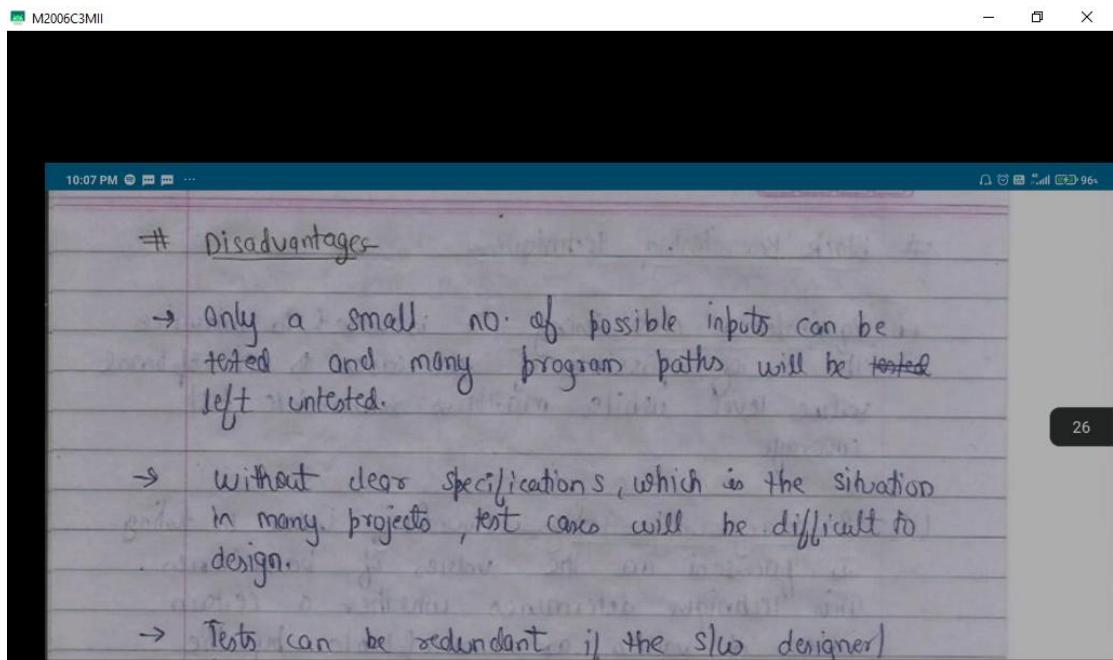
(iii) Decision Table Testing - A decision table puts causes and their effects in a matrix. There is a unique combination in each column.

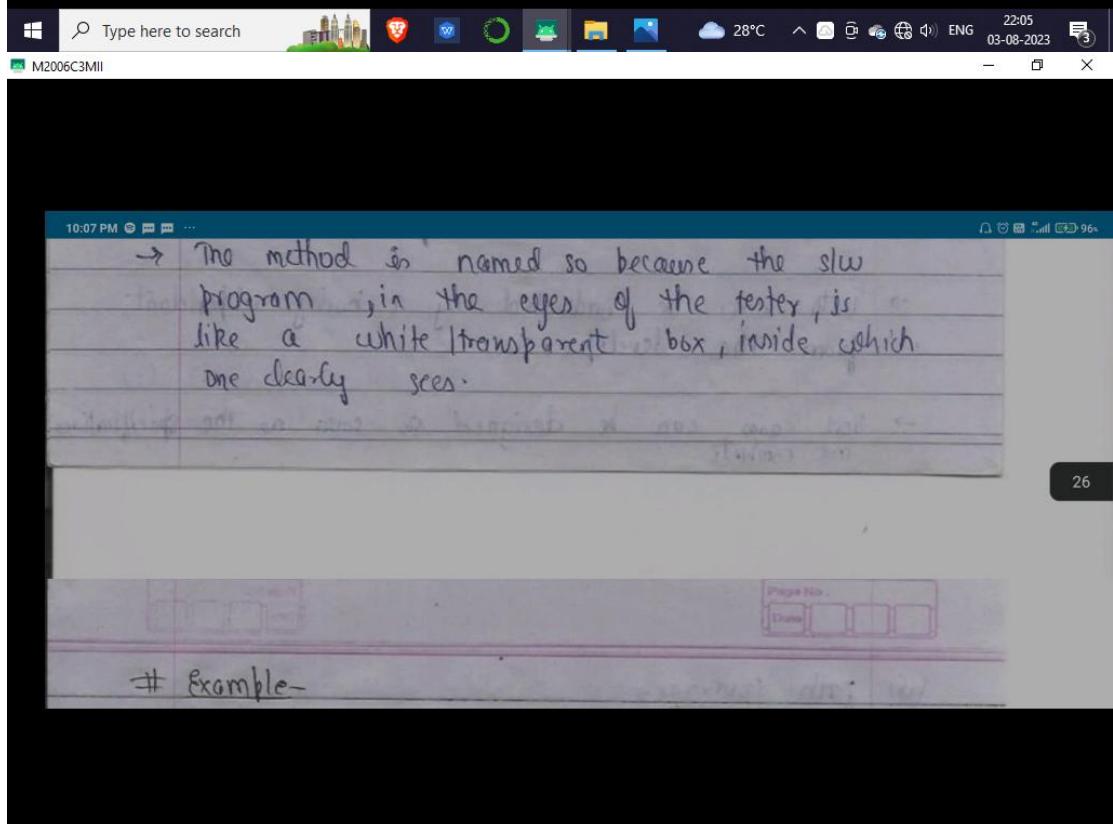
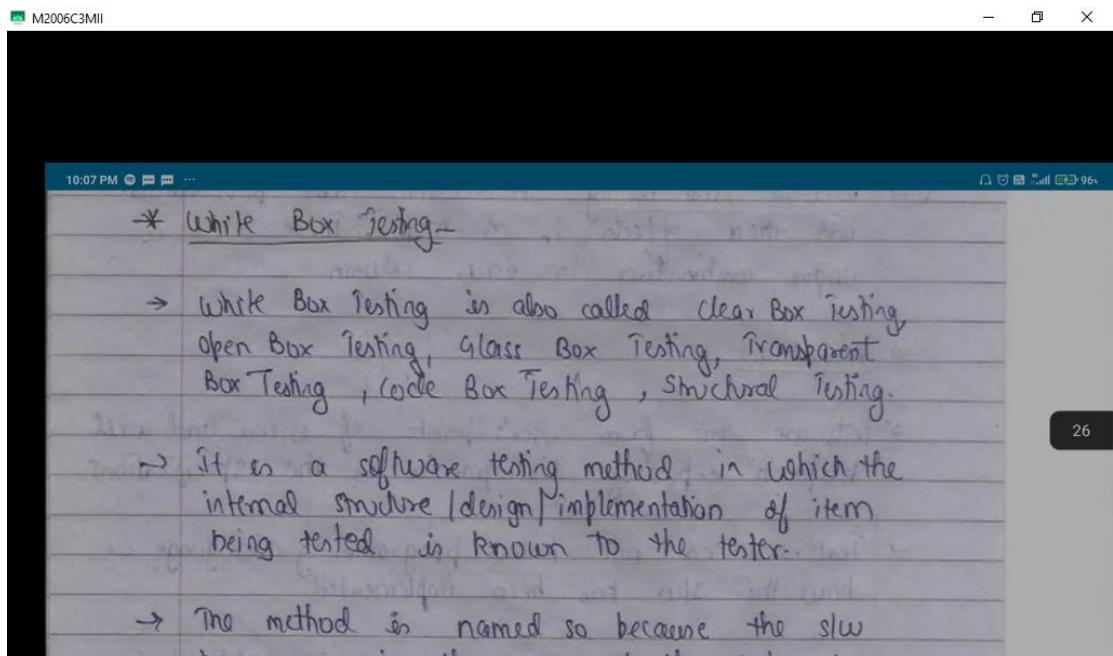
Advantages -

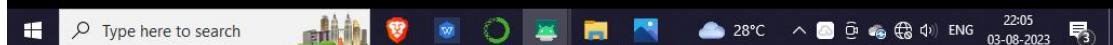
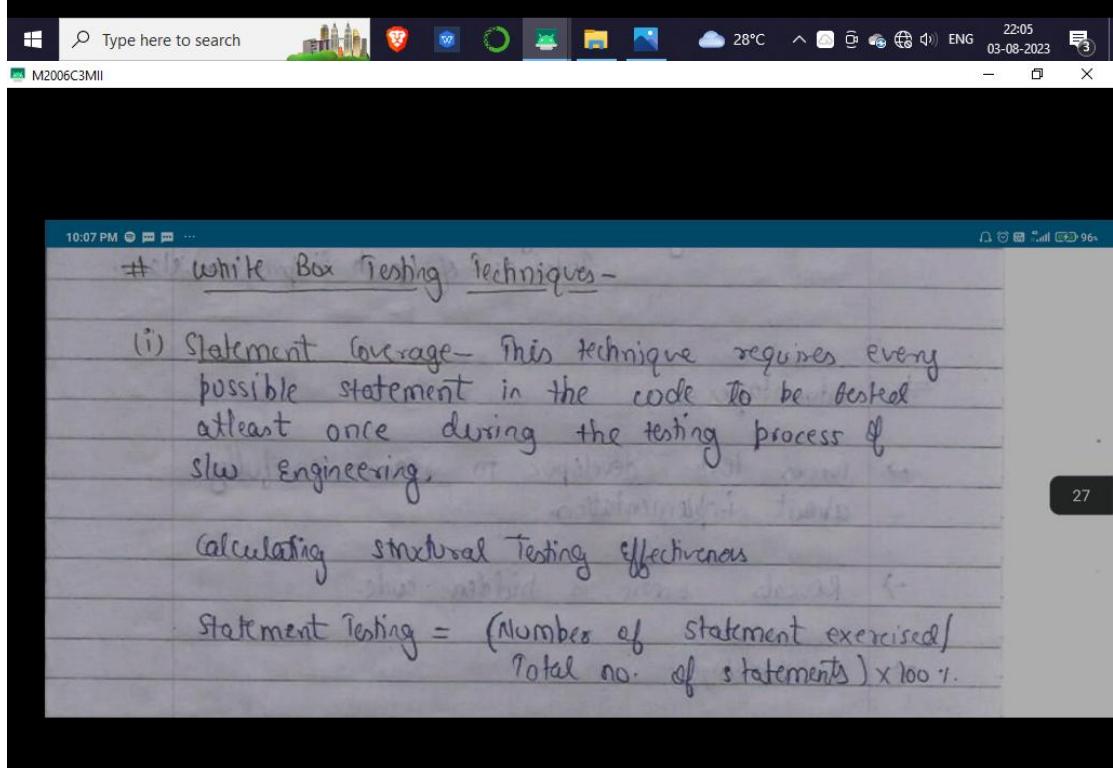
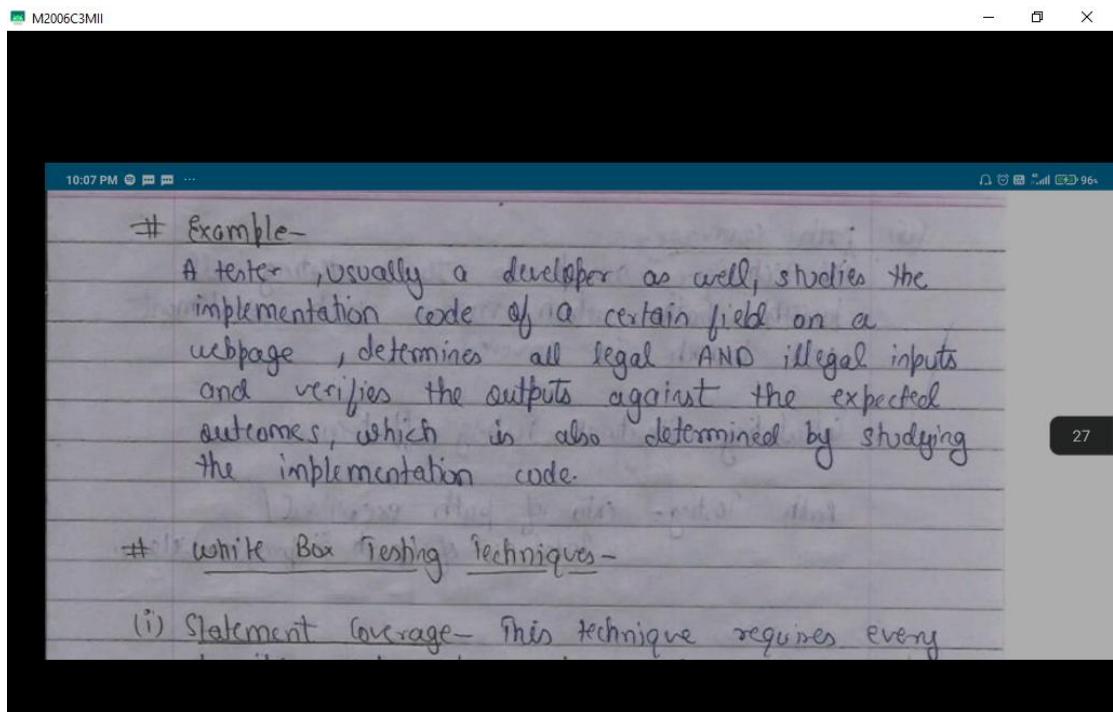
M2006C3MII

10:07 PM 22:05 96%









10:07 PM

96%

(ii) Branch Coverage -

This technique checks every possible path (if-else) and other conditional loop of a SW application.

Calculating structural testing effectiveness.

$$\text{Branch Testing} = \frac{\text{No. of decisions outcome testing}}{\text{Total no. of outcomes}} \times 100\%$$

27

10:08 PM

22:05

(iii) Path Coverage -

This technique corresponds to a testing all possible paths which means each statement and branch is covered.

Calculating Structural testing effectiveness

$$\text{Path Testing} = \frac{\text{No. of path exercised}}{\text{Total no. of path in program}} \times 100\%$$

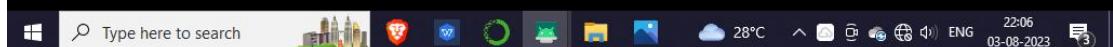
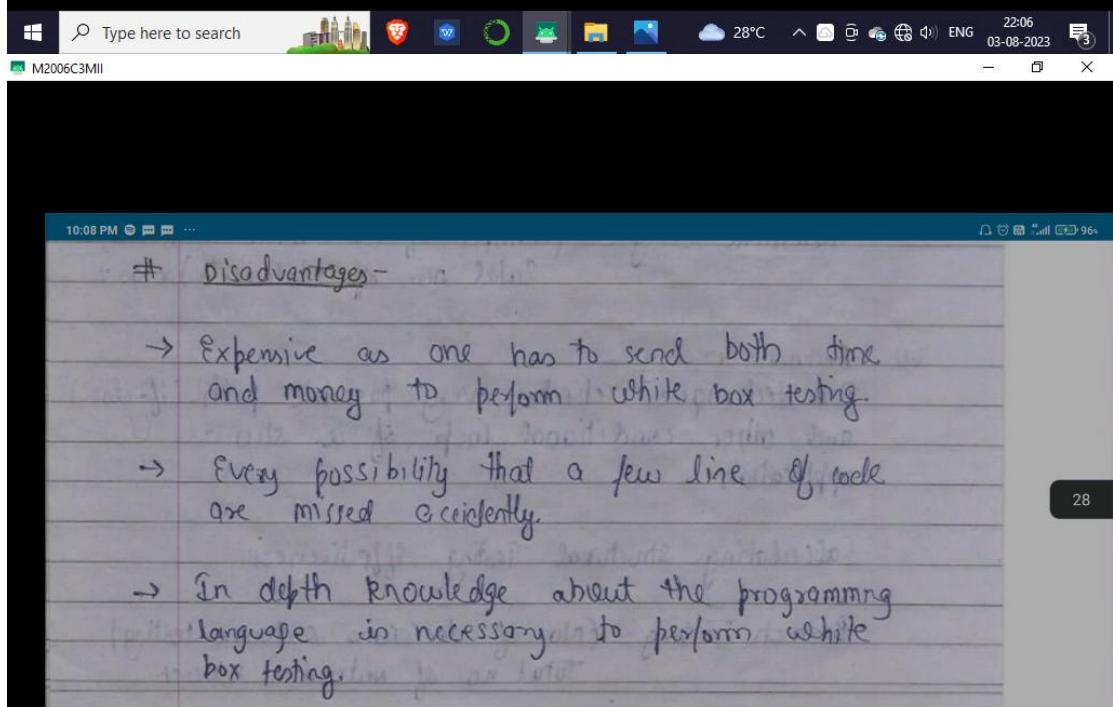
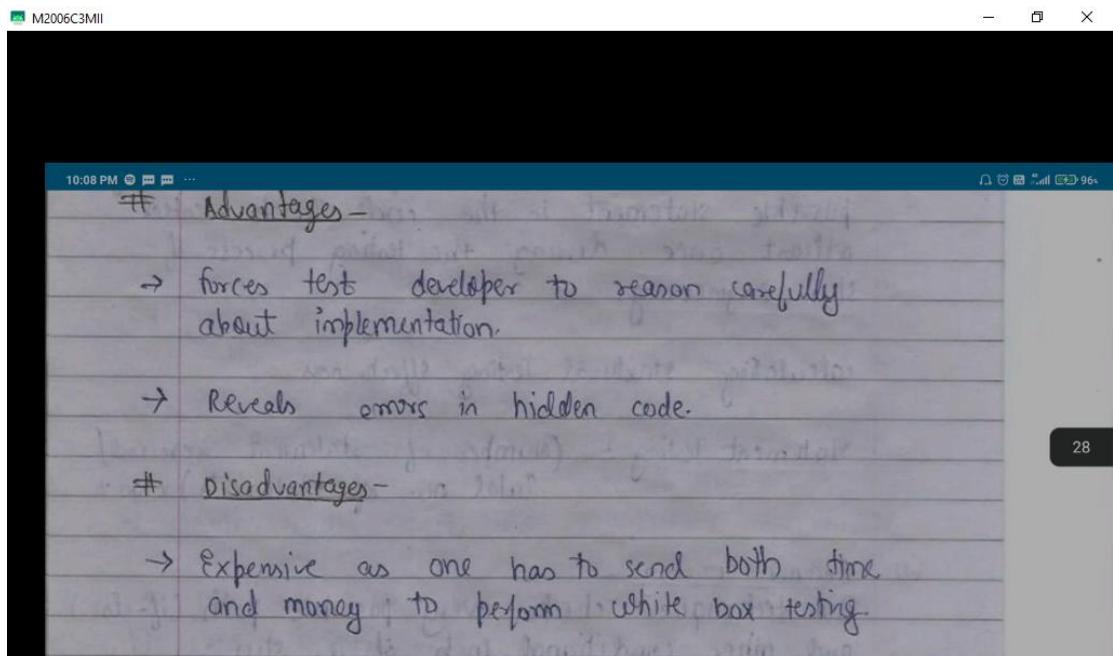
28

Type here to search



22:06

03-08-2023



* difference b/w white Box testing & Black Box Testing -

Black Box testing	white Bbx testing
<p>(i) It is a testing approach which is used to test the software without the knowledge of the internal structure of program or application.</p>	<p>(ii) It is a testing approach in which internal structure is known to the tester.</p>

ii) It is also known as data-driven, B-functional testing.

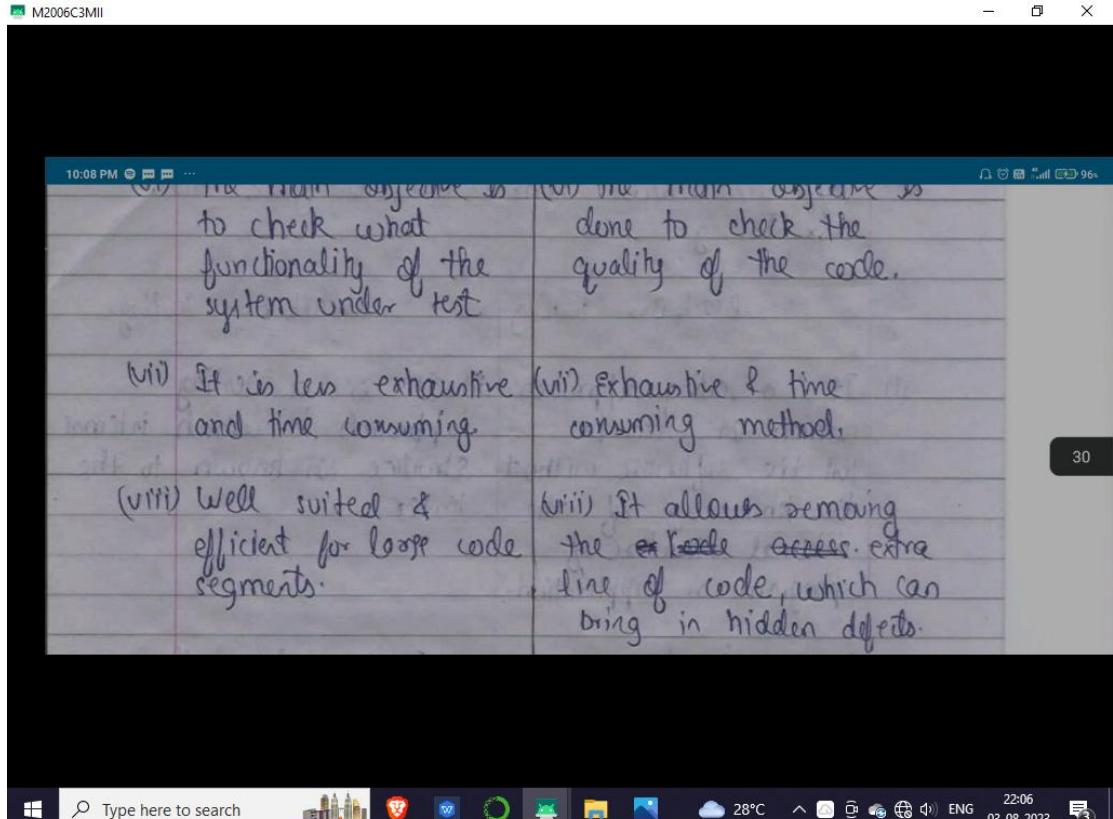
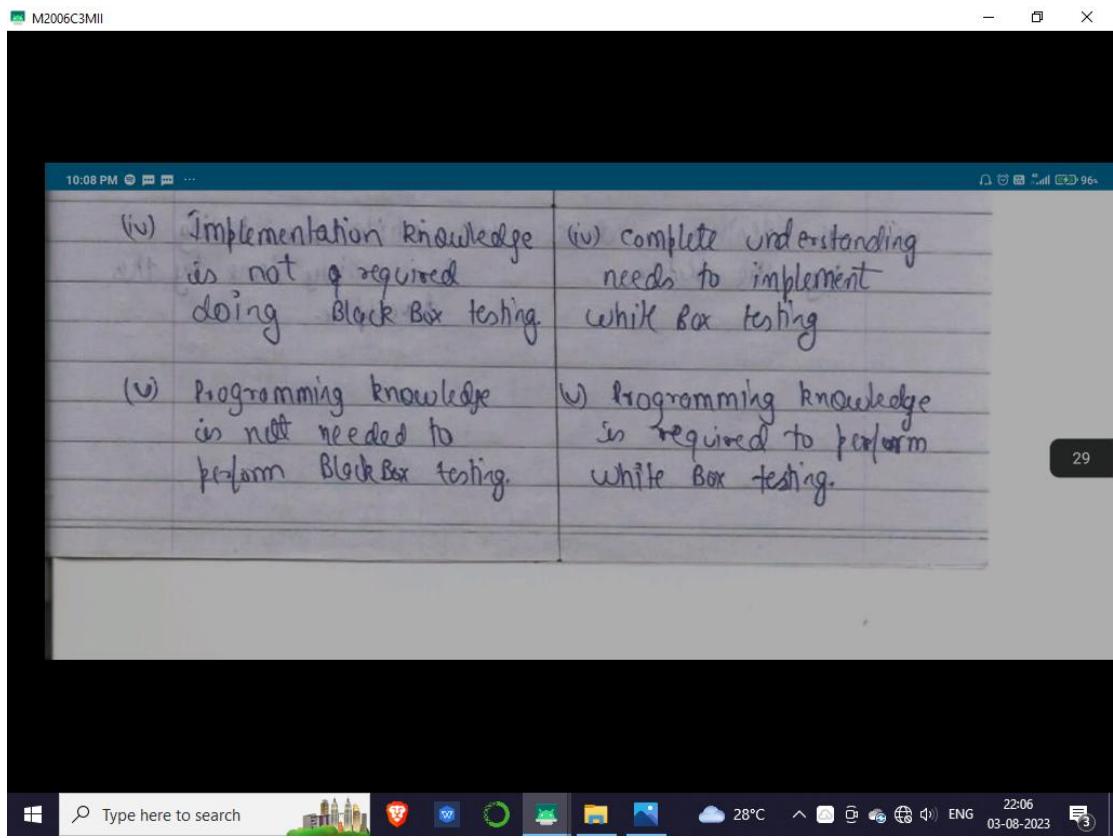
iii) testing is based on external expectations, internal behaviour of the application is unknown.

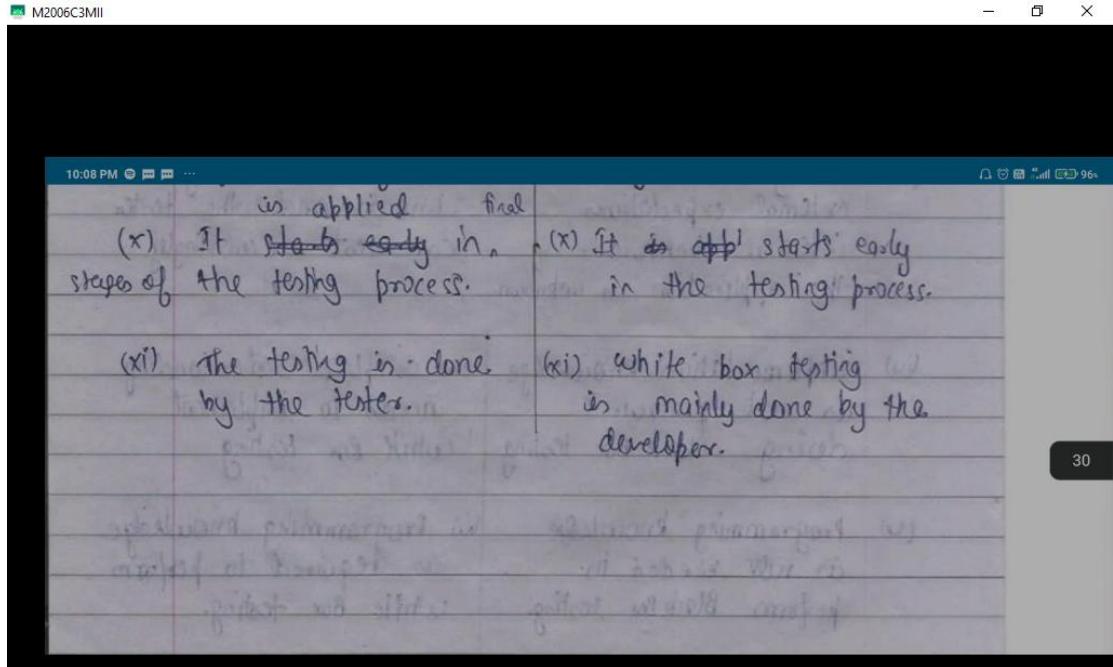
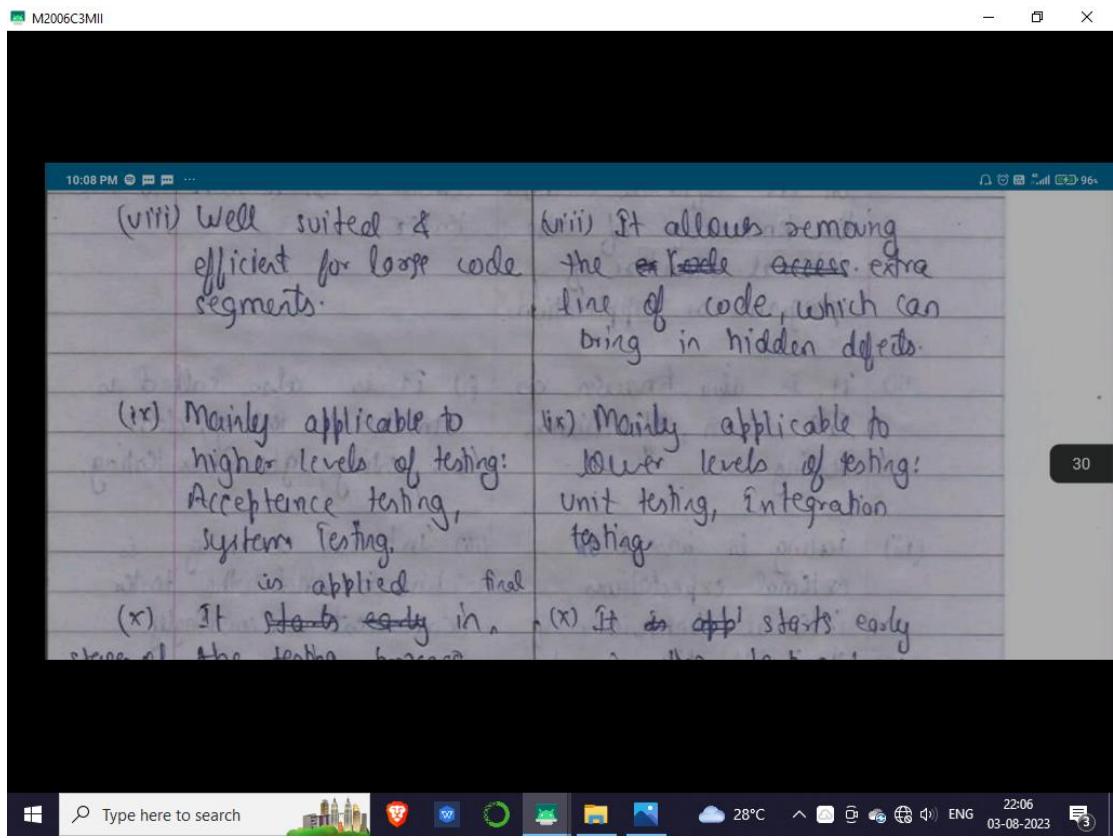
iv) Implementation knowledge is not required

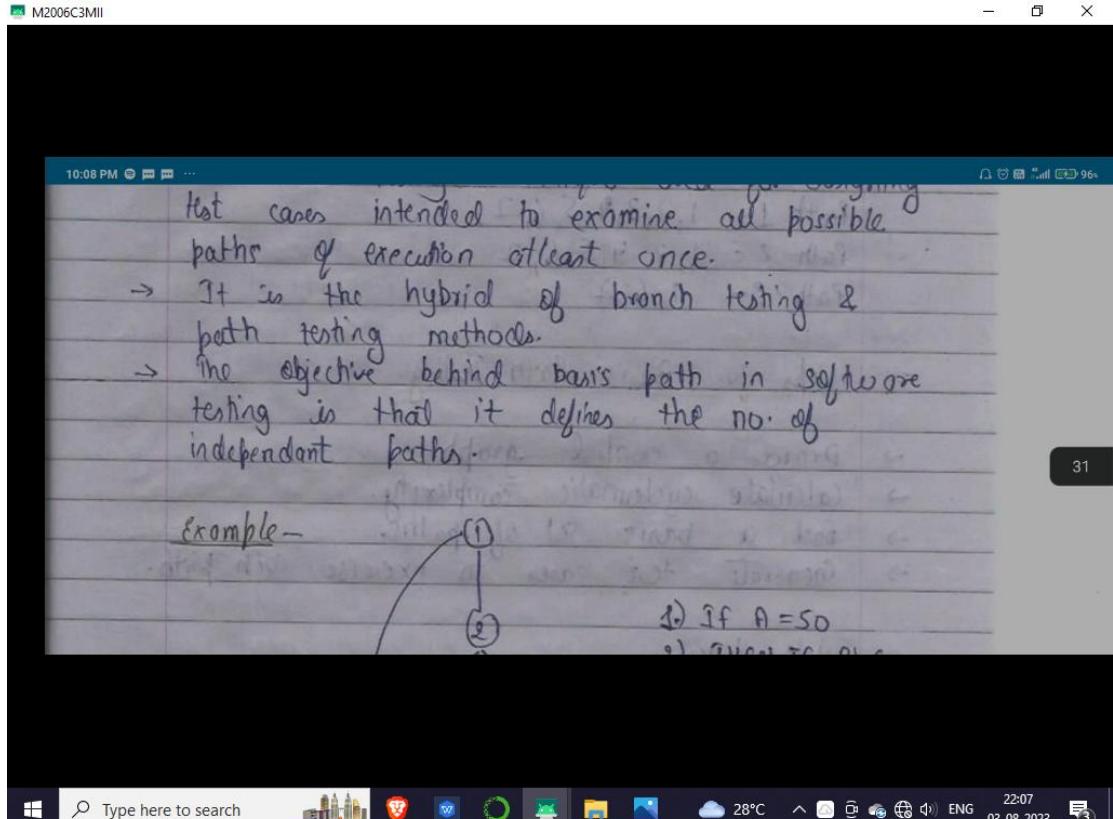
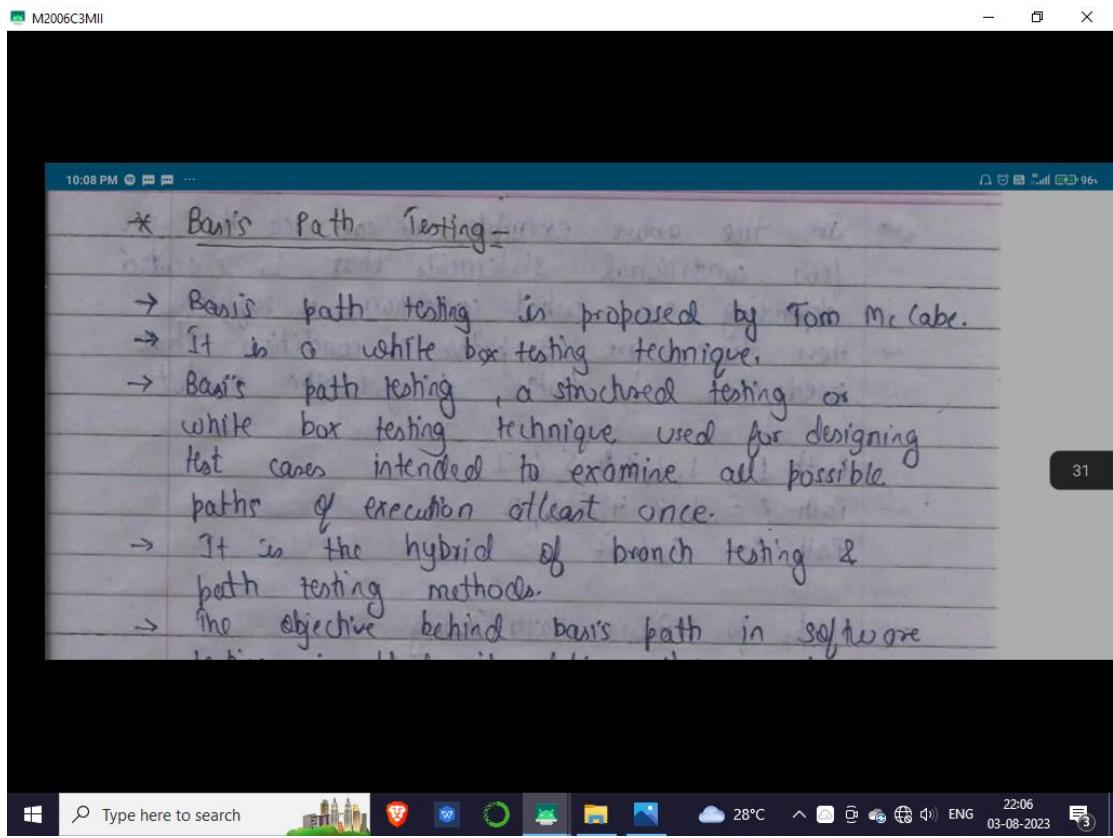
ii) It is also called as structural testing, clear box testing, glass box testing.

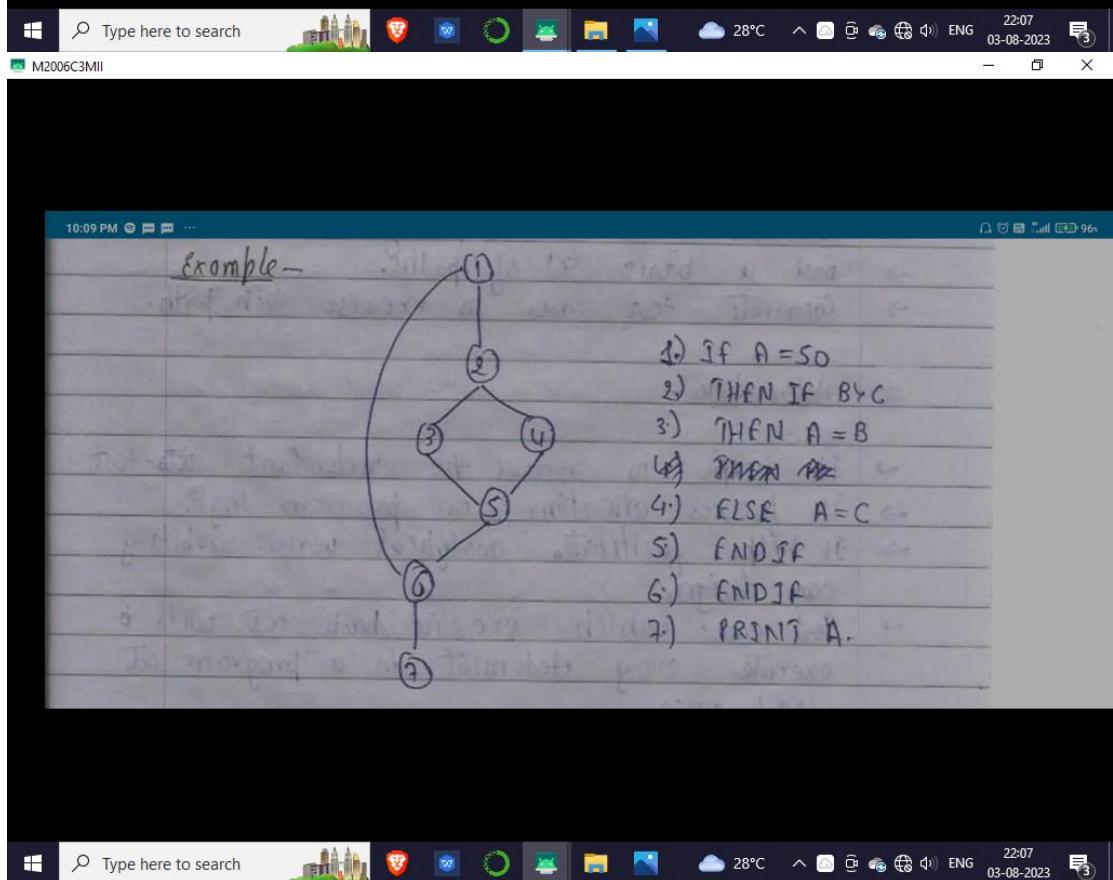
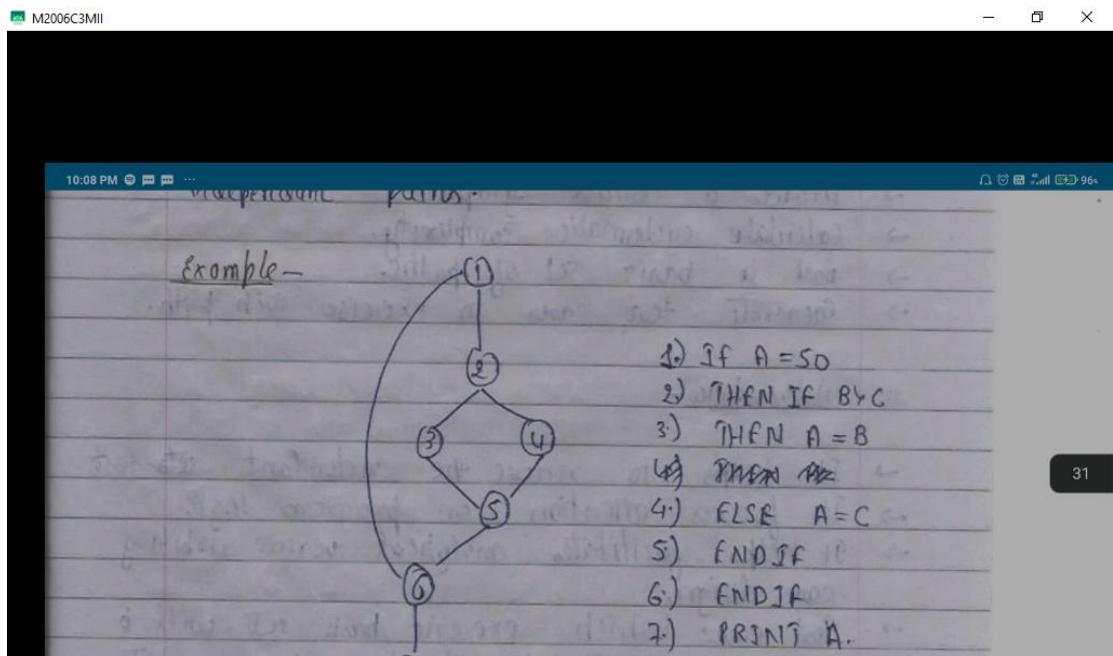
iii) Internal working is known and the tester can test accordingly.

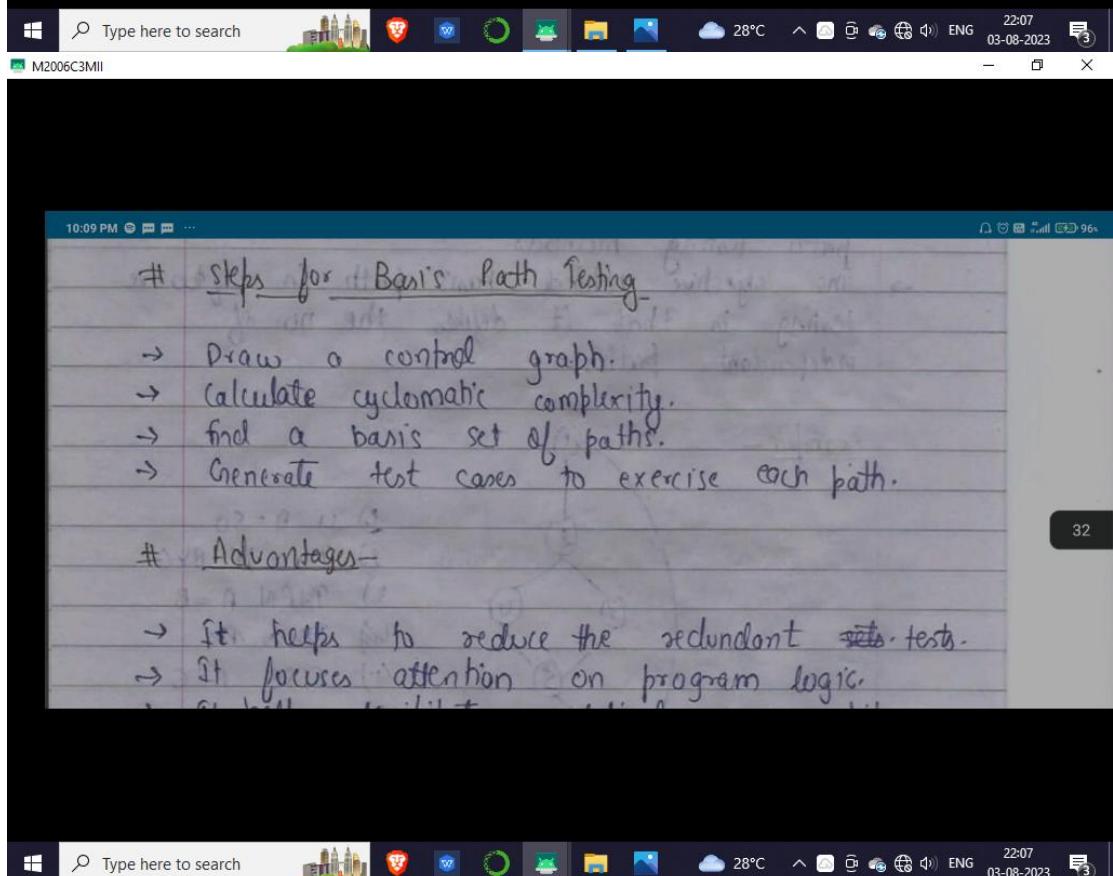
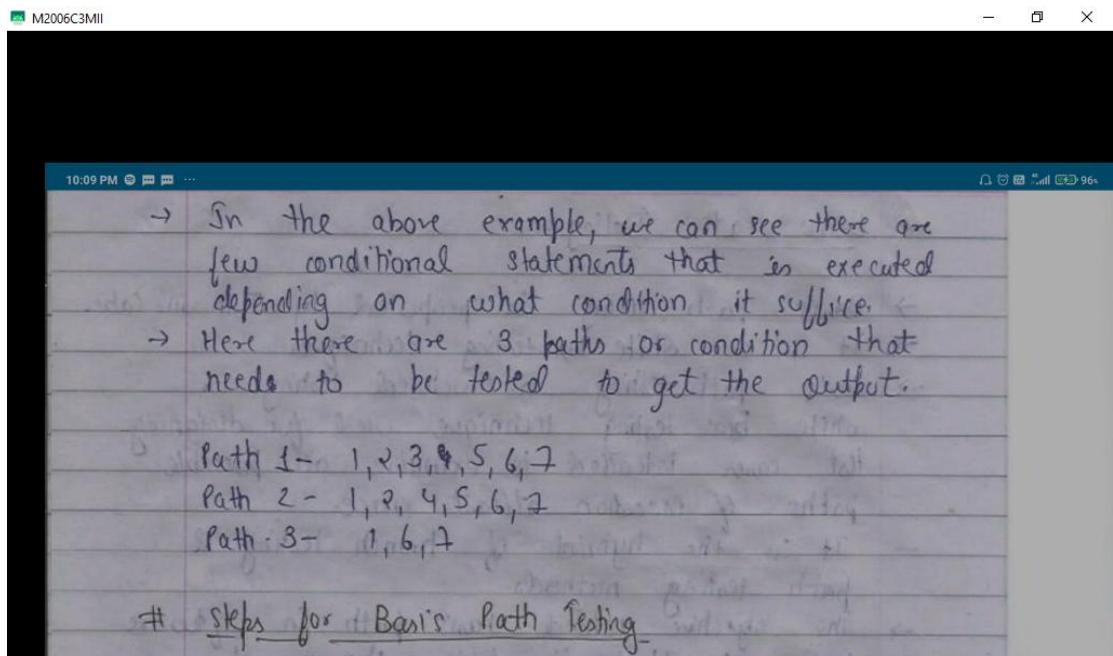
iv) complete understanding needs to implement

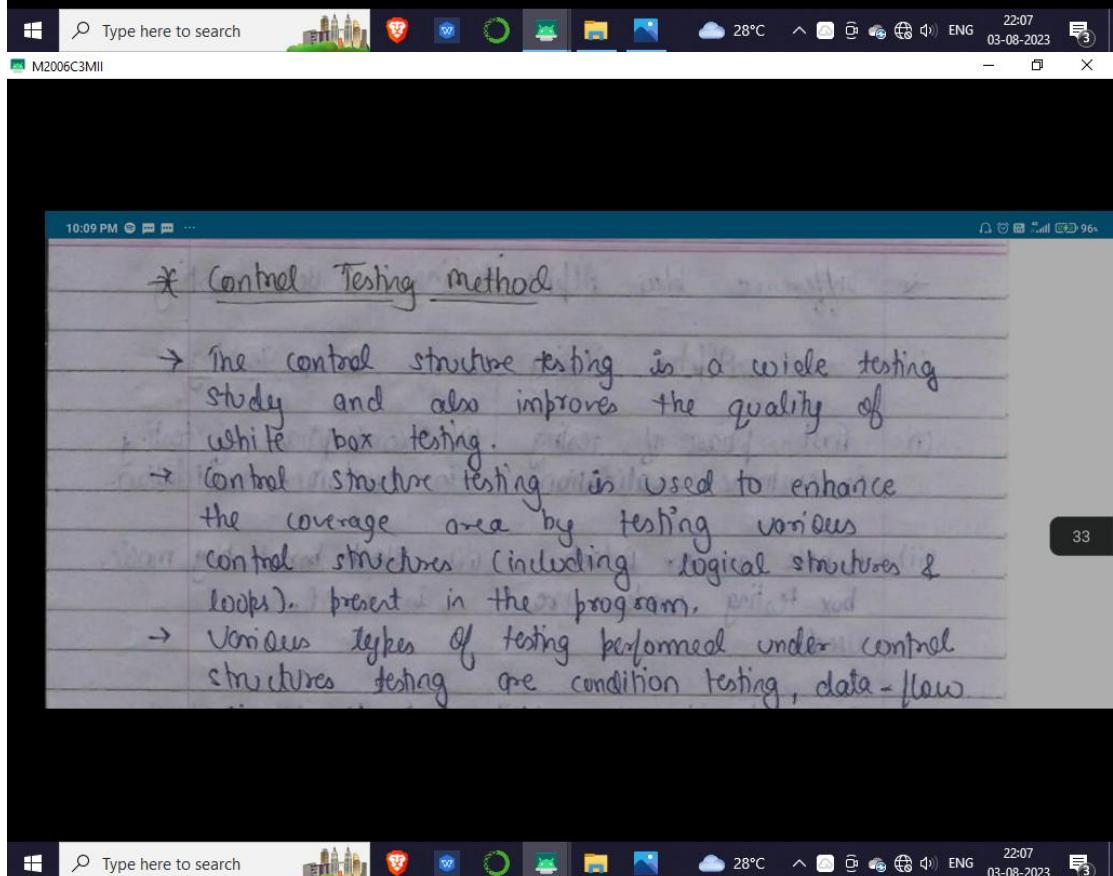
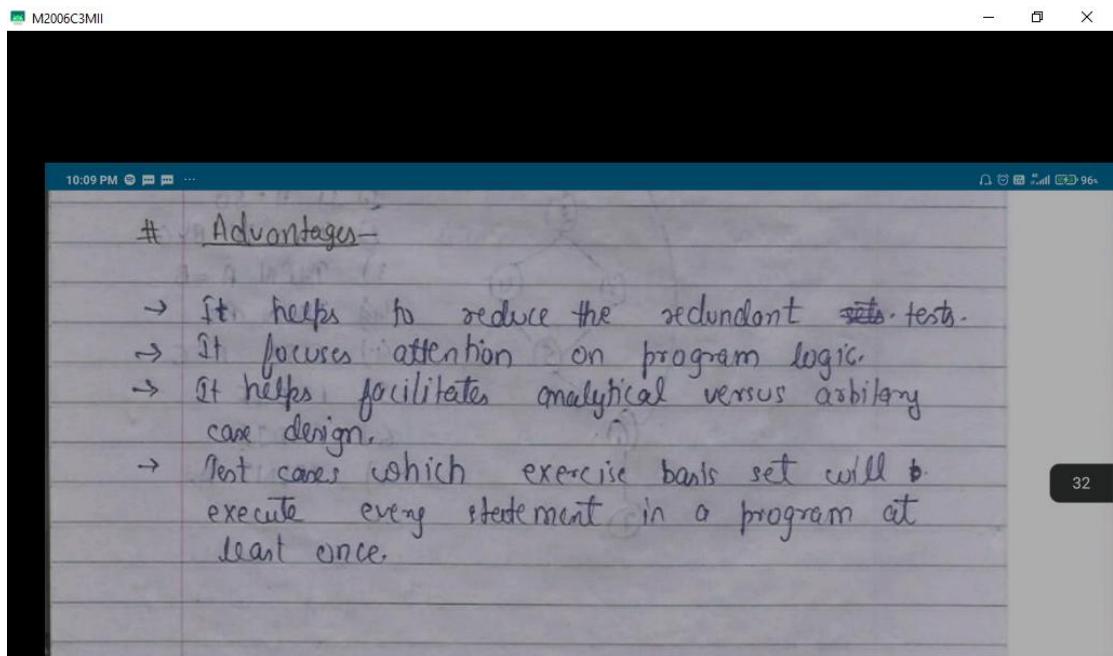


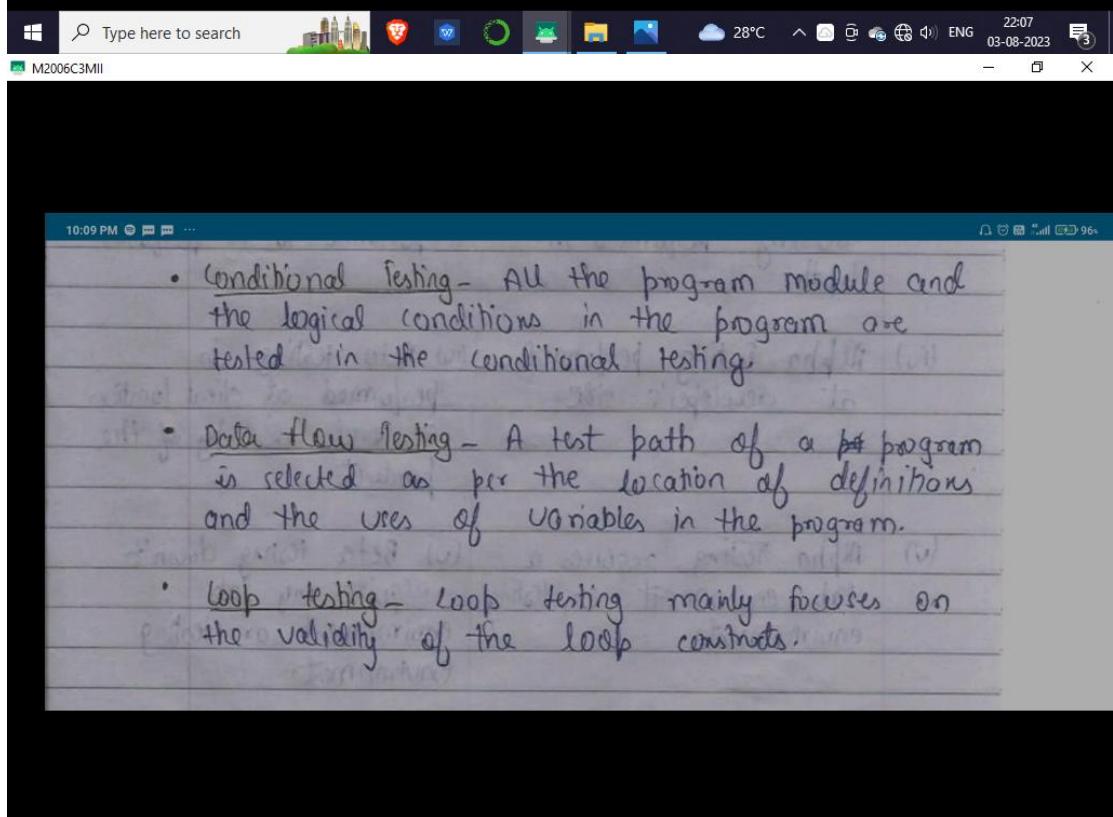
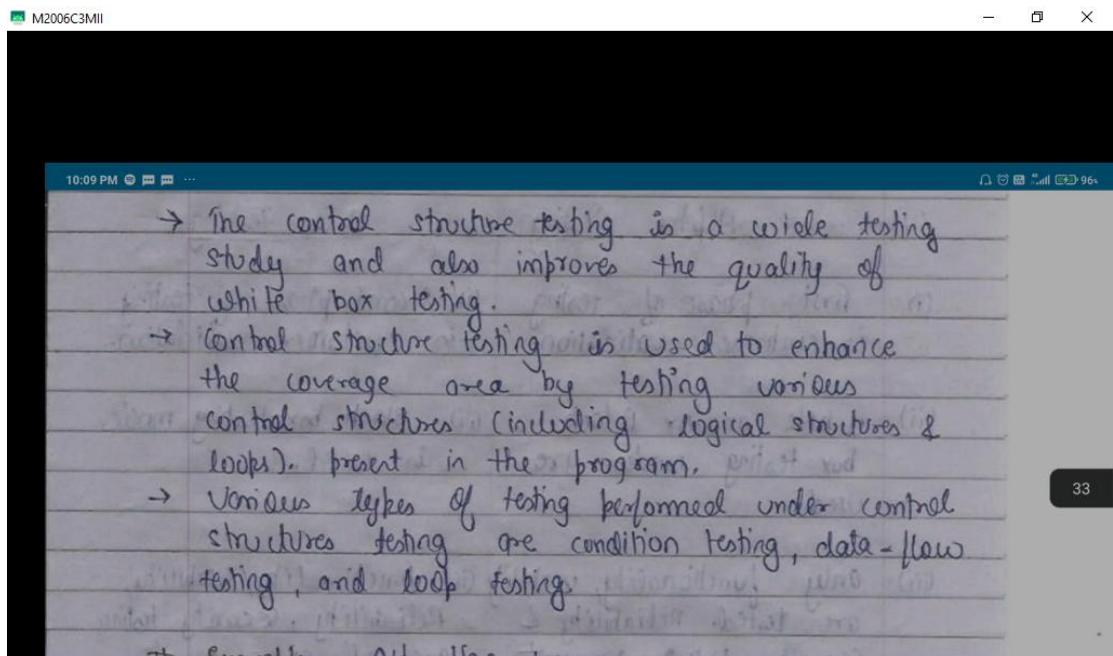












* Difference b/w Alpha testing & Beta testing

<u>Alpha Testing</u>	<u>Beta Testing</u>
(i) first phase of testing in customer validation.	(ii) second phase of testing in customer validation.
(ii) white and/or black box testing modes are used.	(ii) black box testing mode is used.

(ii) white and/or black box testing modes are used.

(iii) only functionality, usability, Reliability & security testing are not usually performed in-depth.

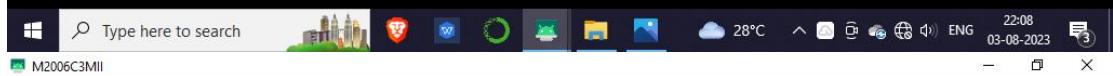
(iv) Alpha testing performed

(ii) black box testing mode is used.

(iii) functionality, Usability, Reliability, security testing are all given equal importance to be performed.

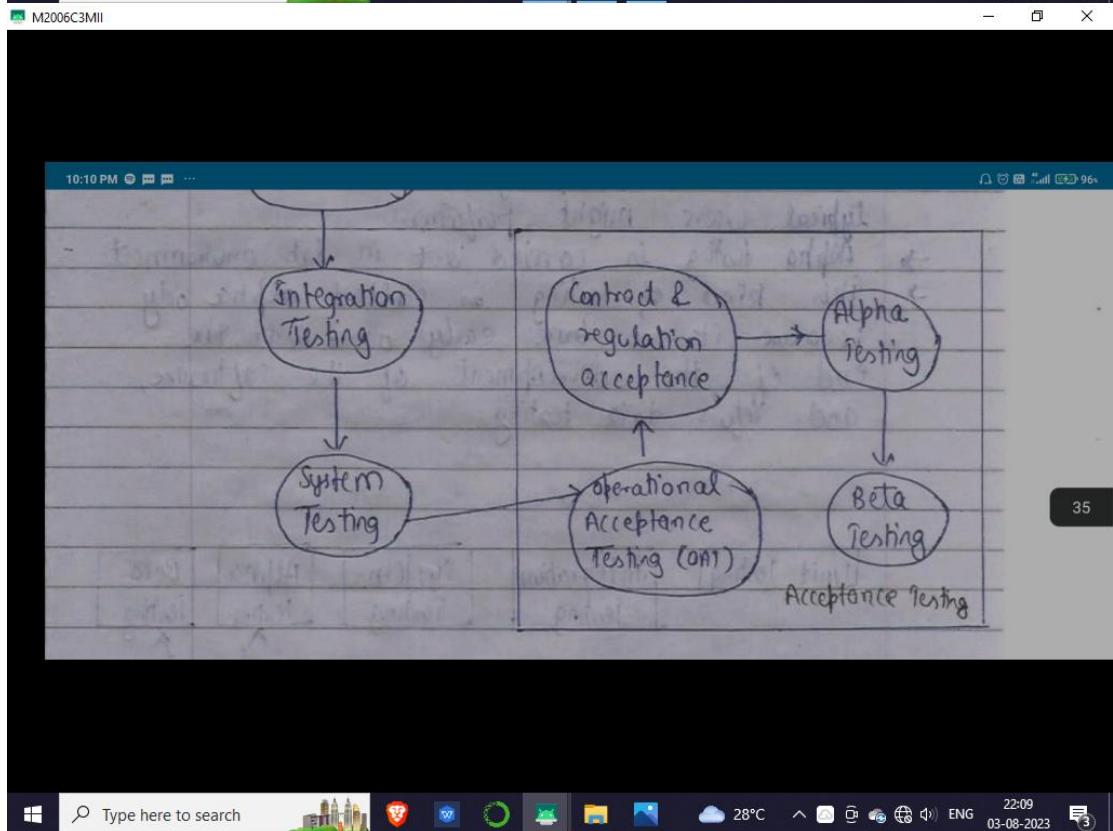
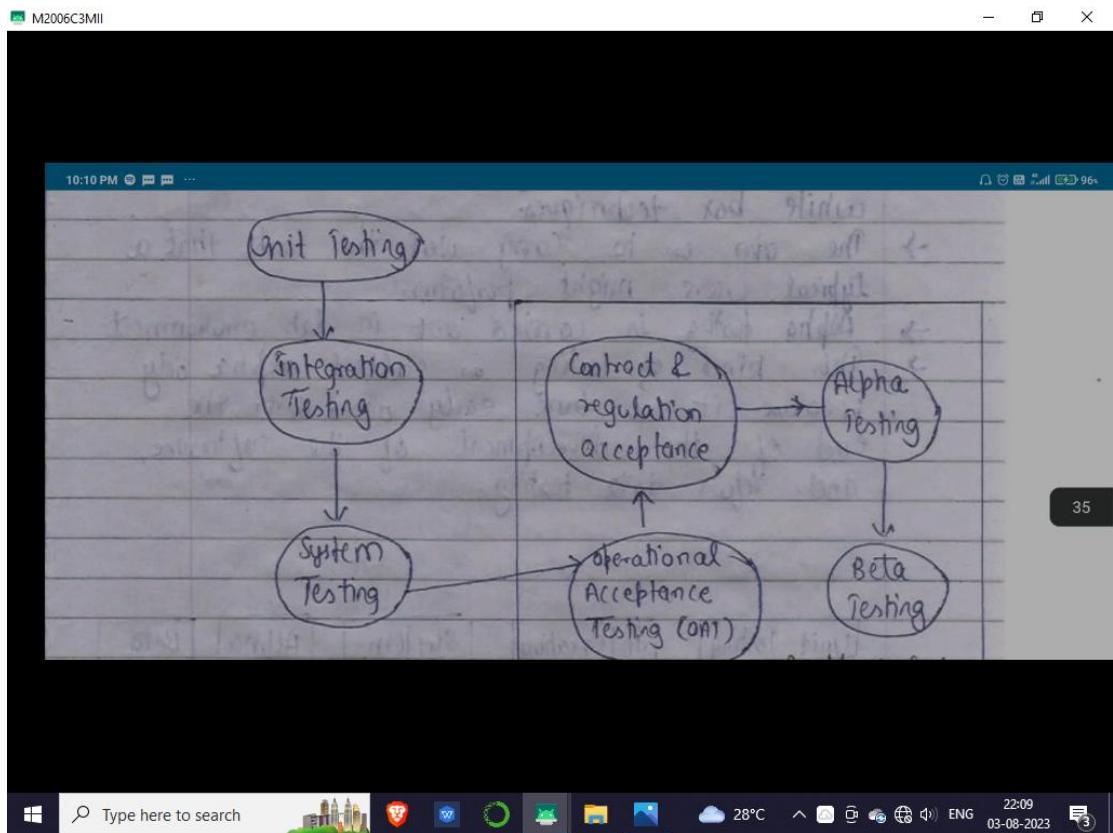
(iv) Beta testing is

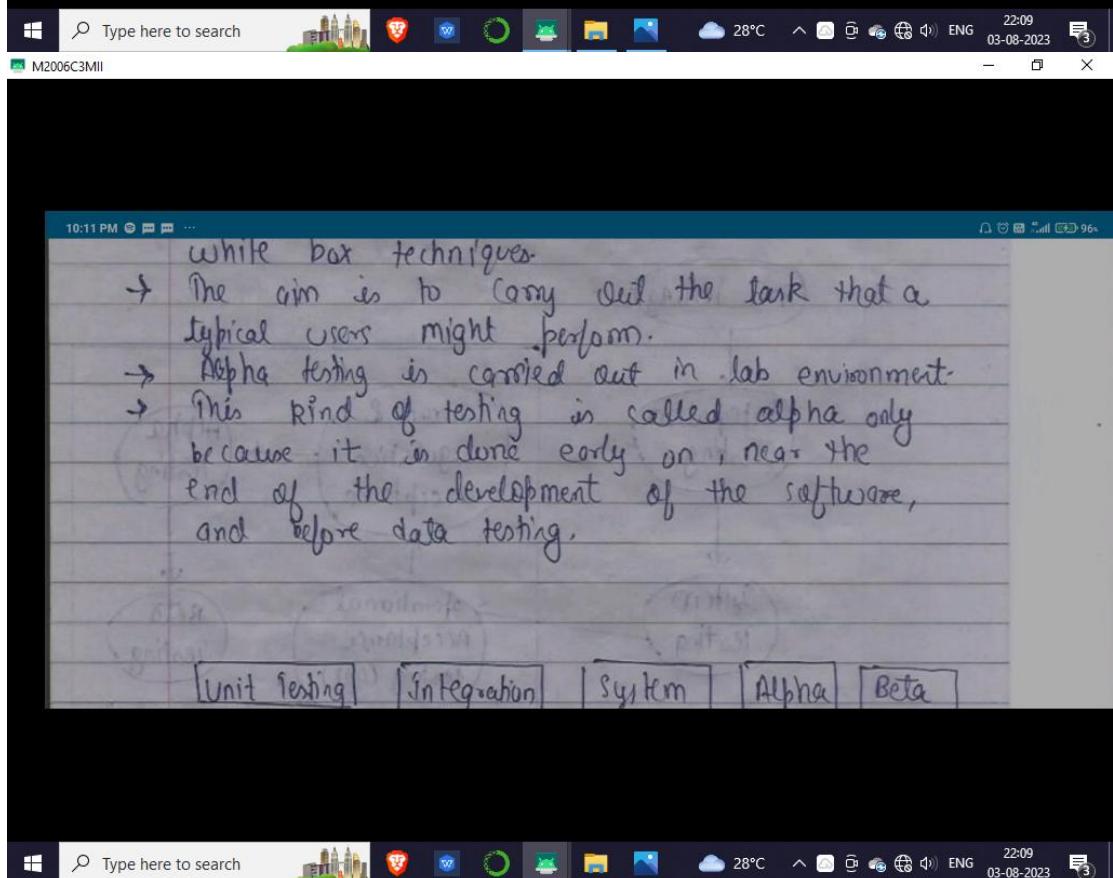
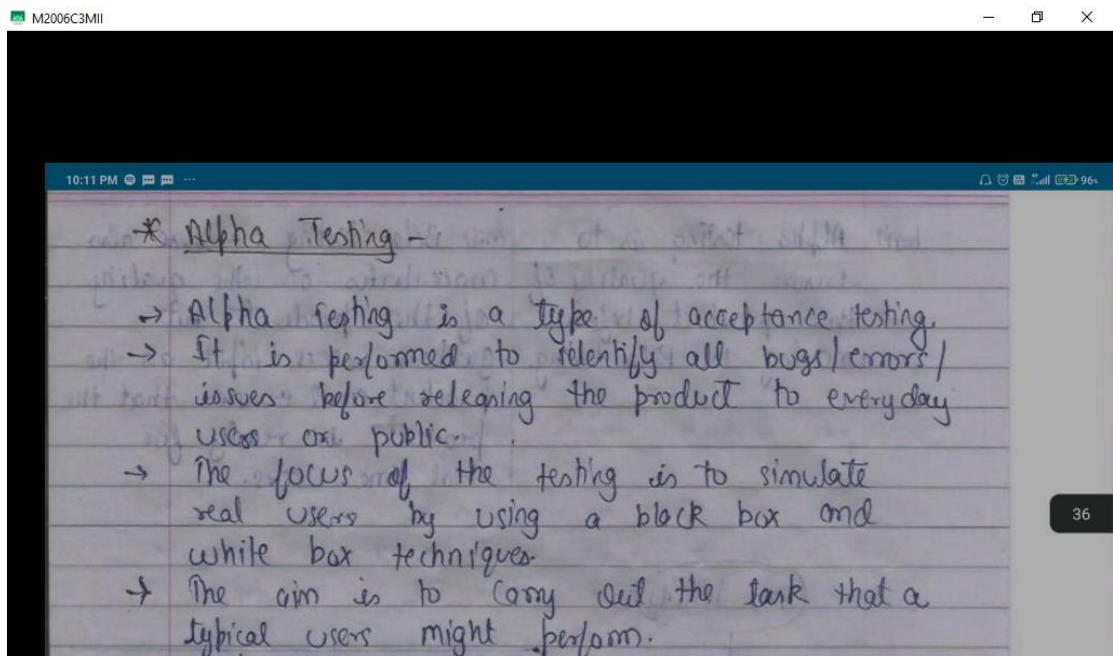
at developer's site.	performed at client location or end user by of the product.
(v) Alpha Testing requires a lab environment or testing environment.	(v) Beta testing doesn't require any lab environment or testing environment.
(vi) long execution cycle may be required for Alpha testing.	(vi) only a few weeks of execution are required for Beta testing.

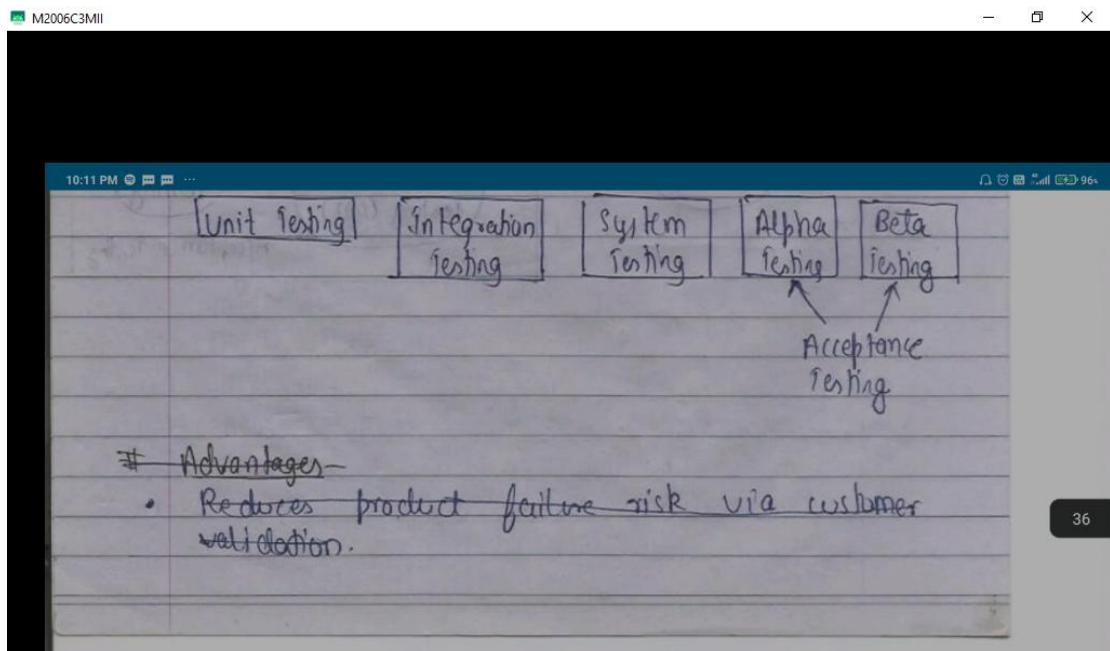


(vii) Alpha testing is to ensure the quality of the product before moving to Beta testing.	(viii) Beta testing ensures also concentrates on the quality of the product, but gathers user input on the product and ensures that the product is ready for real-time users.
Unit Testing	

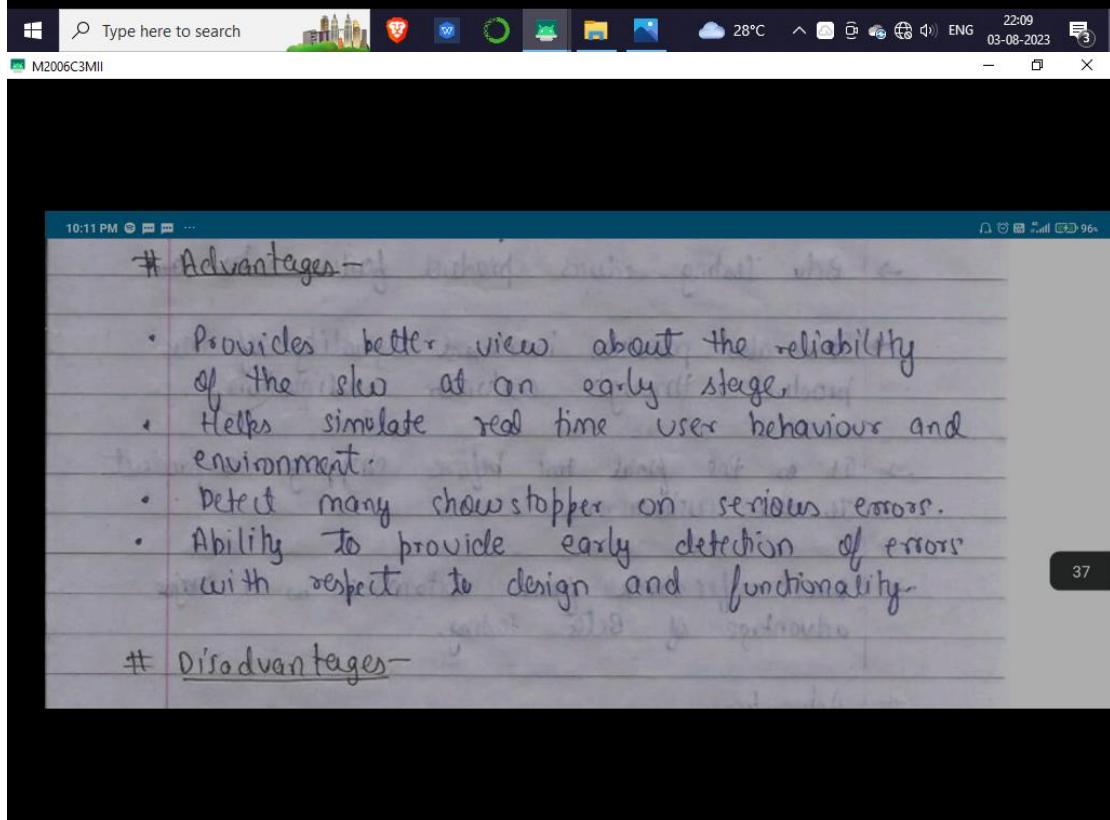




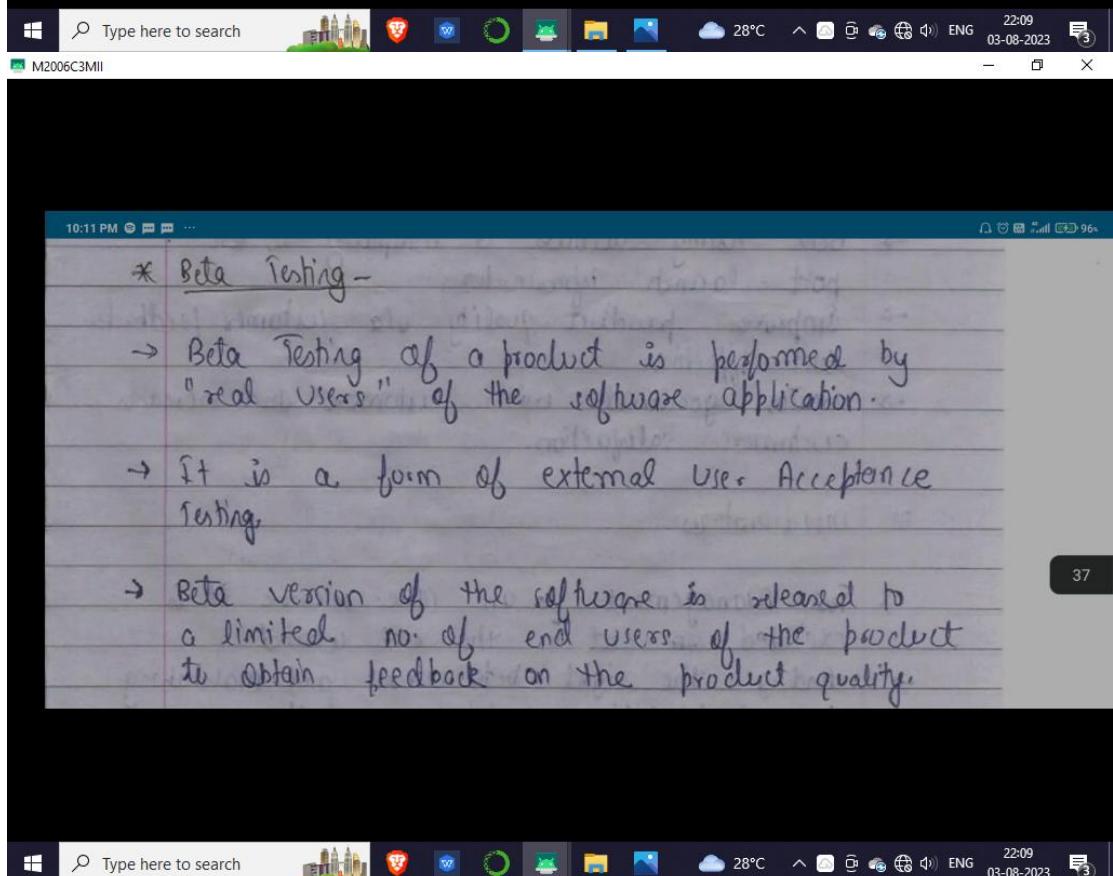
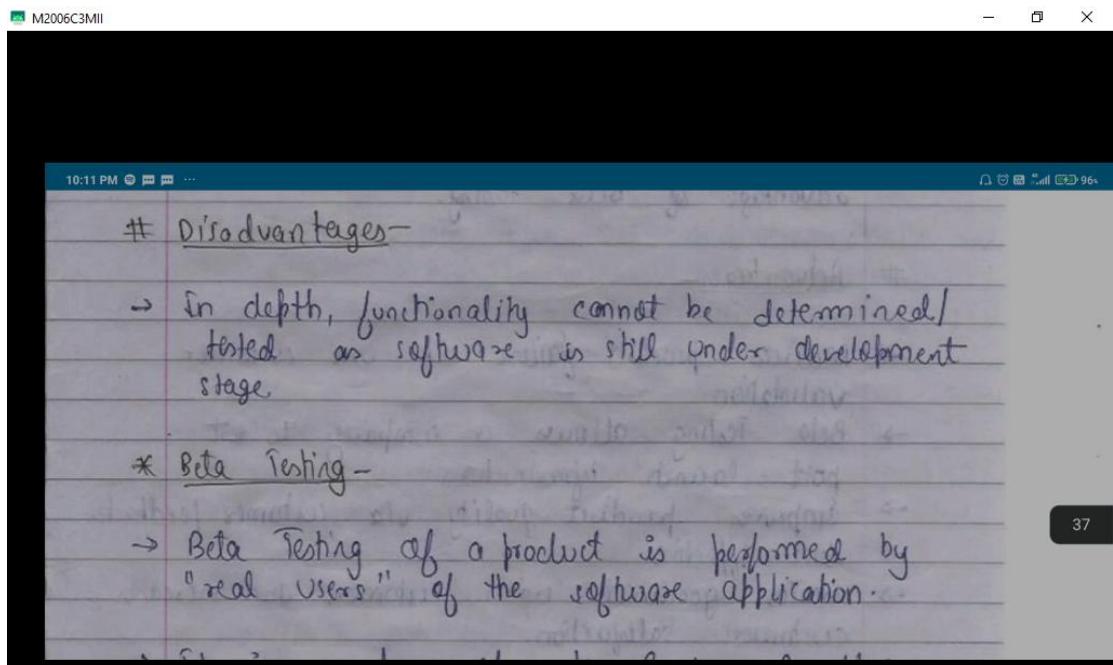


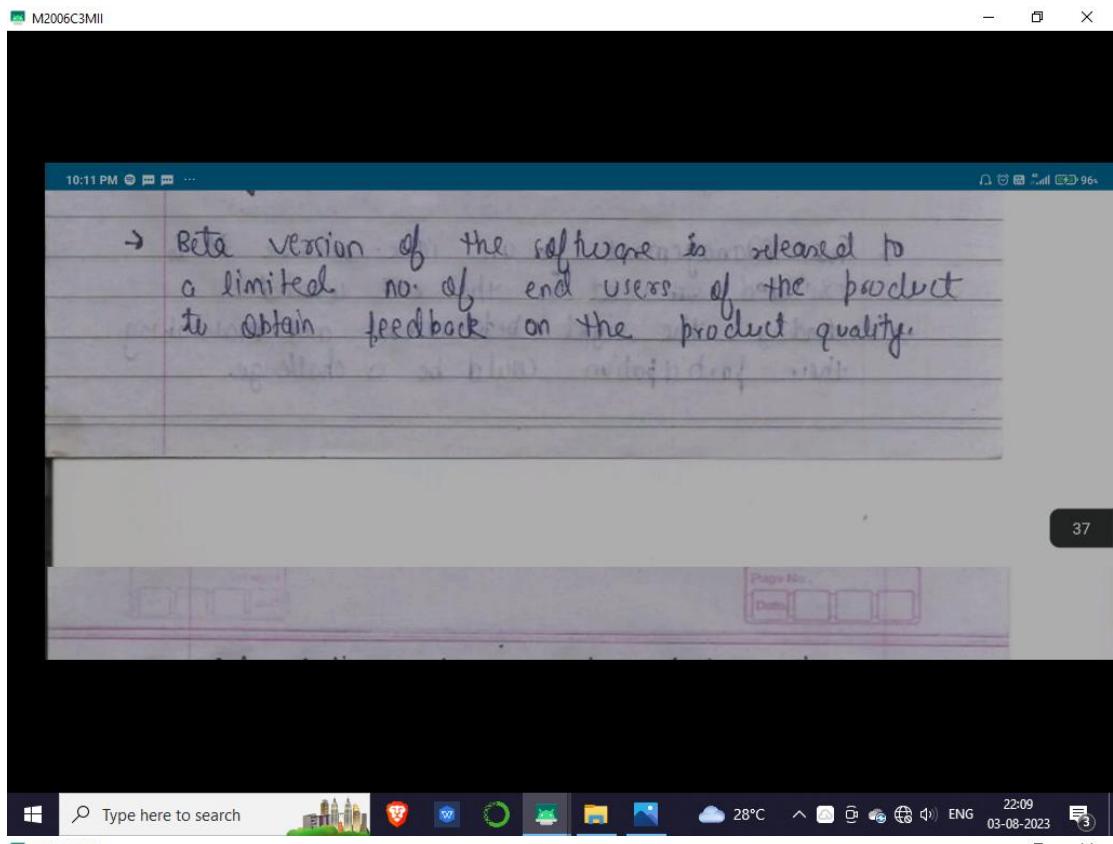


36

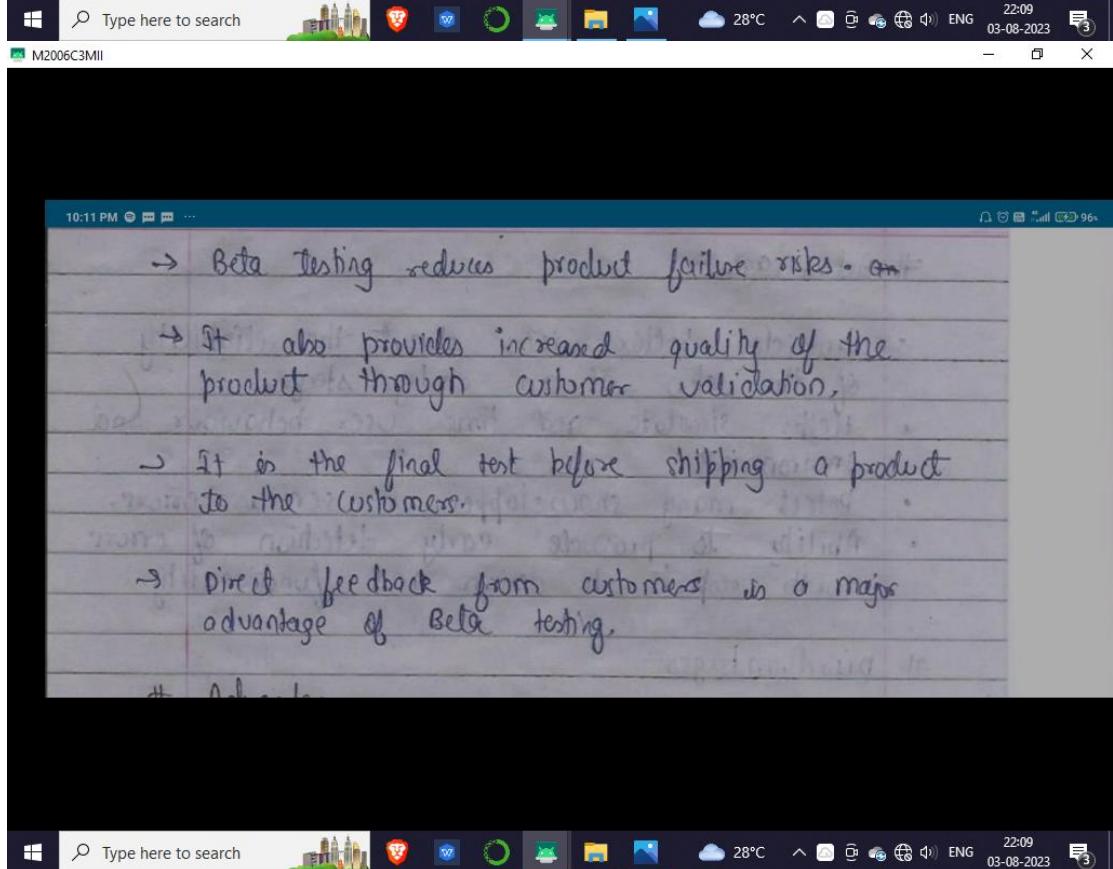


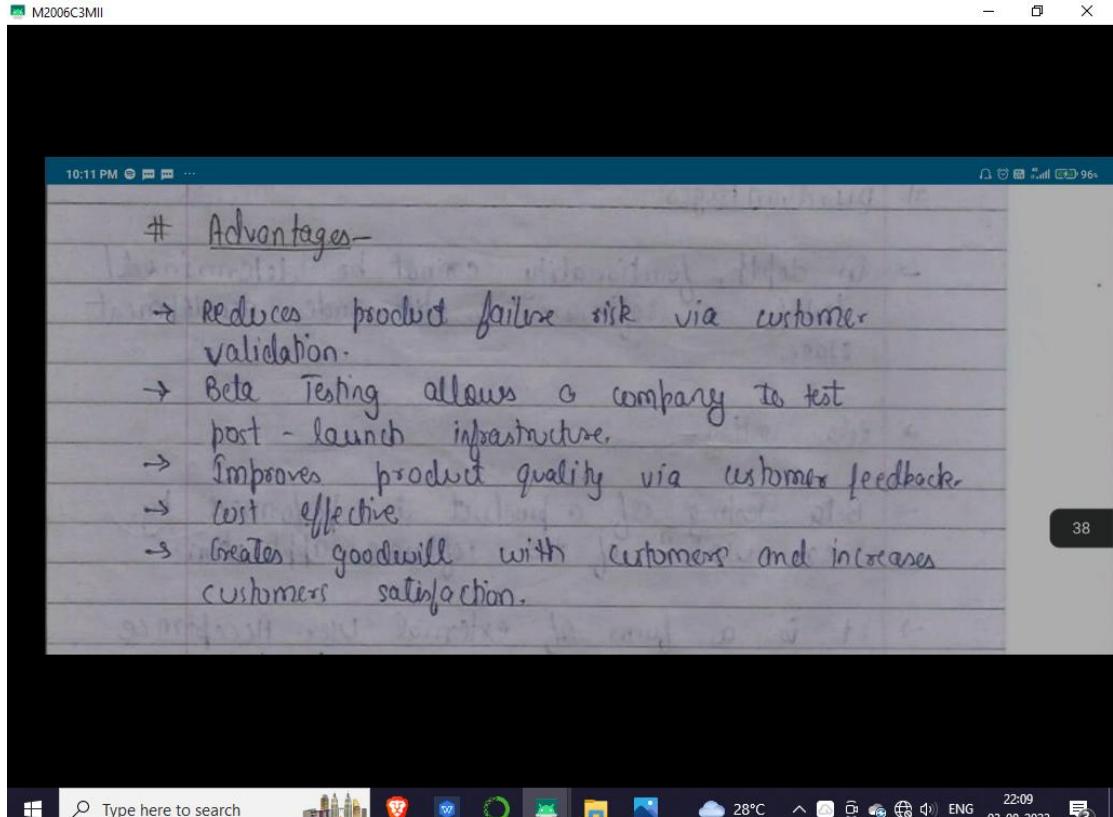
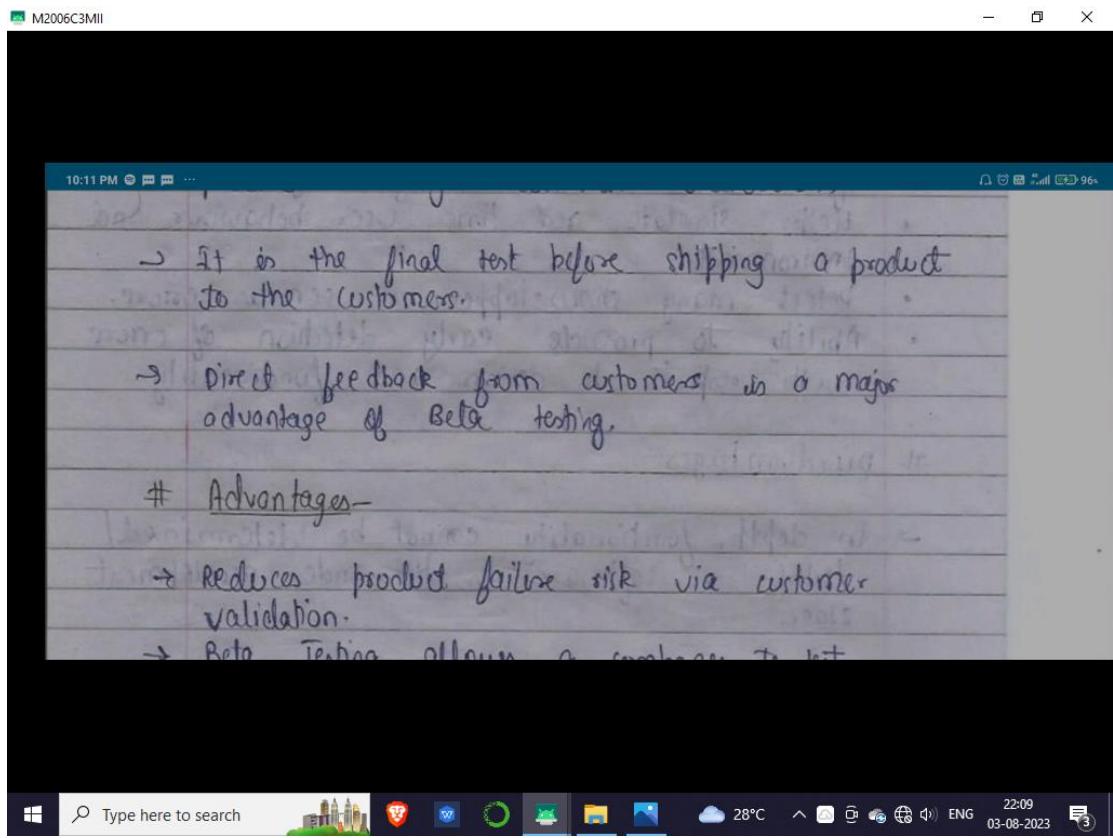
37

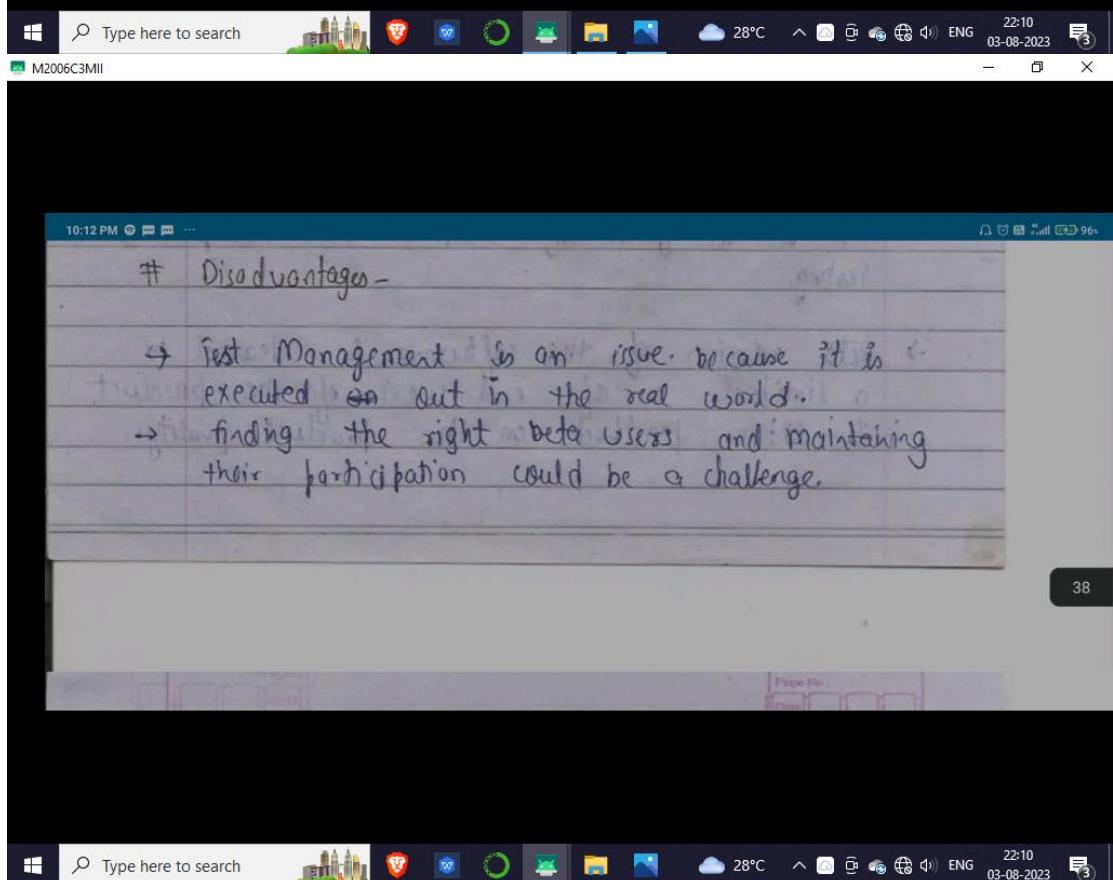
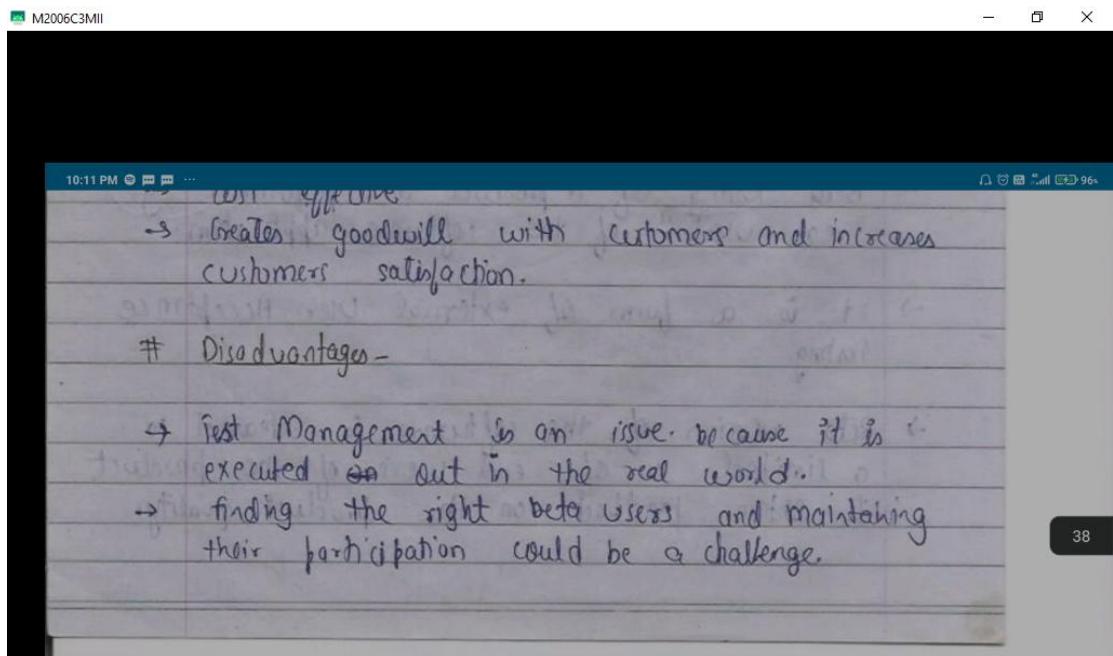


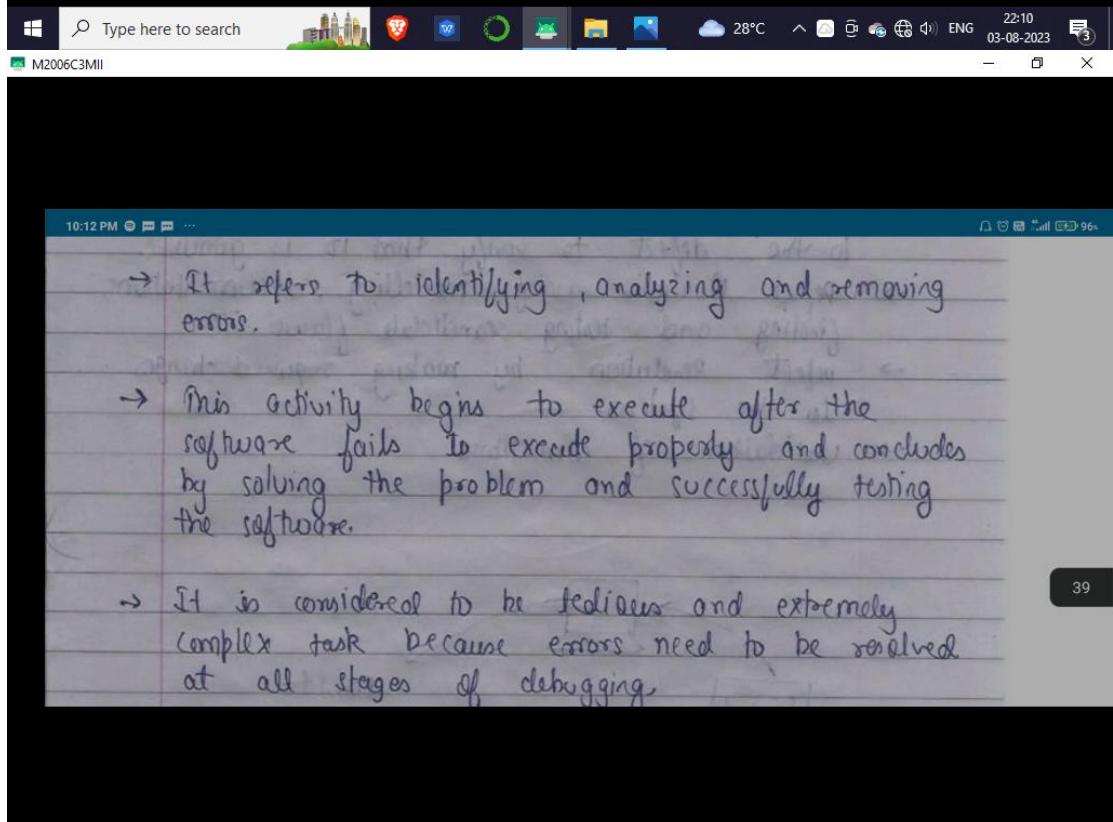
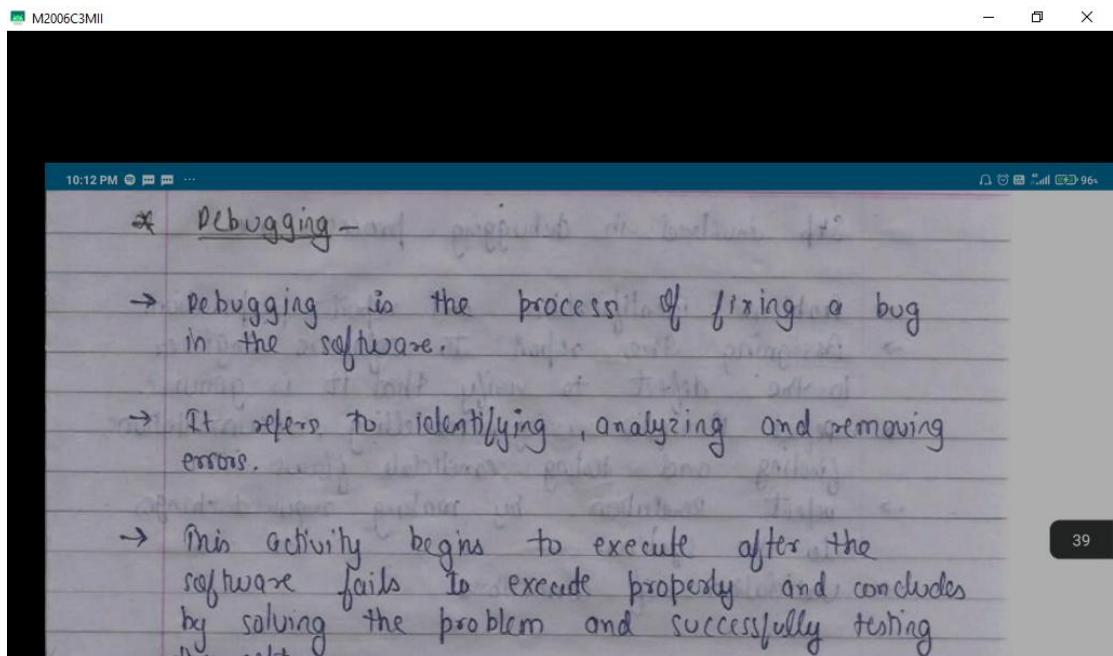


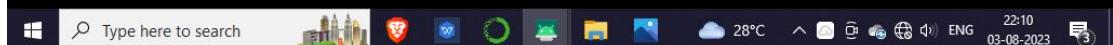
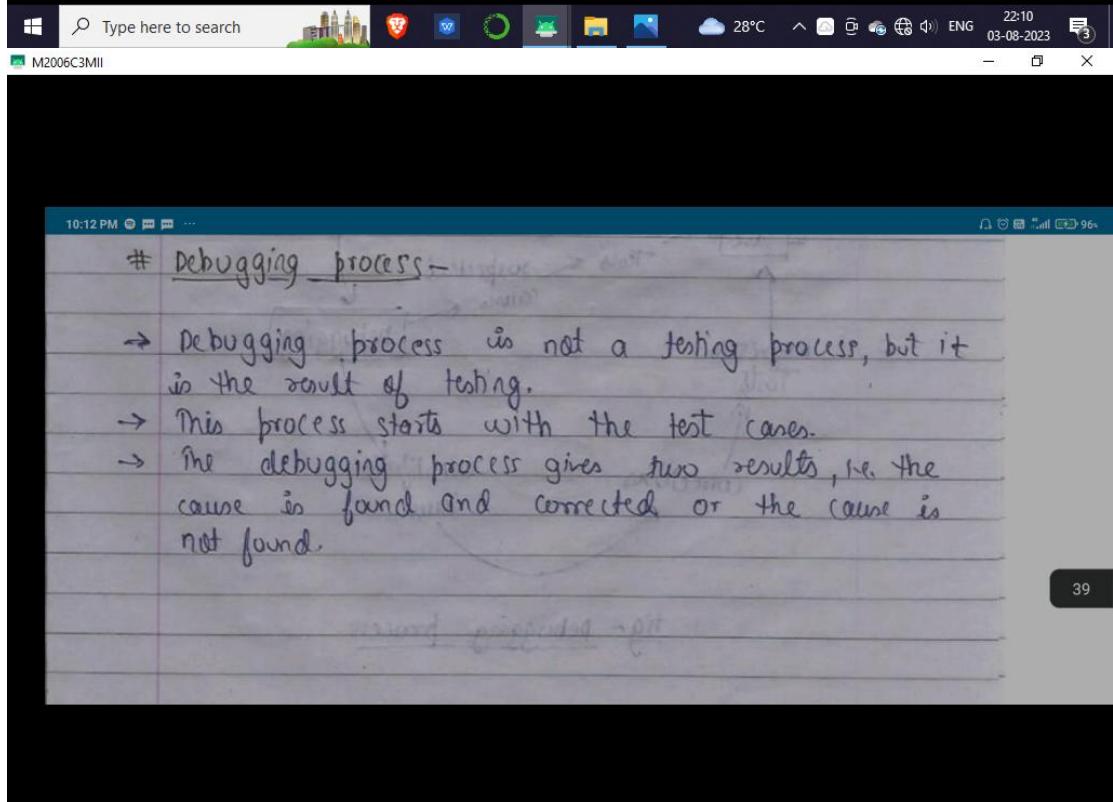
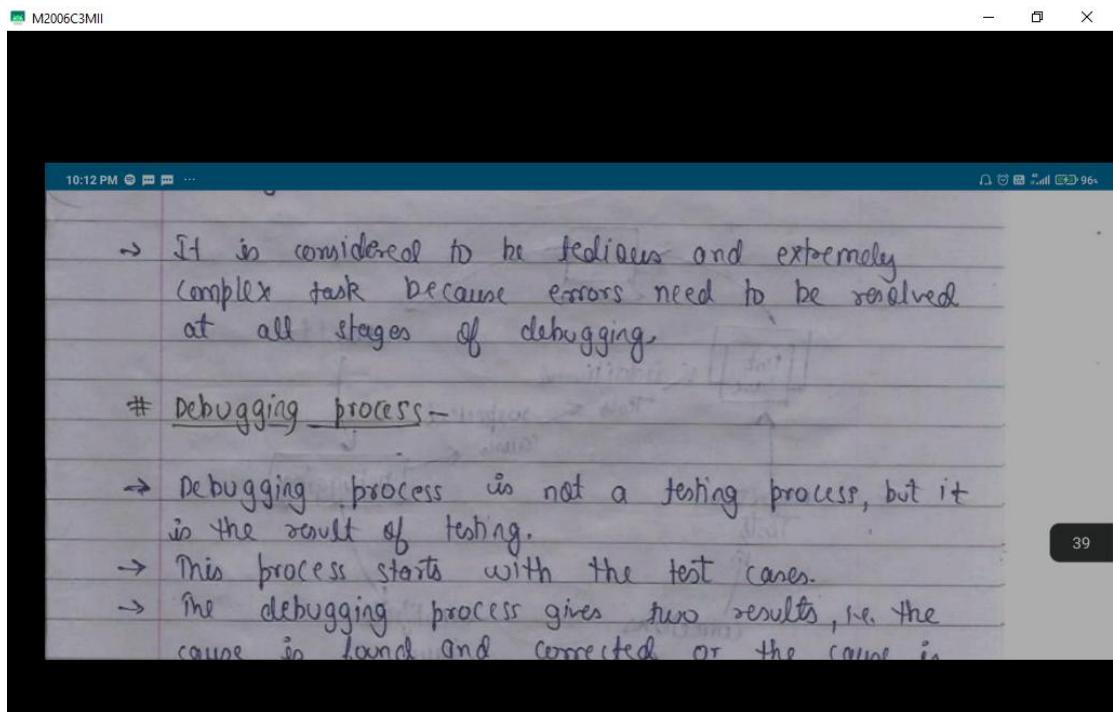
37

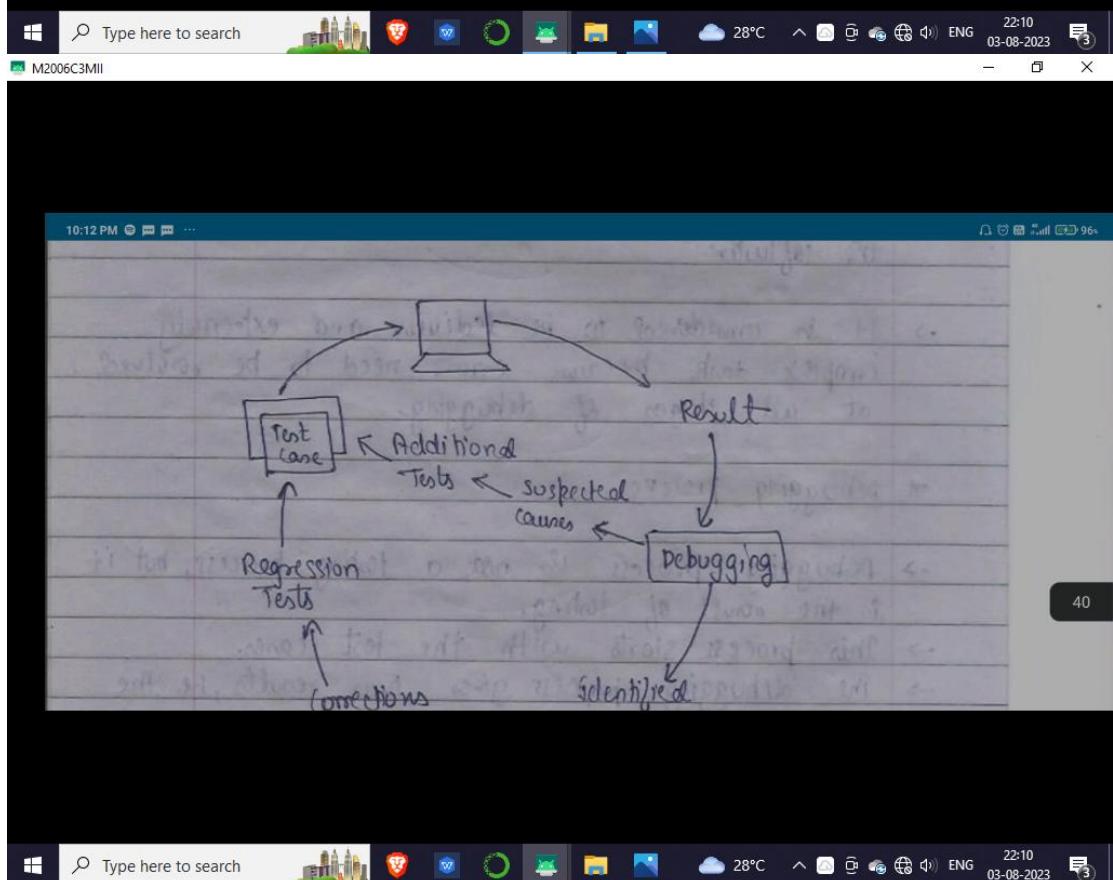
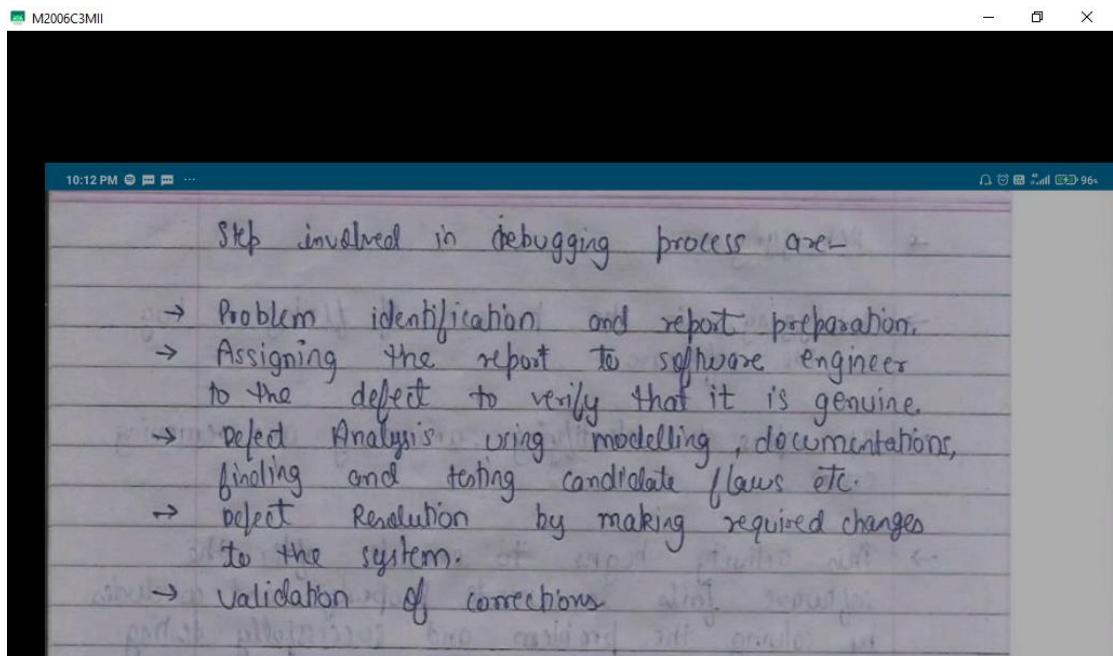


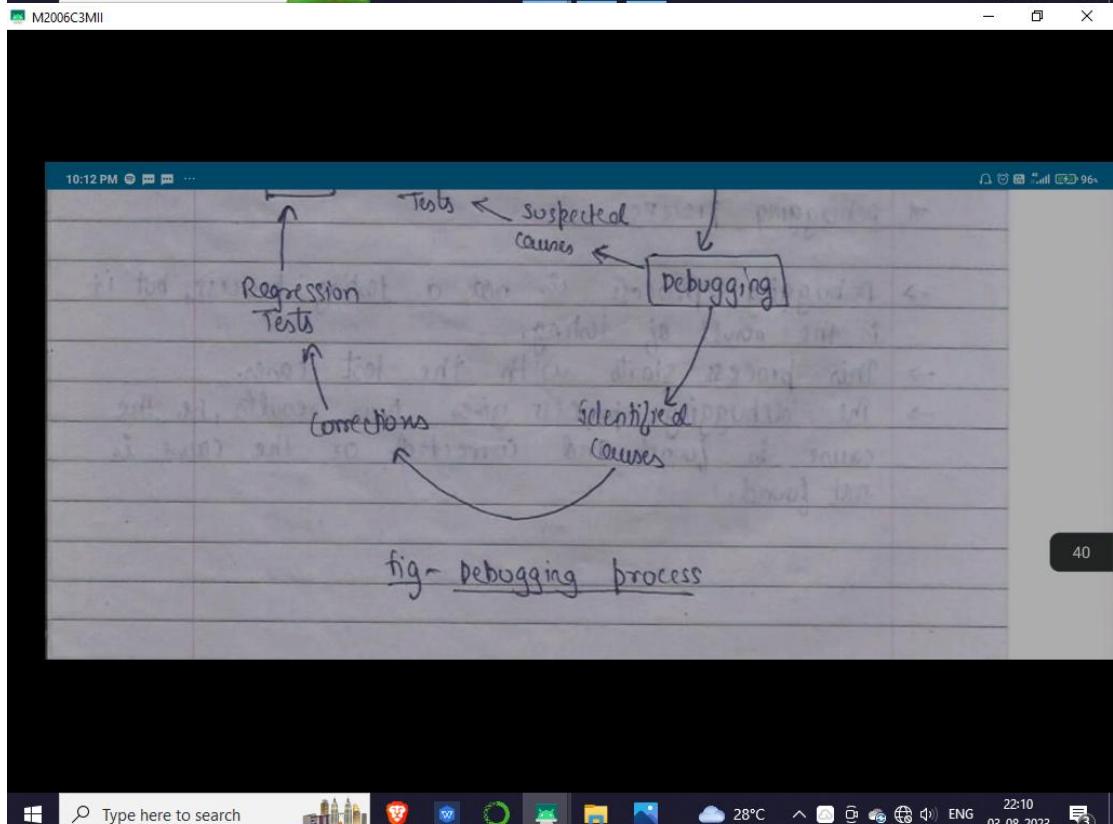
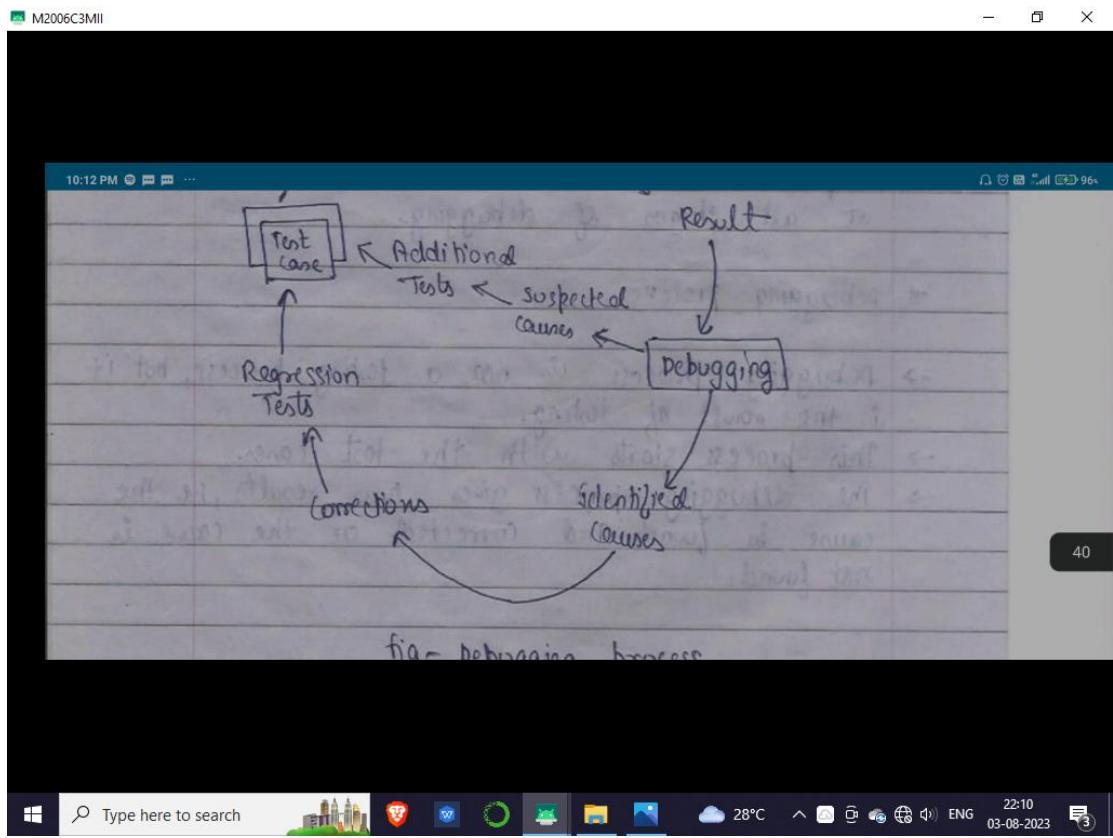


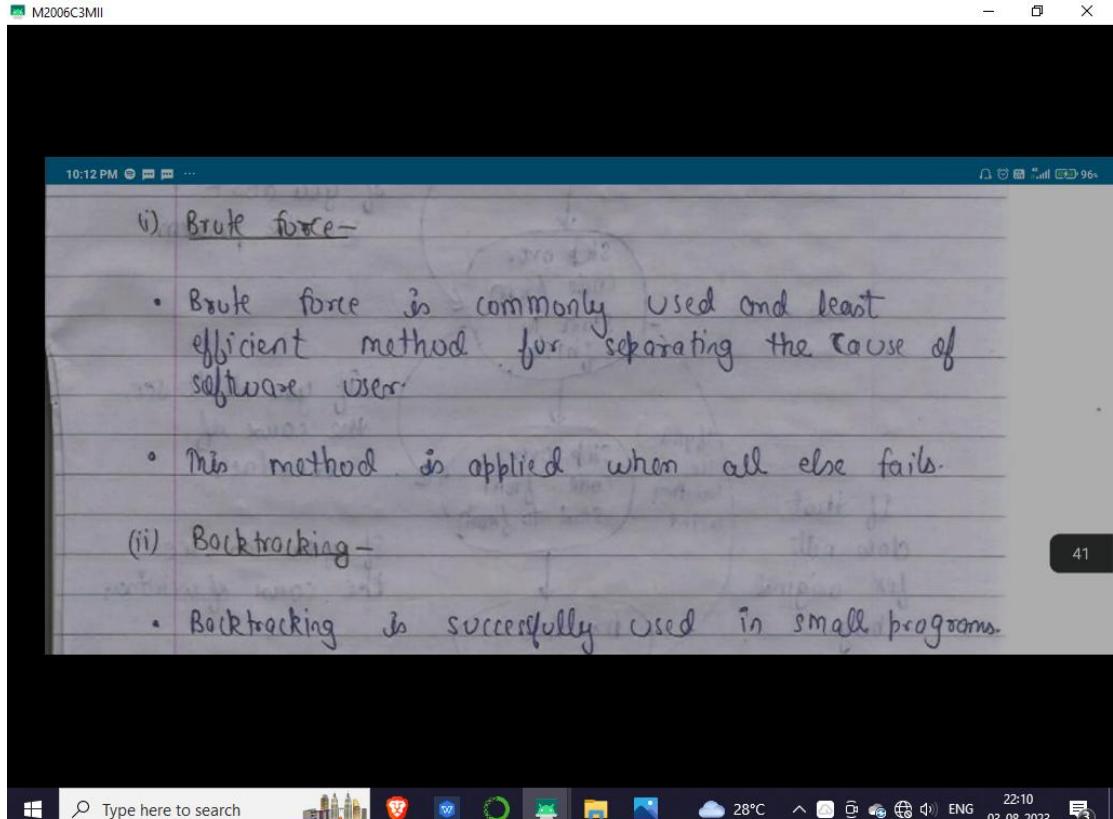
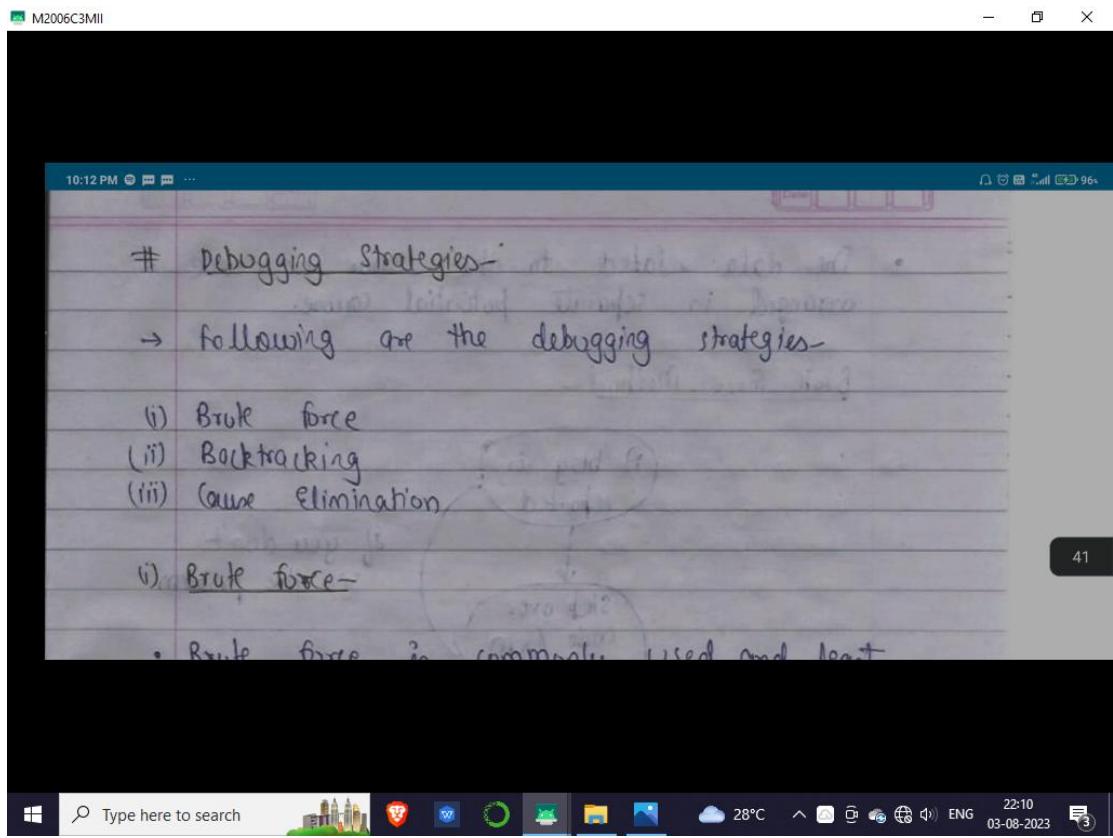


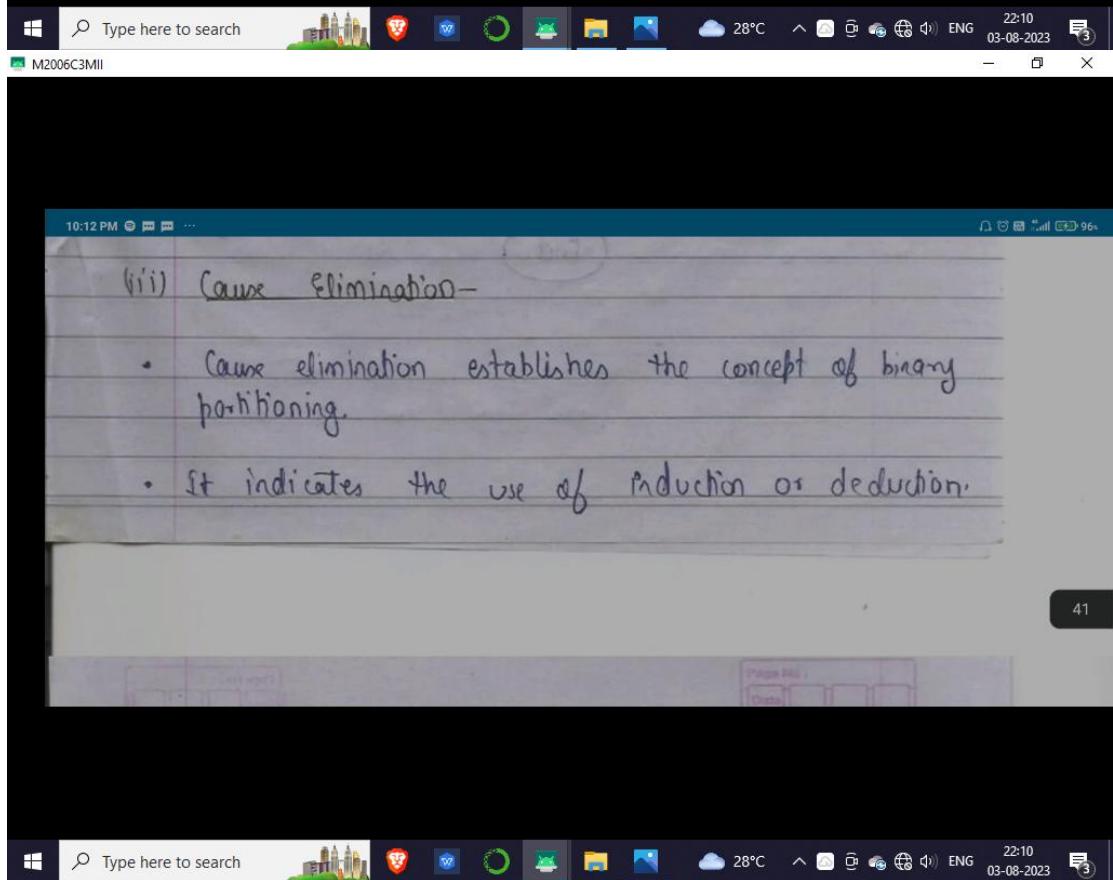
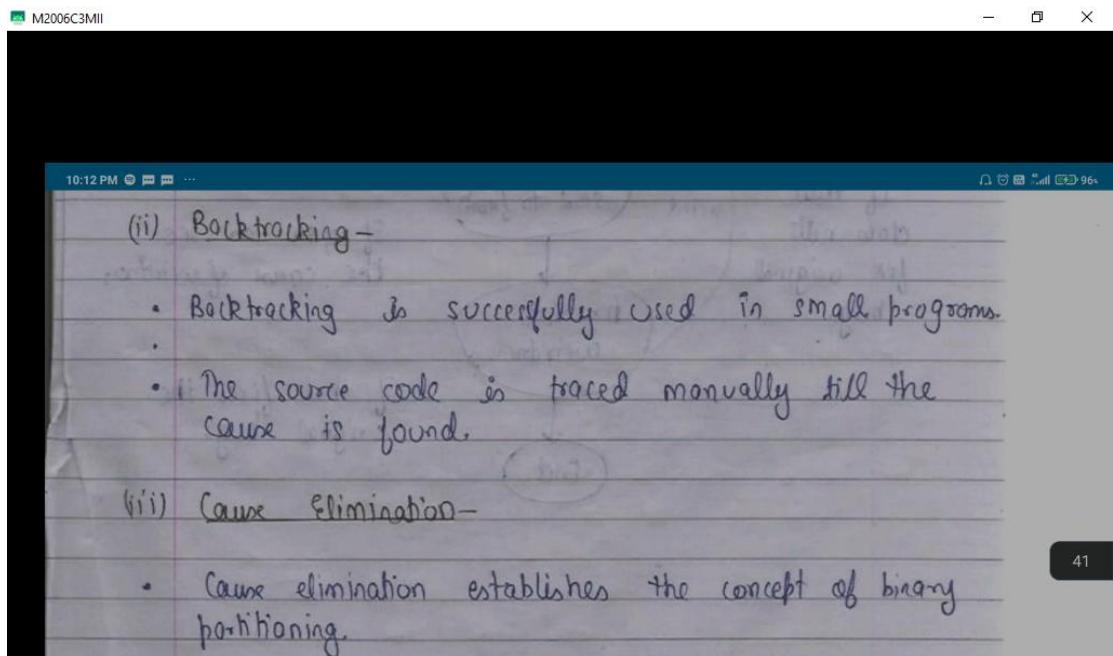


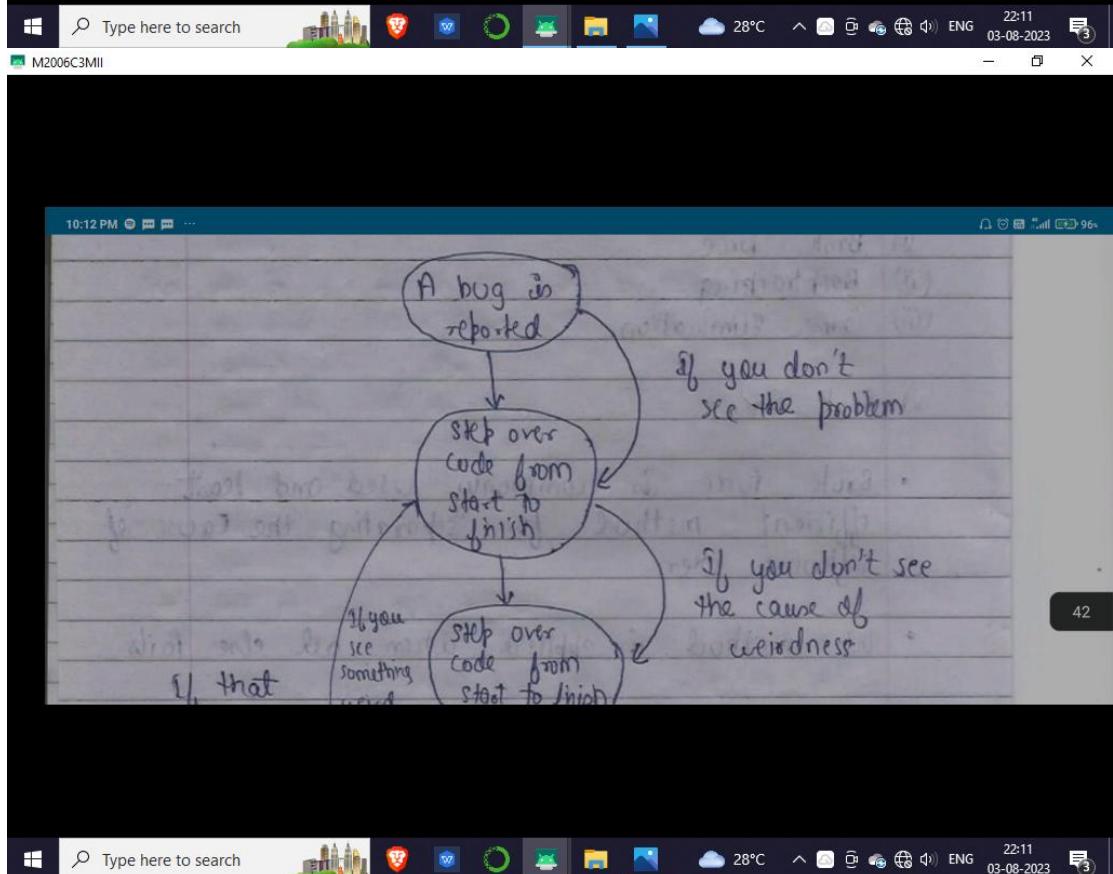
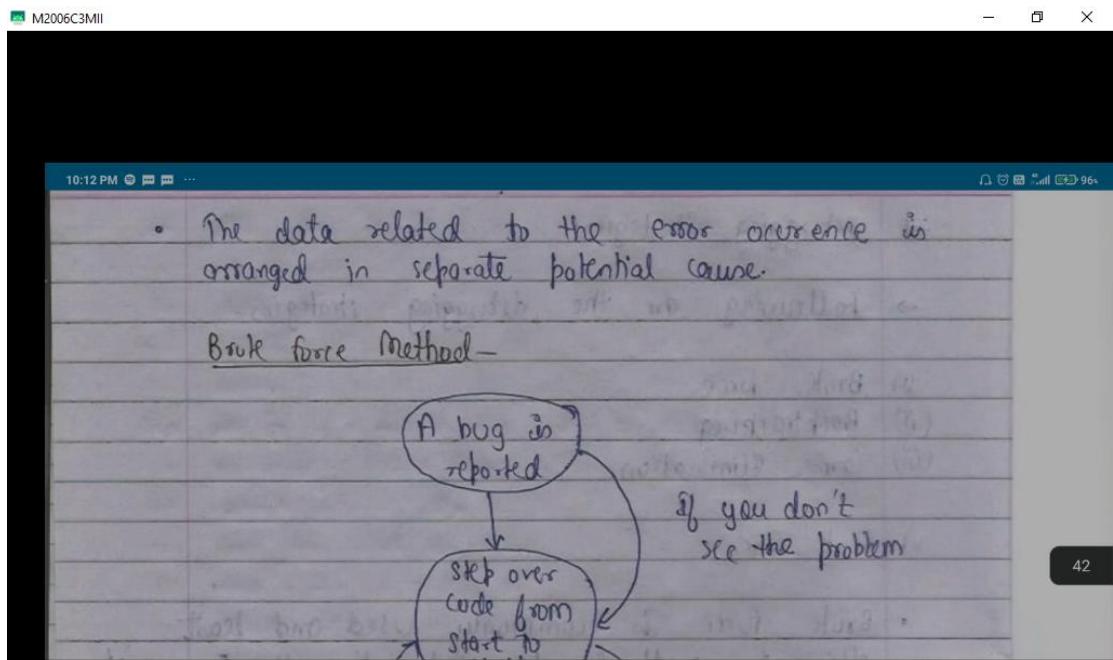


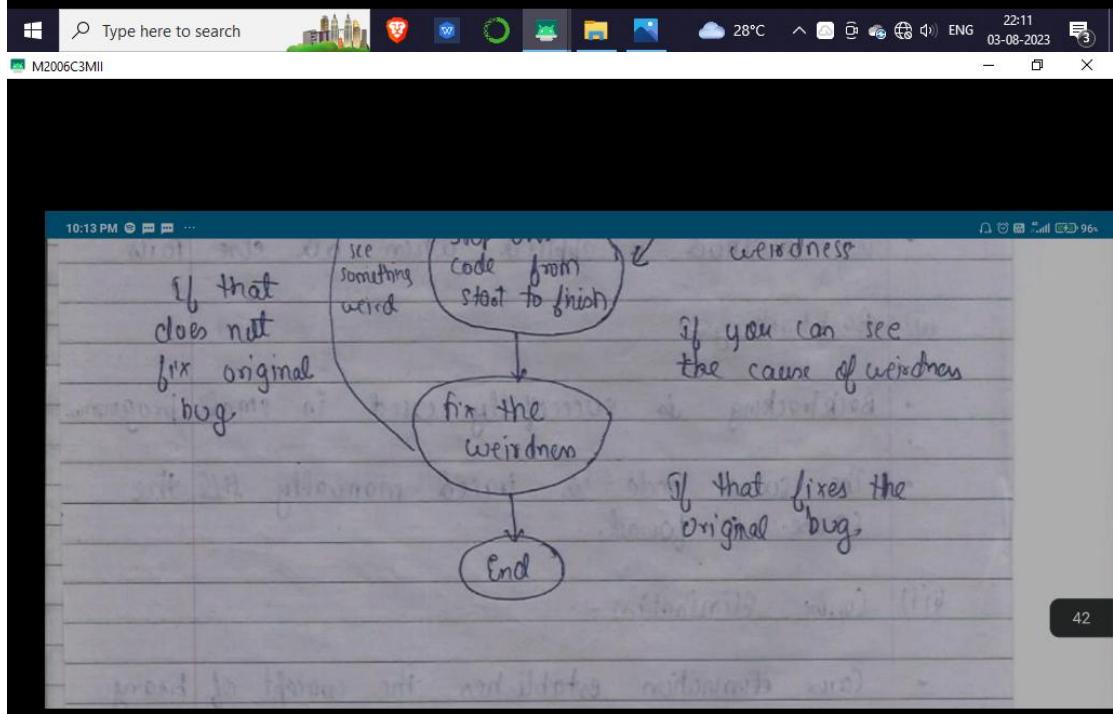
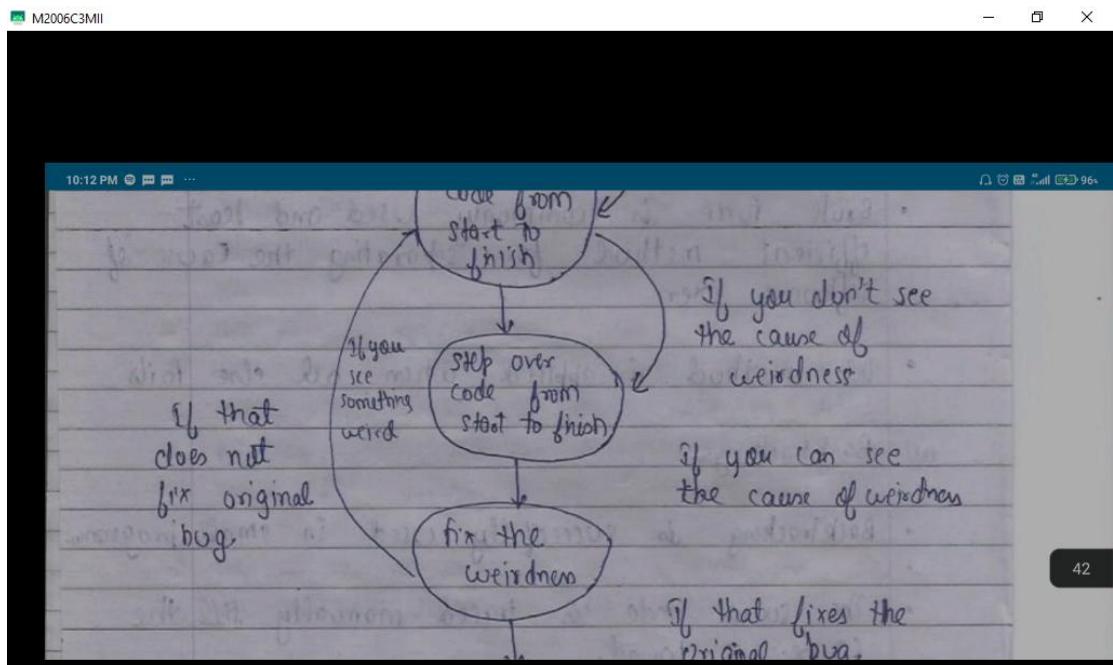












* Difference b/w Testing & Debugging → find (in)

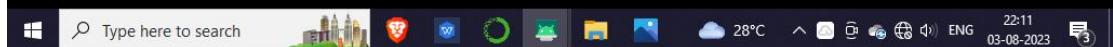
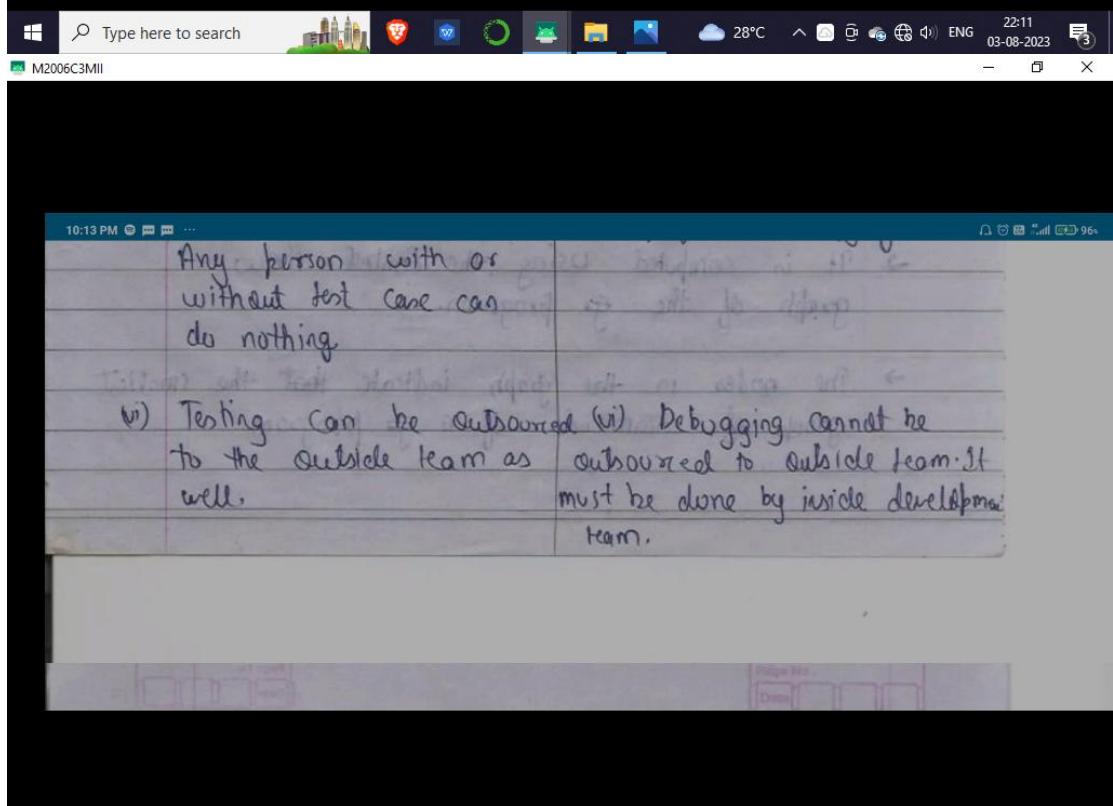
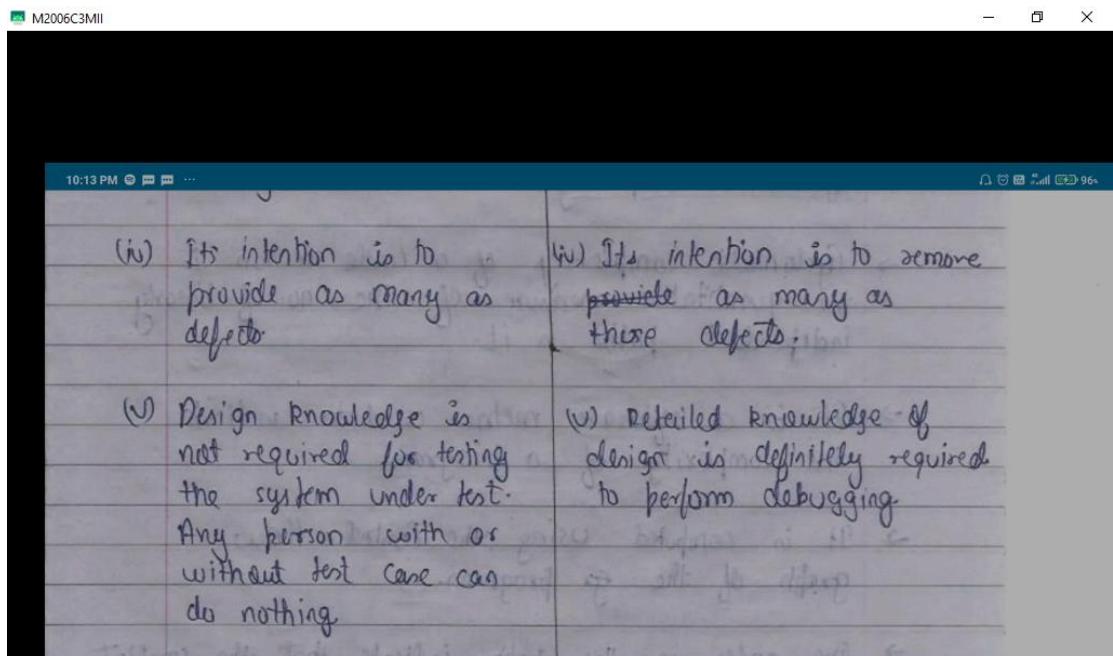
Testing	Debugging
(i) Testing focuses on finding bugs, errors etc	(i) Debugging starts after a bug has been identified in the s/w.
(ii) It is the process of finding and locating of a defect.	(ii) It is the process of fixing that defect.

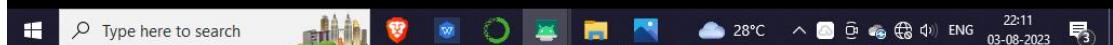
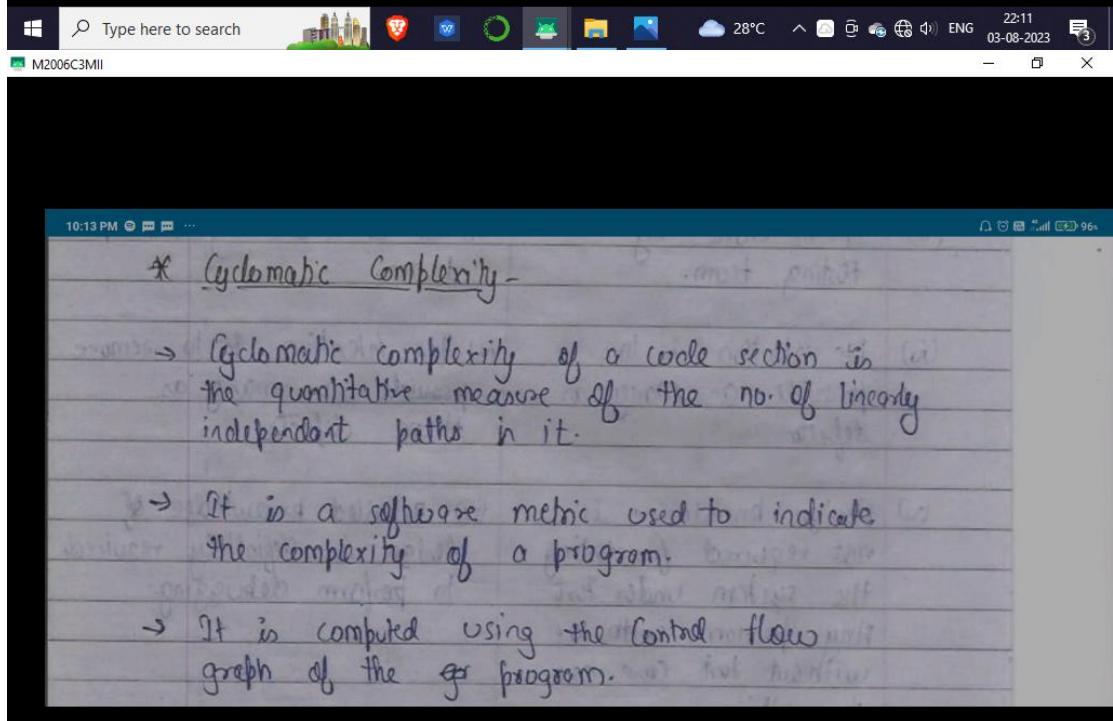
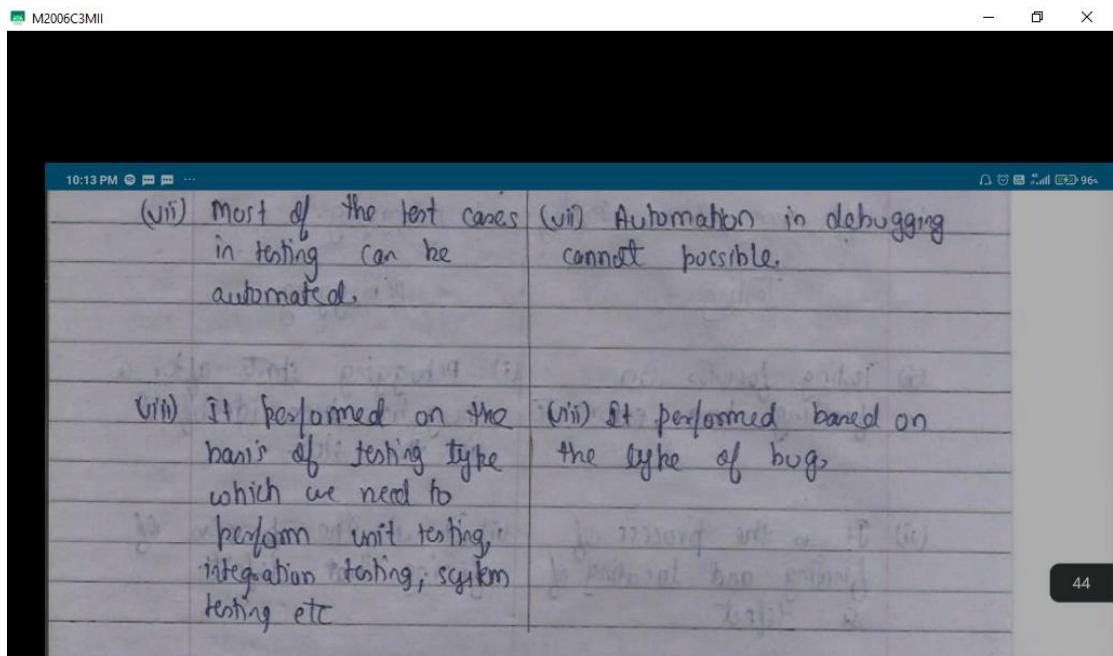
10:13 PM 22:11 03-08-2023

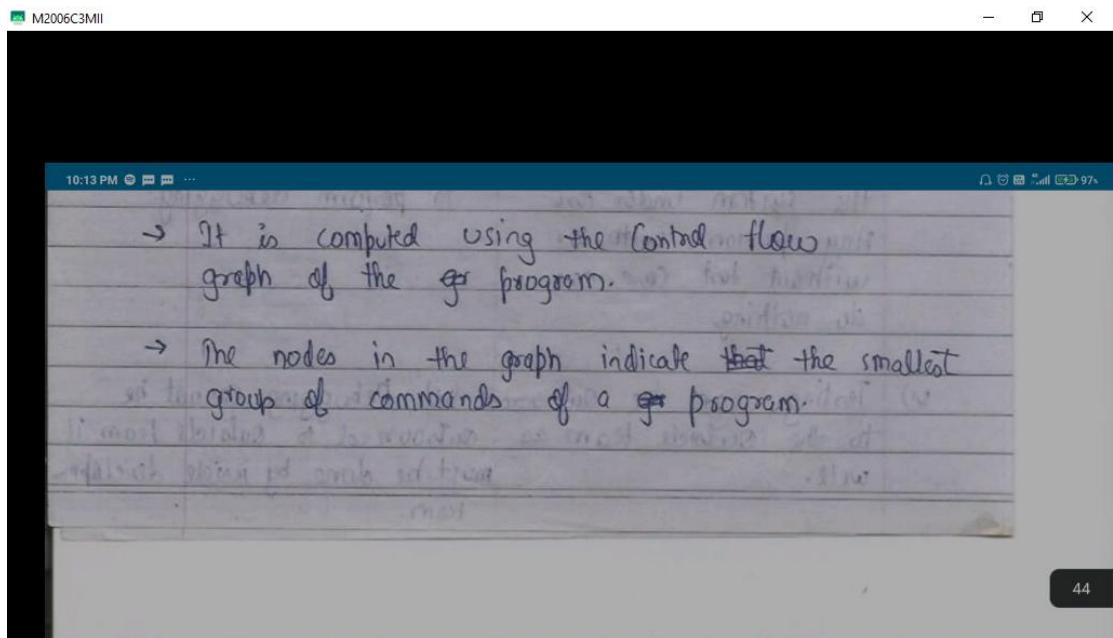
(i) Testing focuses on finding bugs, errors etc	(i) Debugging starts after a bug has been identified in the s/w.
(ii) It is the process of finding and locating of a defect.	(ii) It is the process of fixing that defect.
(iii) It is done by testing team.	(iii) It is done by development team.

10:13 PM 22:11 03-08-2023

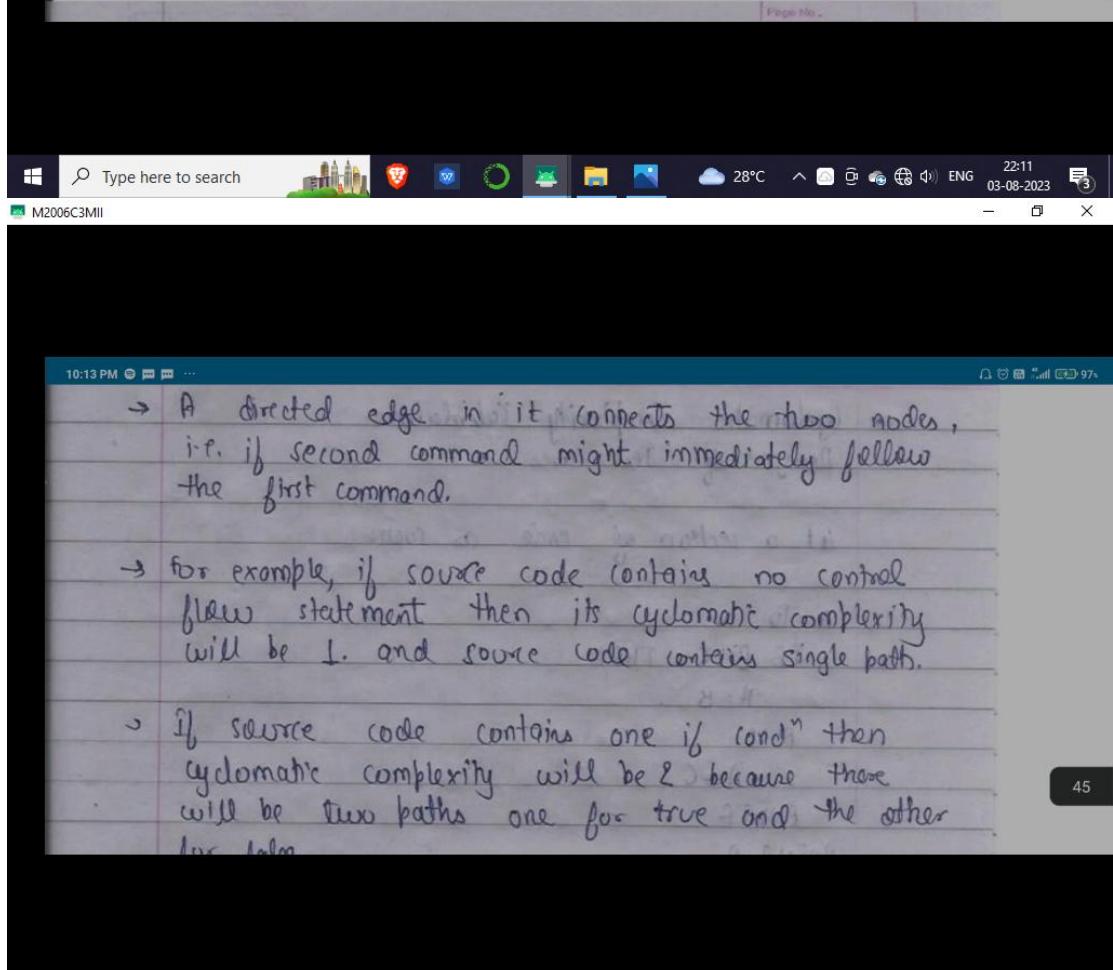
43



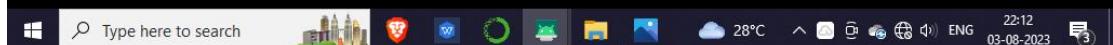
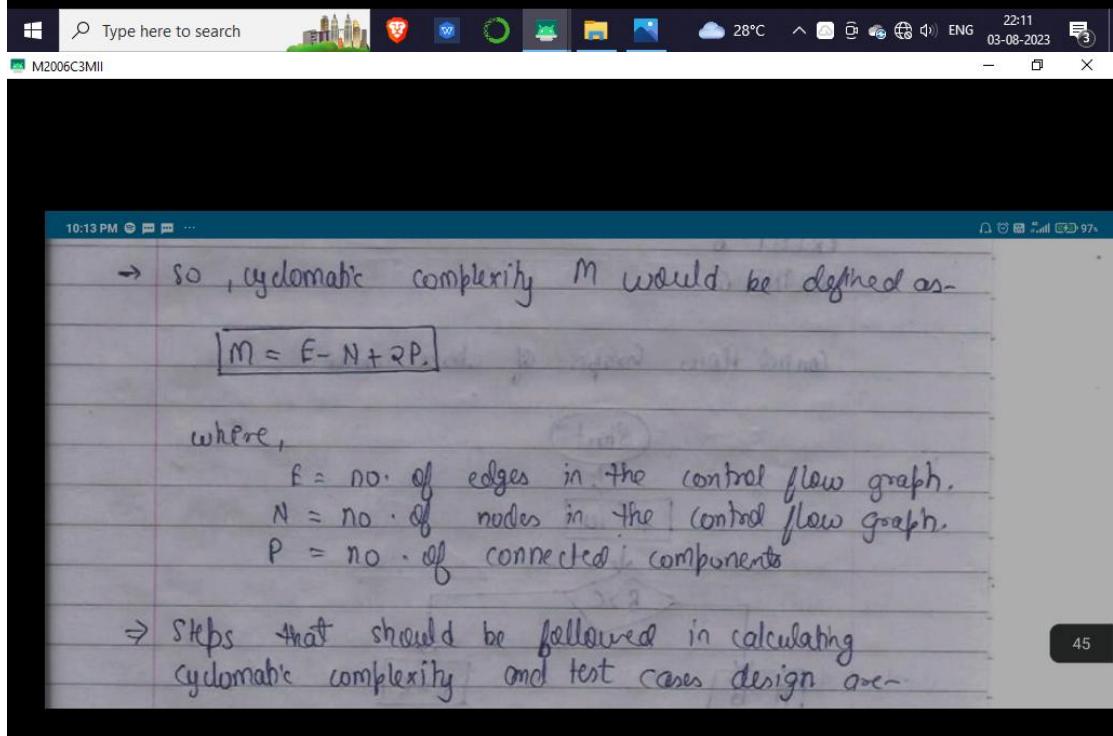
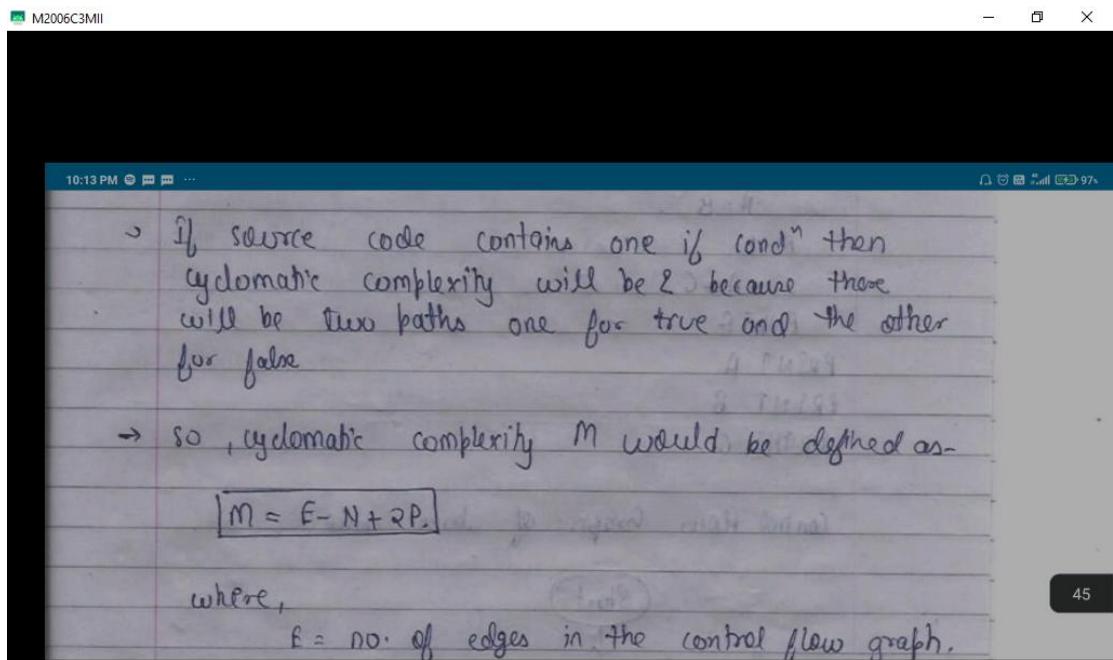


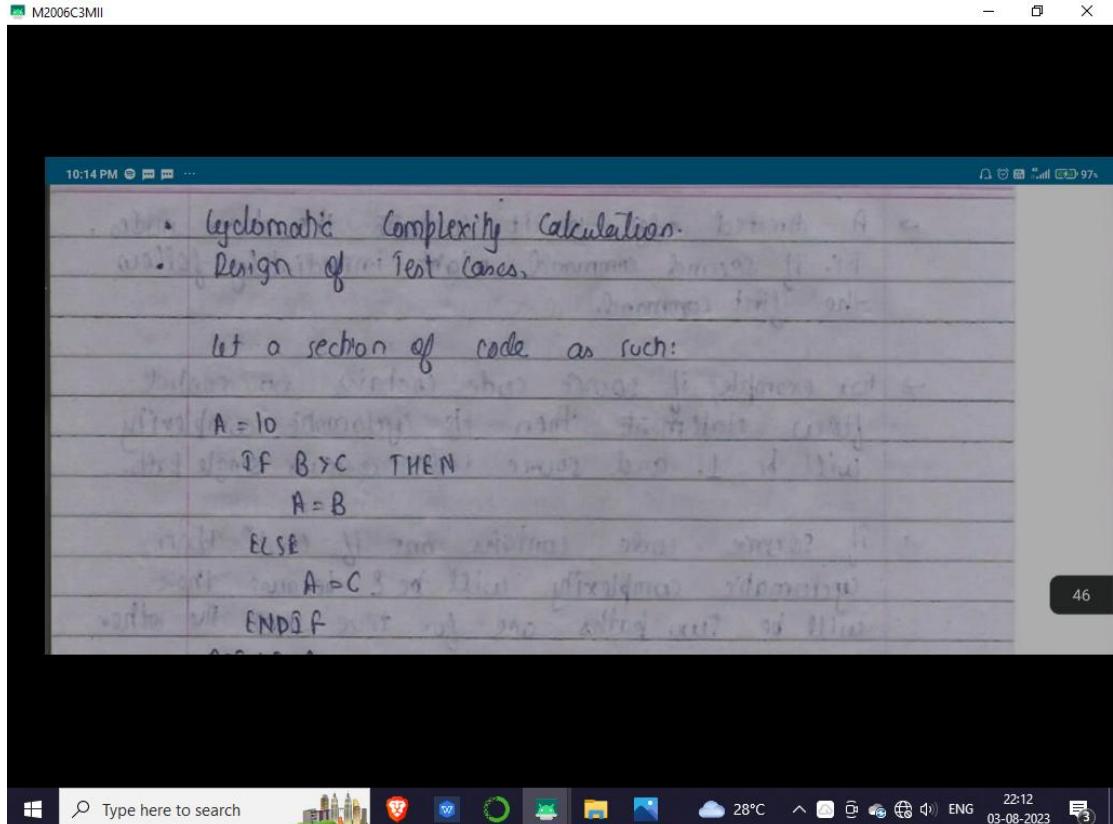
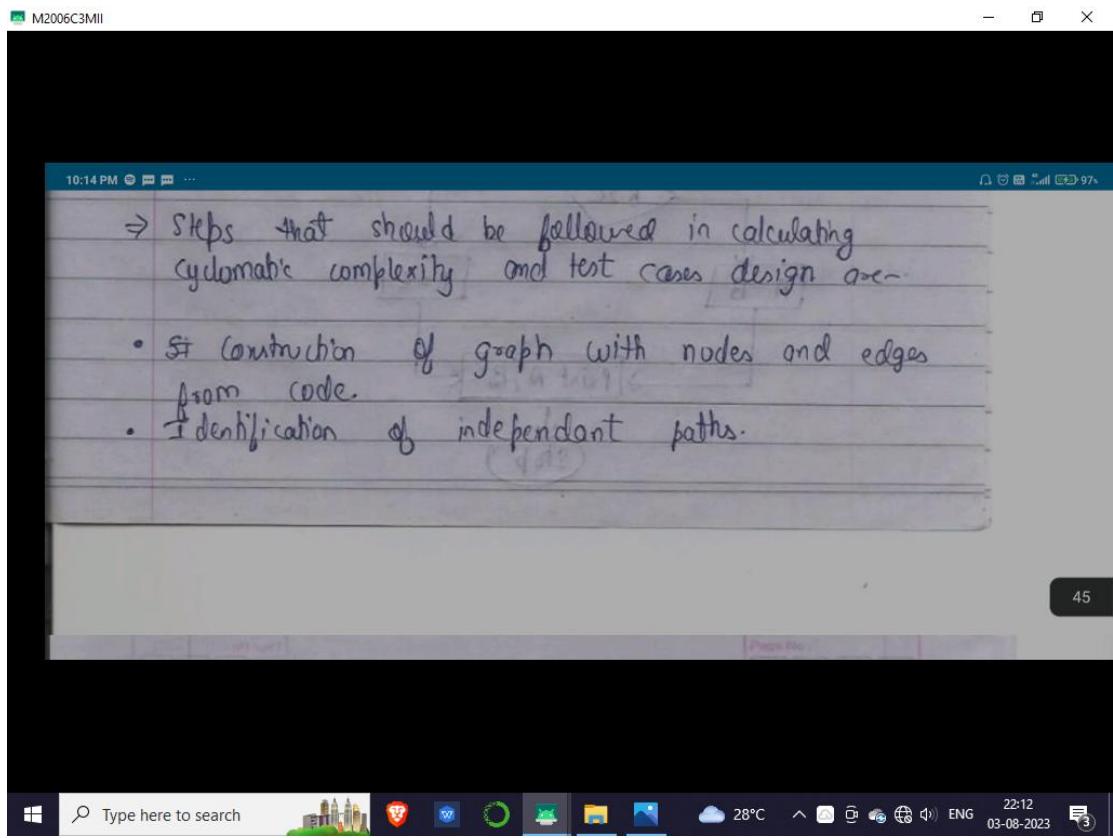


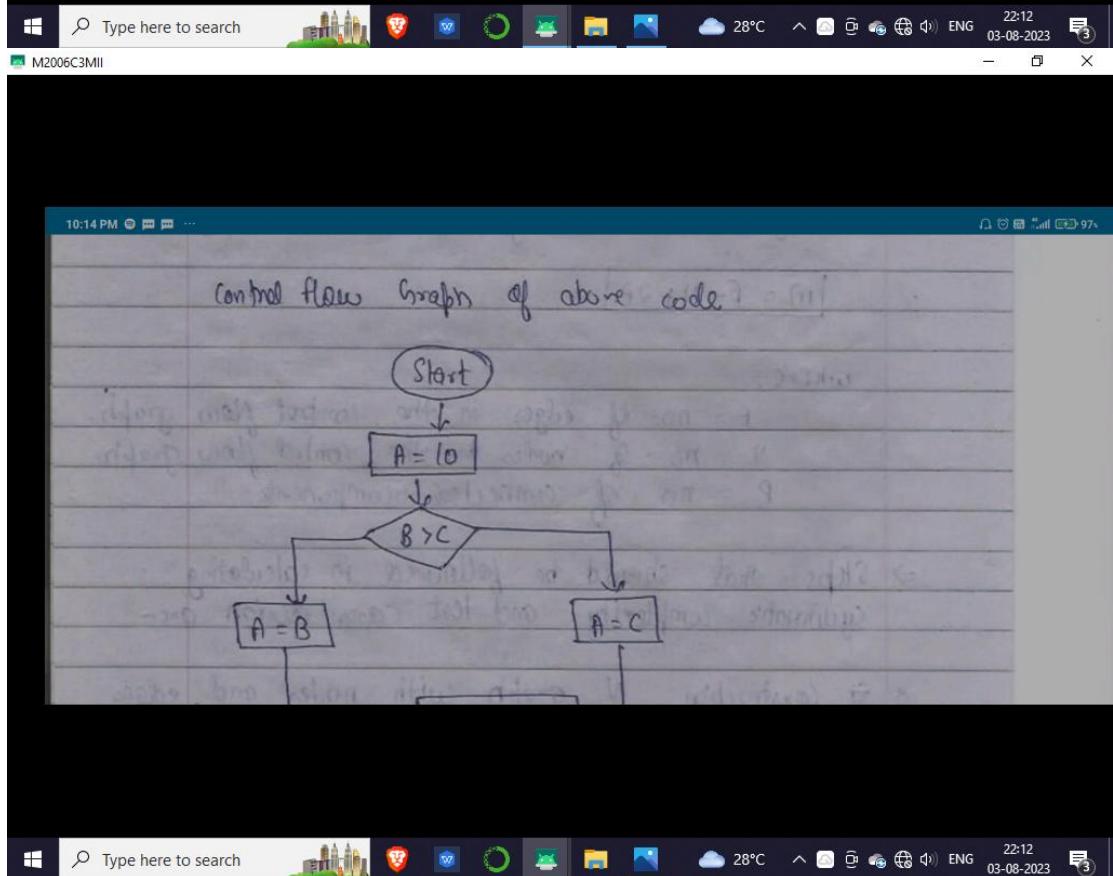
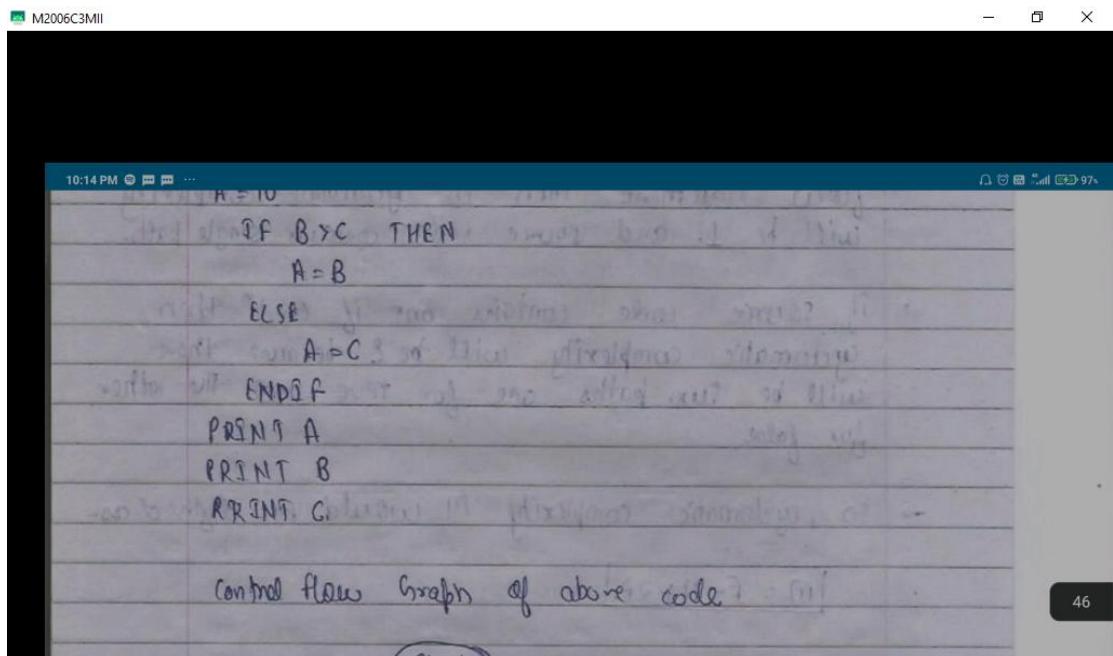
44

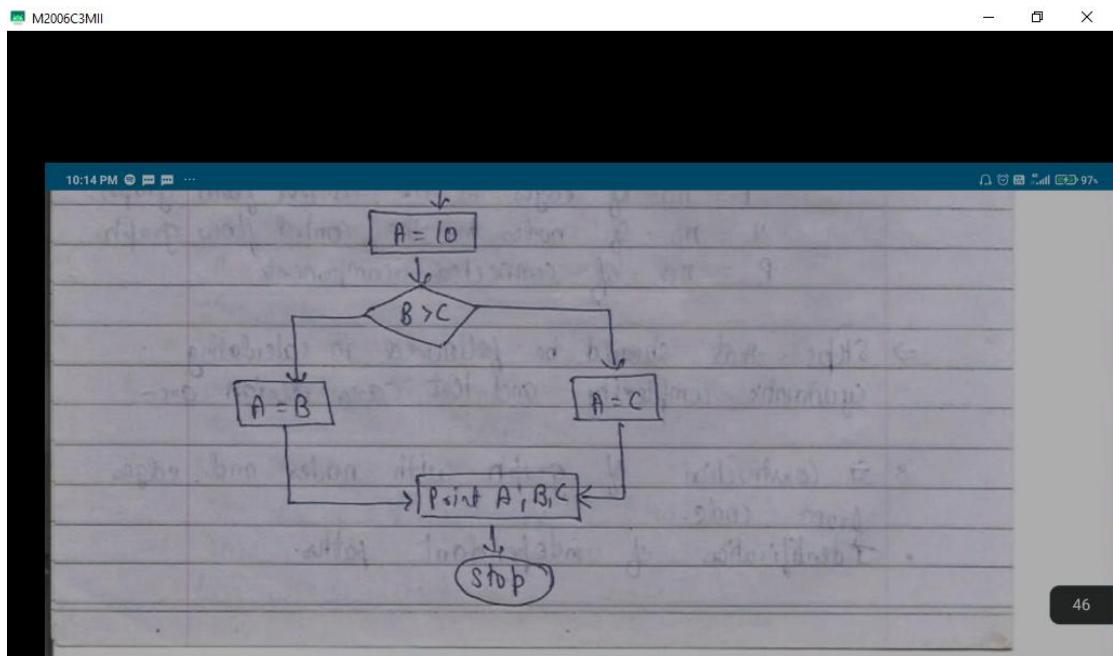


45









Type here to search 28°C ENG 03-08-2023 22:12

M2006C3MII

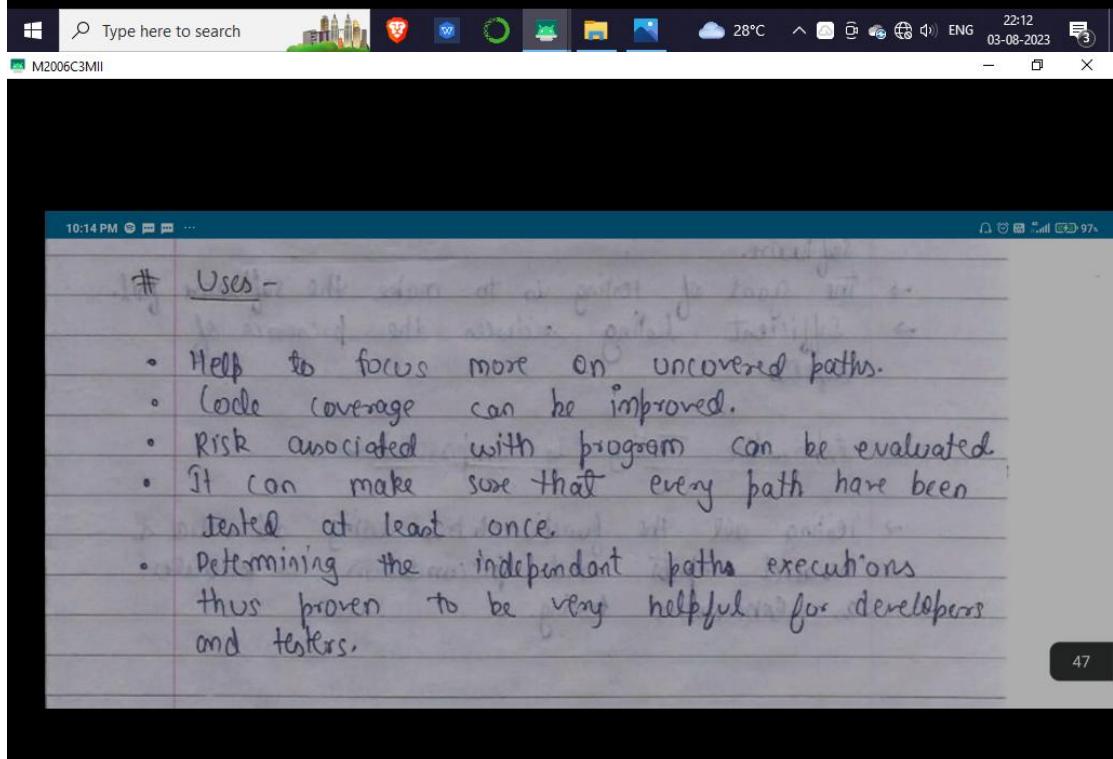
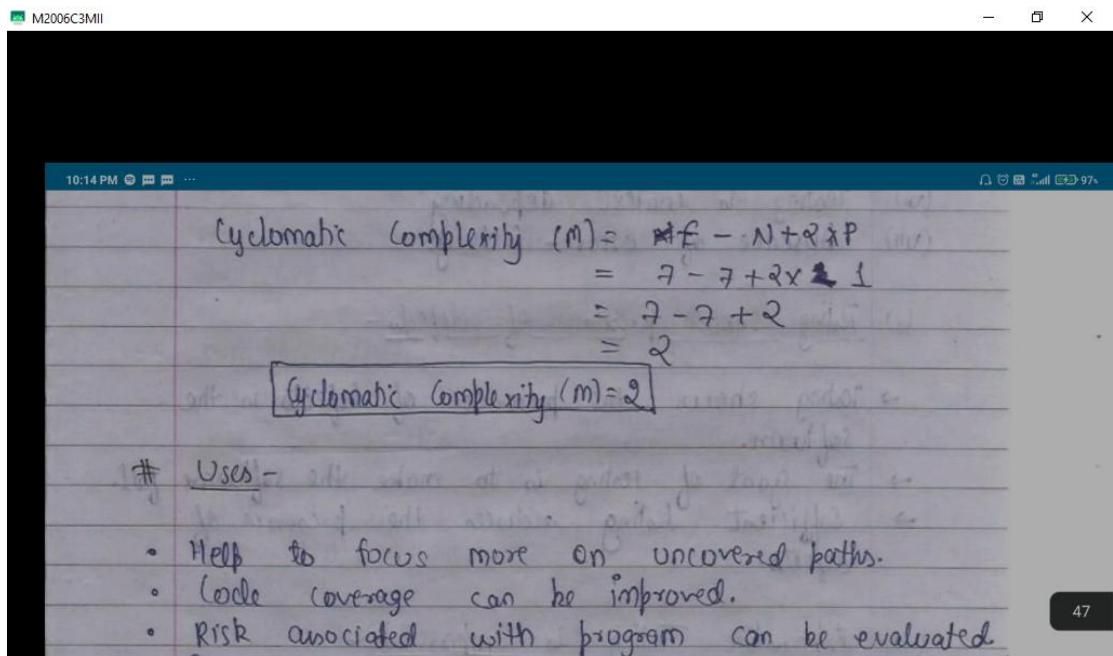
The cyclomatic complexity calculated for above code will be from control flow graph.

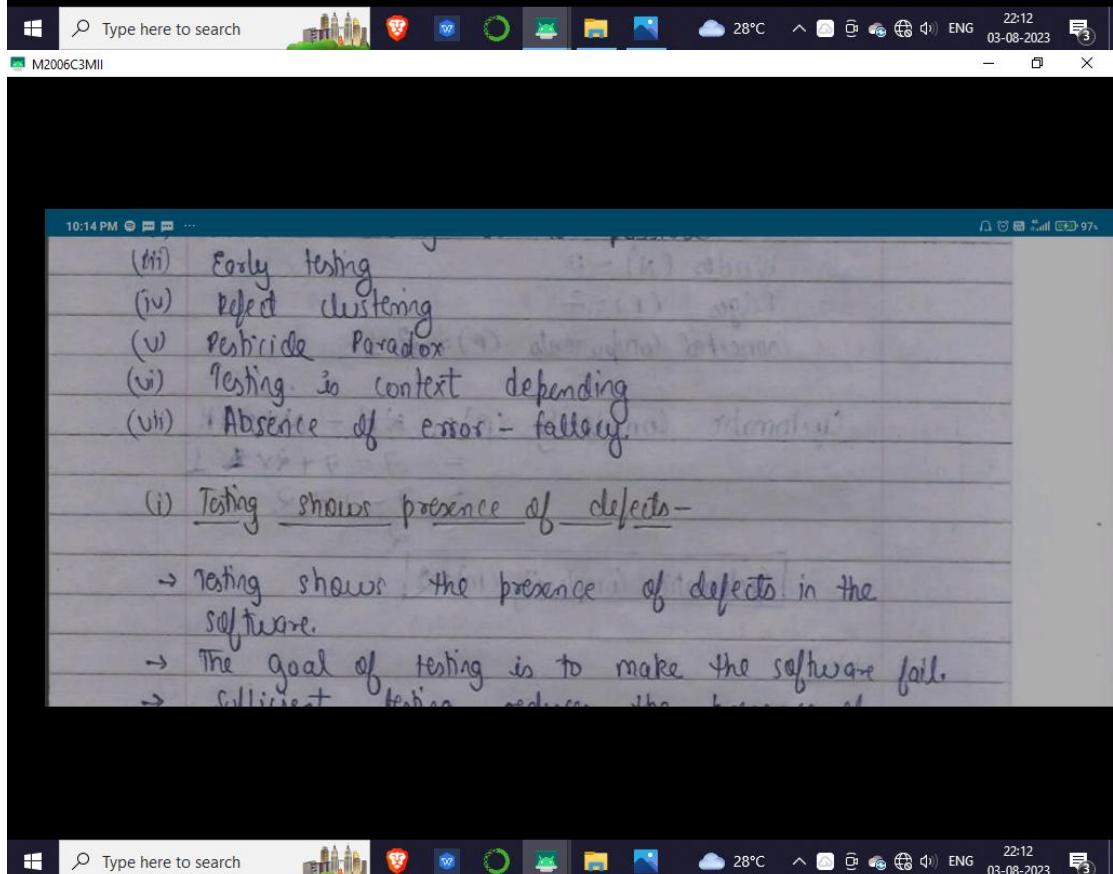
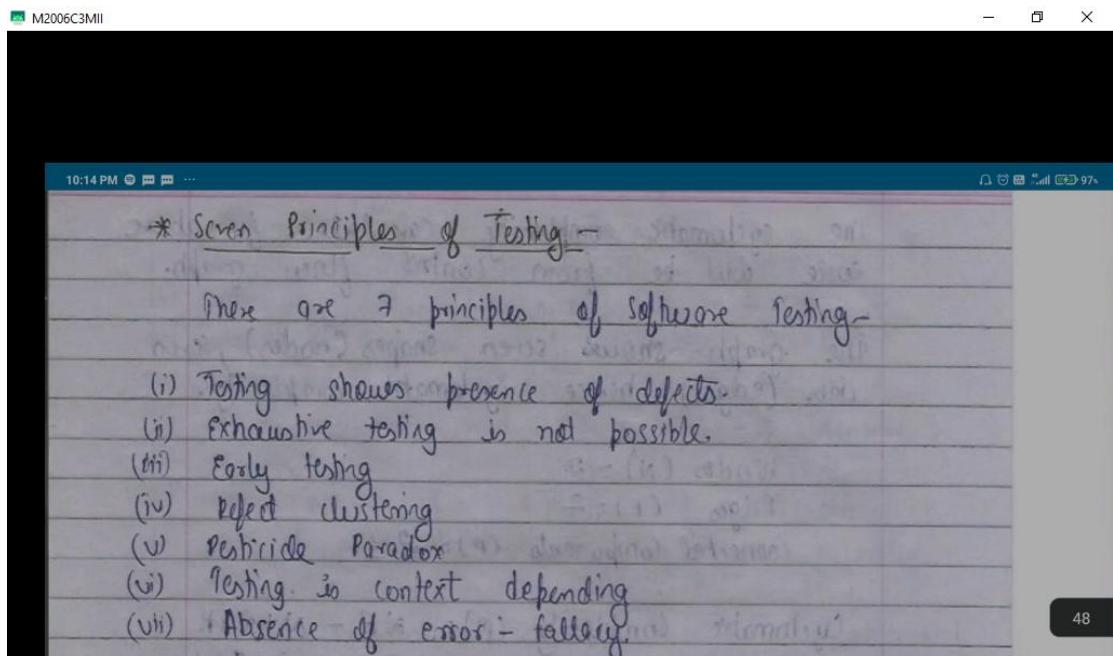
The graph shows seven shapes (nodes), seven lines (edges), hence cyclomatic complexity.

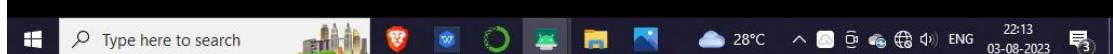
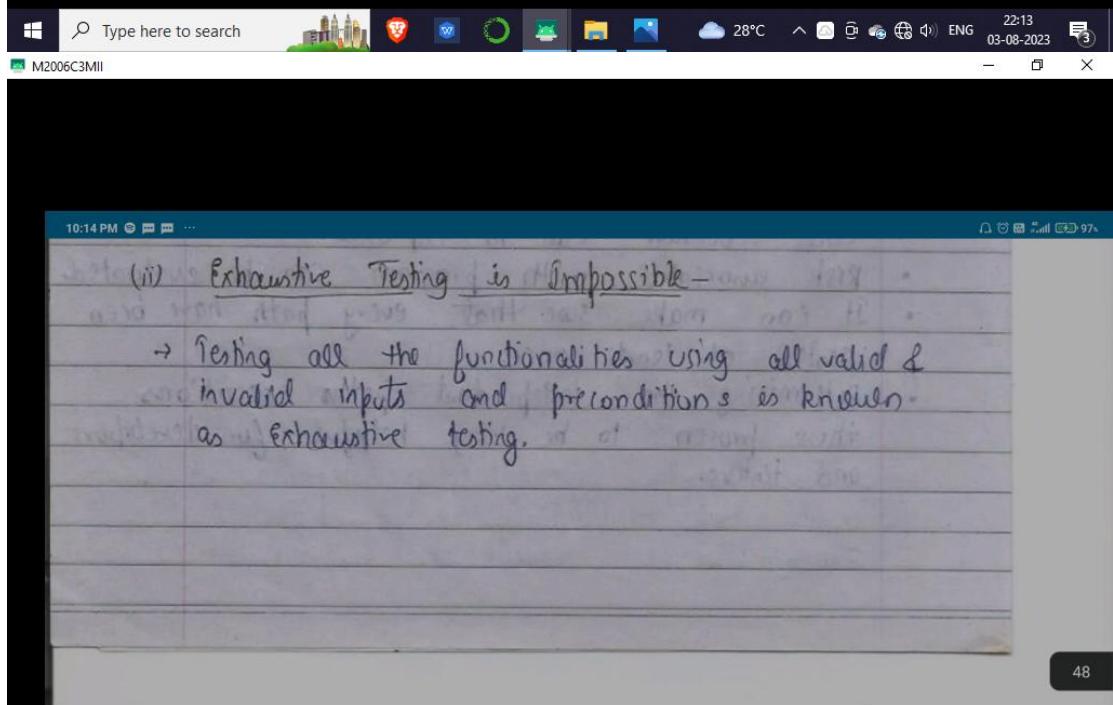
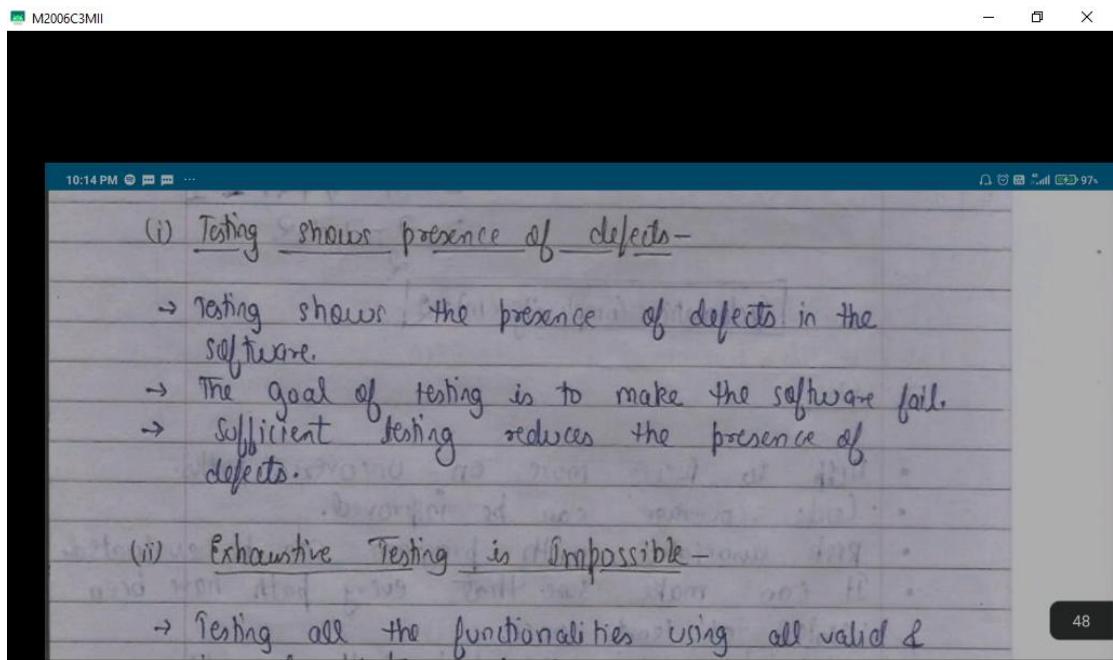
Nodes (N) = 7
 Edges (E) = 7
 Connected Components (P) = 2

Cyclomatic Complexity (M) = $E - N + 2 \times P$
 $= 7 - 7 + 2 \times 1$

Type here to search 28°C ENG 03-08-2023 22:12







10:15 PM

22:13 97%

(iii) Early Testing -

- Defects detected in early phases of SDLC are less expensive to fix
- So conducting early testing reduces the cost of fixing defects.

(iv) Defect Clustering -

- Defect clustering in software testing means that a small module or functionality contains most of the bugs or it has the most operational failures.

49



Type here to search



28°C 03-08-2023

10:15 PM

22:13 97%

(iv) Defect Clustering -

- Defect clustering in software testing means that a small module or functionality contains most of the bugs or it has the most operational failures.

(v) Pesticide Paradox -

Pesticide Paradox in software engineering testing is the process of retesting the same test cases again

49



Type here to search



28°C 03-08-2023

M2006C3MII

10:15 PM 97%

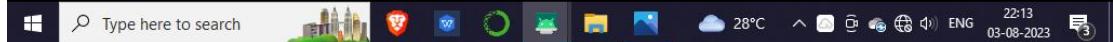
(v) Pesticide Paradox—

Pesticide Paradox in software engineering testing is the process of repeating the same test cases again and again. Eventually, the same test cases will no longer find new bugs.

(vi) Testing is context Dependant—

- Testing ~~context~~ approach depends on the context of the software we develop.
- for ex, online Banking application requires

49



10:15 PM 100% 97% (vi) Testing is context Dependant -

- Testing ~~context~~ approach depends on the context of the software we develop.
- for ex, online Banking application requires different approach of testing compared to an e-commerce site.

49

Type here to search 28°C 22:13 03-08-2023

10:15 PM 100% 97%

(vii) Absence of an error-fallacy -

- 99.9% of bug-free software may still be unusable, if wrong requirements were incorporated into the software and the software is not addressing the business needs.

50

Type here to search 28°C 22:13 03-08-2023