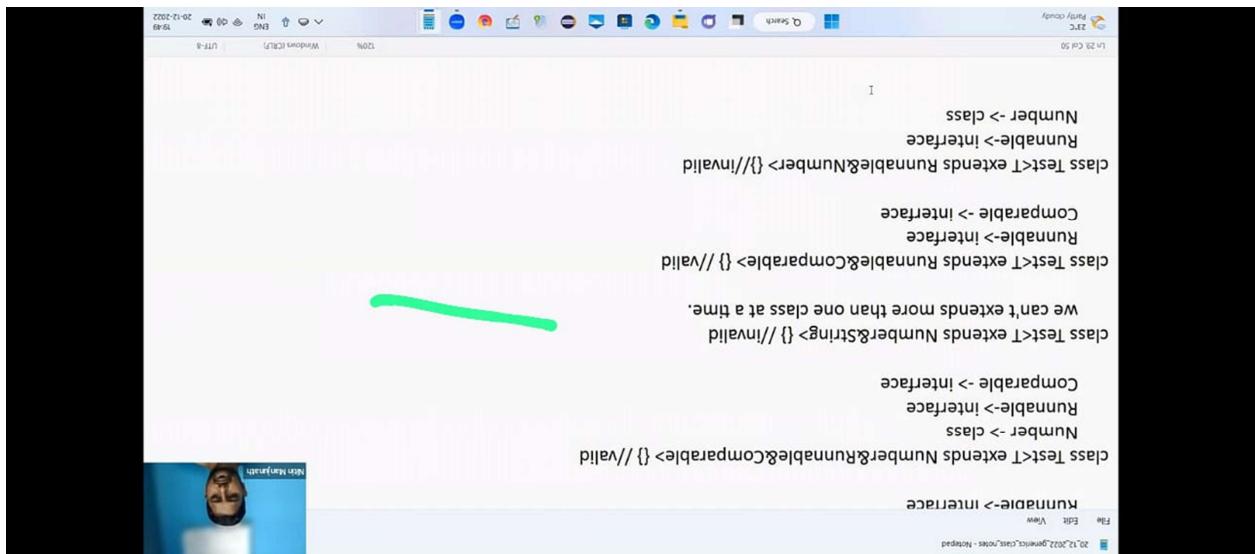


44Java_IO_Operation_part1



```
20-12-2022-generic-classes-notepad - Notepad
File Edit View
Runnable-> interface

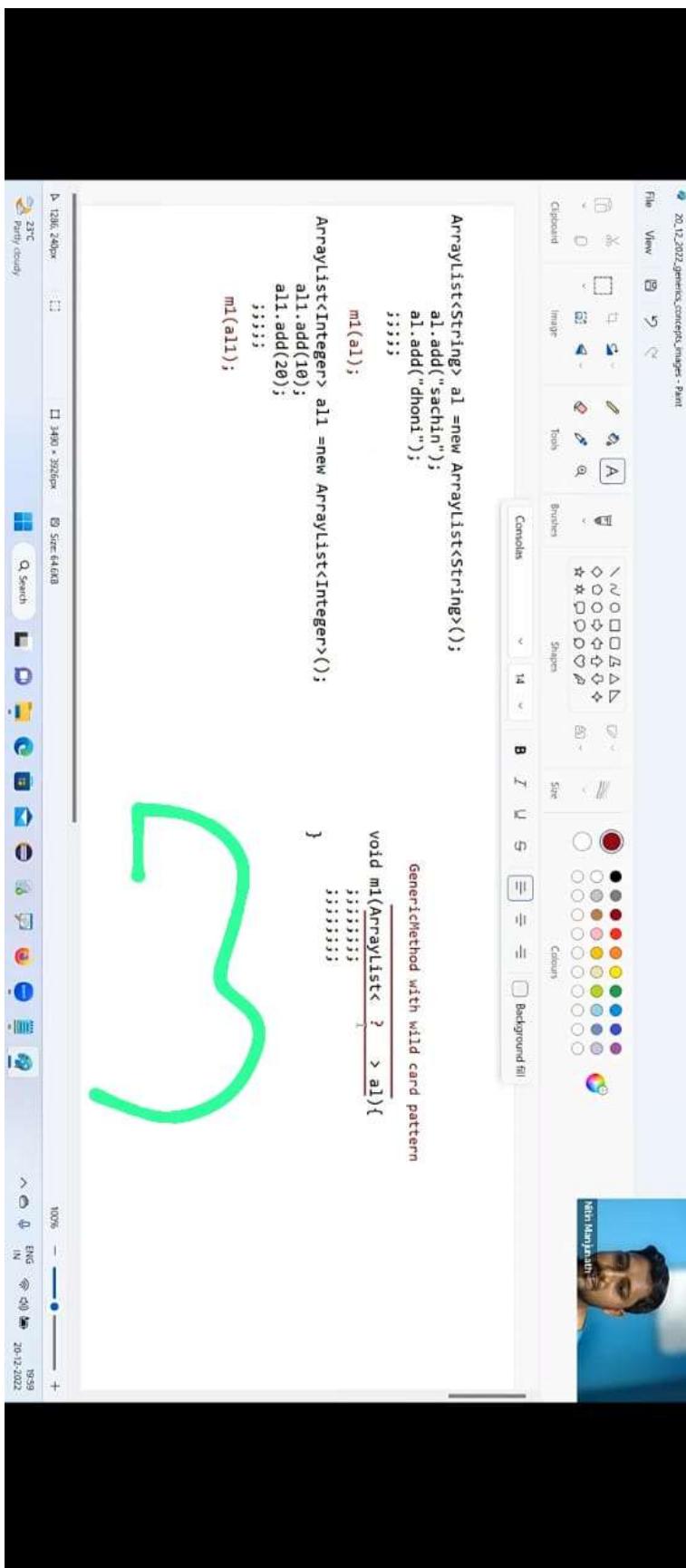
class Test<T extends Number&Runnable&Comparable> {} //valid
Number -> class
Runnable-> interface
Comparable -> interface

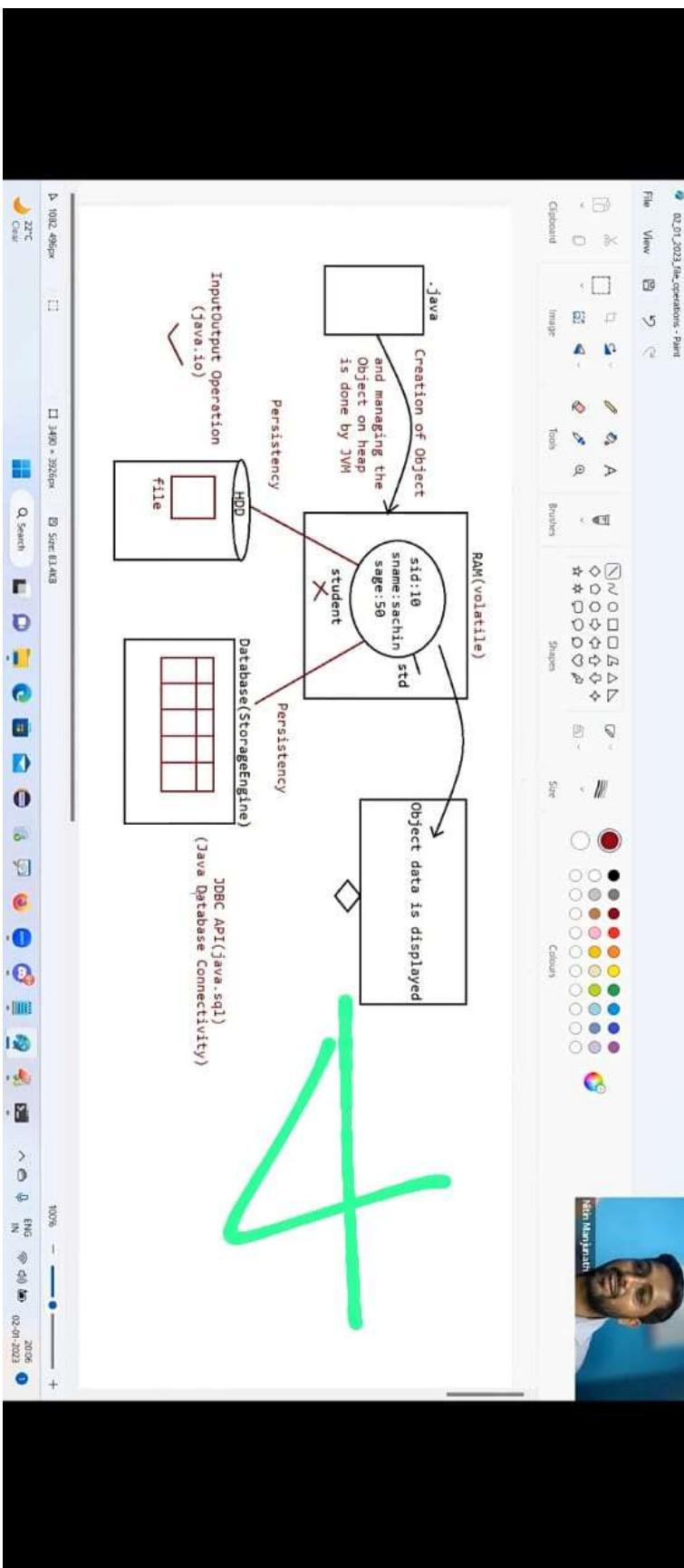
class Test<T extends Number&String> {} //invalid
we can't extends more than one class at a time.

class Test<T extends Runnable&Comparable> {} //valid
Runnable-> interface
Comparable -> interface

class Test<T extends Runnable&Number> {}//invalid
Runnable-> interface
Number -> class
rule: first inherit and then implement
```







What is persistency?
It is a mechanism of storing the data permanently on to the file.
In java persistency can be achieved through an API's available inside package called "java.io".

Input and Output Operations
=====

Agenda:

1. File
2. FileWriter
3. FileReader
4. BufferedWriter
5. BufferedReader

CSQ

02.01.2013 file-operations_class-notes - Notepad

File Edit View

Nitin Manjrekar

Unread 22:52

Q. Share Clear

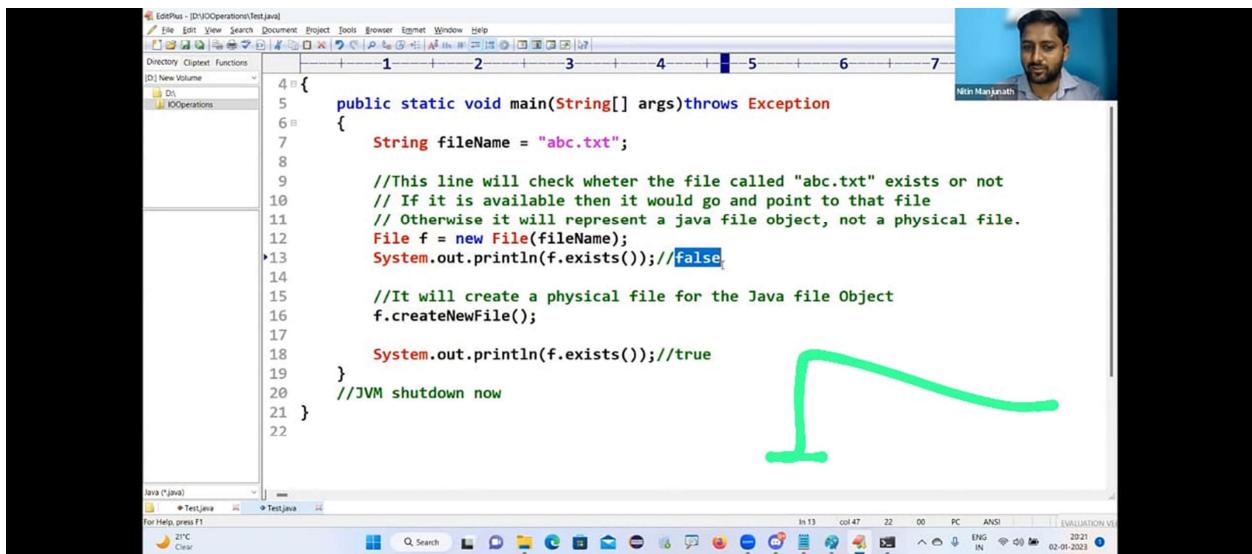
Received from TeenPatti bank Today, 02:52 PM + ₹9500 Received 3 s

paytm

D:\IOOperations>javap java.io.File

Compiled from "File.java"

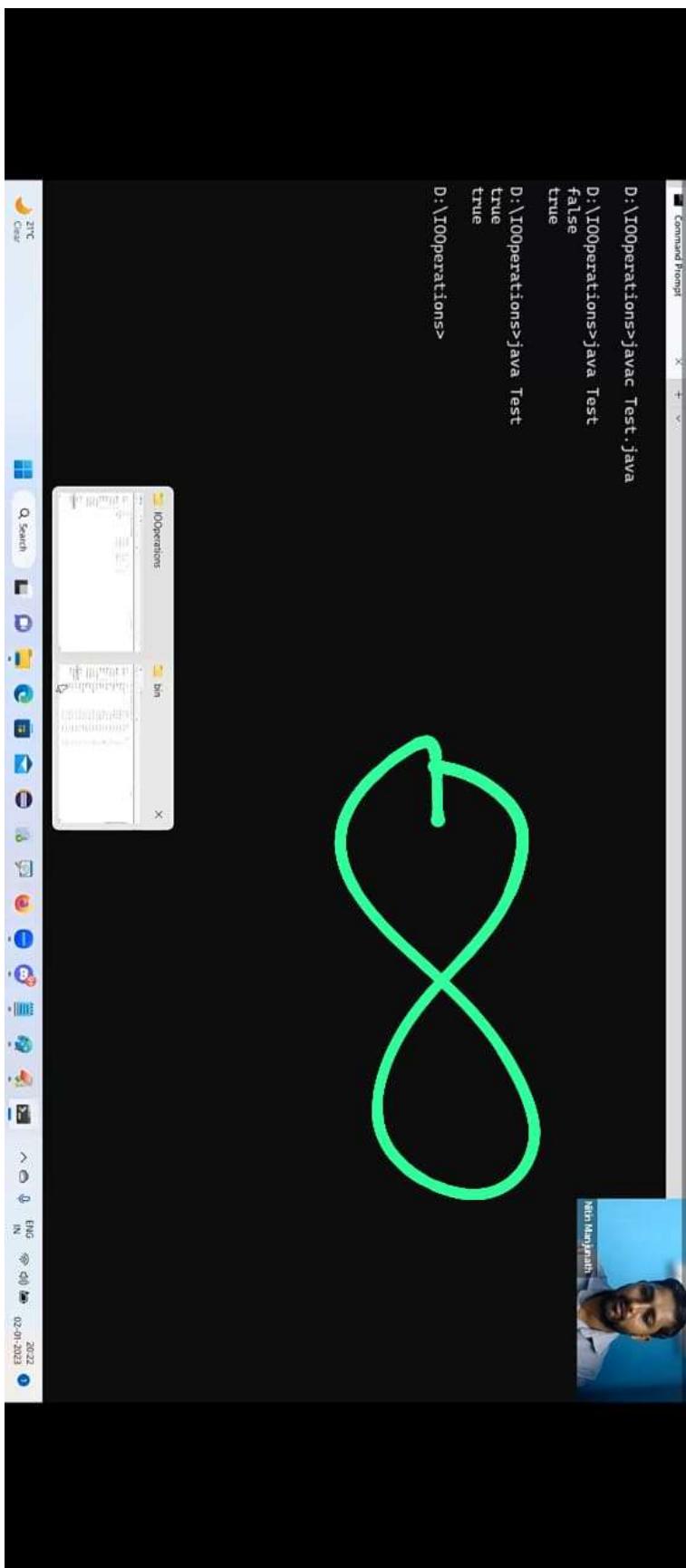
```
public class java.io.File implements java.io.Serializable, java.lang.Comparable<java.io.File> {
    public static final char separatorChar;
    public static final java.lang.String separator;
    public static final char pathSeparatorChar;
    public static final java.lang.String pathSeparator;
    static final boolean $assertionsDisabled;
    final boolean isInvalid();
    int getPrefixLength();
    public java.lang.String getName();
    public java.io.File(java.lang.String, java.lang.String);
    public java.io.File(java.io.File, java.lang.String);
    public java.io.File(java.net.URI);
    public java.lang.String getParent();
    public java.io.File getParentFile();
    public java.lang.String getPath();
    public boolean isAbsolute();
    public java.lang.String getAbsolutePath();
    public java.io.File getAbsoluteFile();
    public java.lang.String getCanonicalPath() throws java.io.IOException;
    public java.io.File getCanonicalFile() throws java.io.IOException;
    public java.net.URL toURL() throws java.net.MalformedURLException;
    public java.net.URI toURI();
    public boolean canRead();
    public boolean canWrite();
    public boolean exists();
    public boolean isDirectory();
    public boolean isFile();
    public boolean isHidden();
```



A screenshot of a Java code editor window titled "EditPlus - [D:\IOOperations\TestJava].java". The code is as follows:

```
4 {
5     public static void main(String[] args) throws Exception
6     {
7         String fileName = "abc.txt";
8
9         //This line will check whether the file called "abc.txt" exists or not
10        // If it is available then it would go and point to that file
11        // Otherwise it will represent a java file object, not a physical file.
12        File f = new File(fileName);
13        System.out.println(f.exists());//false
14
15        //It will create a physical file for the Java file Object
16        f.createNewFile();
17
18        System.out.println(f.exists());//true
19
20    }
21
22 }
```

The code demonstrates creating a file named "abc.txt" and then checking if it exists. It prints "false" because the file does not exist yet, and then prints "true" after the file is created.



02-01-2023 file_operations_class_notes - Notepad

File Edit View

5. BufferedReader

File:

```
File f=new File("abc.txt");
```

This line 1st checks whether abc.txt file is already available (or) not if it is already available then "f" simply refers that file. If it is not already available then it won't create any physical file just creates a java File object represents name of the file.

Example:

```
import java.io.*;
```

```
class FileDemo{
```

```
    public static void main(String[] args)throws IOException{
```

```
        File f=new File("cricket.txt");
```

```
        System.out.println(f.exists());//false
```

```
        f.createNewFile();
```

```
        System.out.println(f.exists());//true
```

Clear



Q. Search



Help



File



Java



Help



File



Java



Help



File



Java



Help



File



Java



Help

File

Java

Help

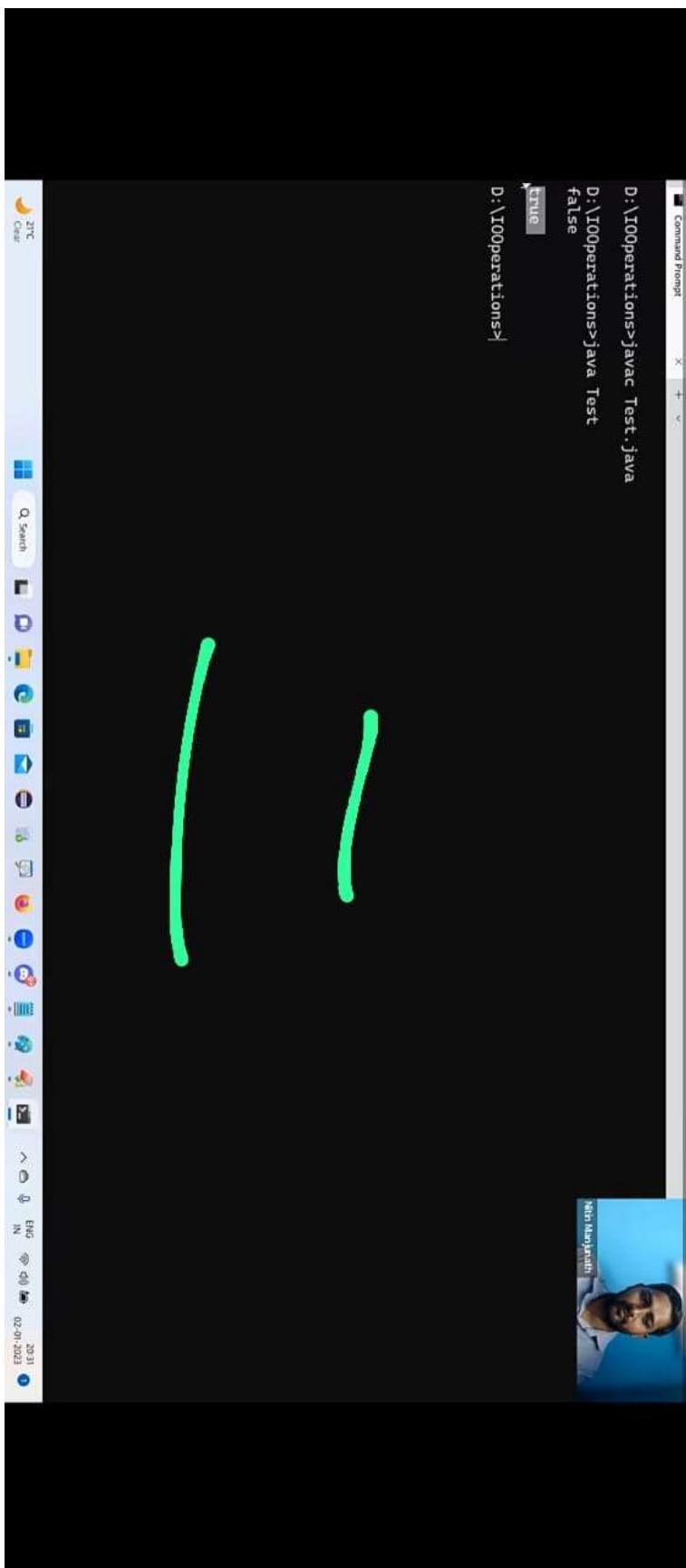
File

Java</

A screenshot of a Java code editor window titled "EditPlus - [IOOperationsTest.java]". The code is as follows:

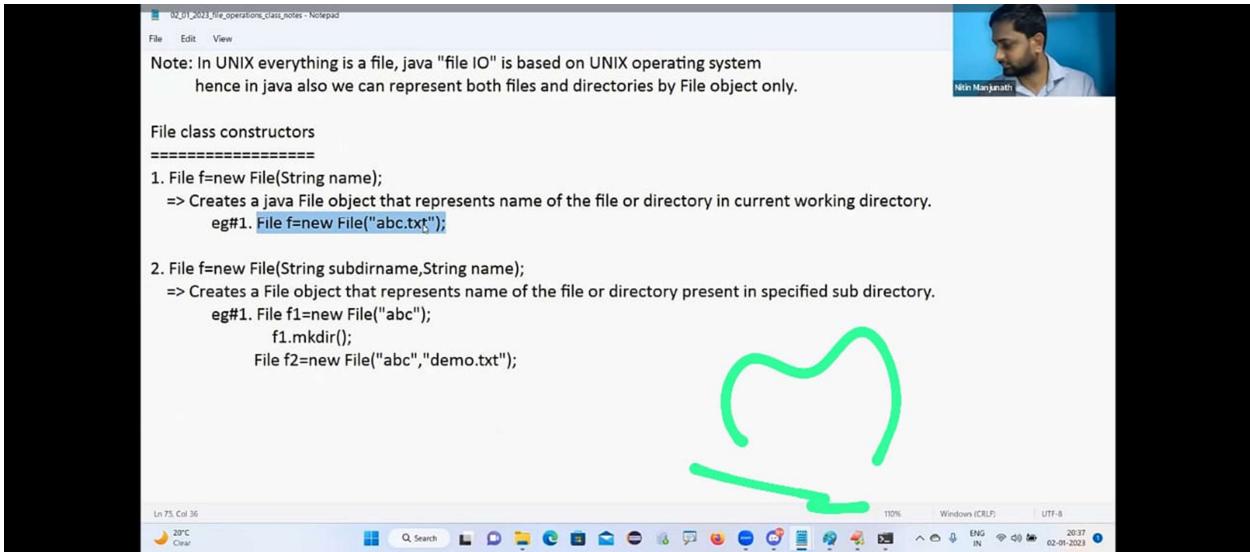
```
1 import java.io.*;
2
3 //java.io api classes are built using the standards of Unix os.
4 // In linux/unix os there is no difference b/w file and directory.
5 class Test
6 {
7     public static void main(String[] args) throws Exception
8     {
9         String directoryName = "IPLteams";
10
11        //it treats as a directory,for JVM no difference b/w directory and filename
12        File f= new File(directoryName);
13        System.out.println(f.exists());//false
14
15        f.mkdir();
16
17        System.out.println();
18        System.out.println(f.exists());
19    }
20    //JVM shutdown now
21 }
```

The code demonstrates that in Unix-like operating systems, there is no difference between files and directories when using Java's `java.io` API. A green oval and a green arrow point to the line `System.out.println(f.exists());` at line 13.





```
02-01-2023 file-operations class notes - Notepad  
File Edit View  
File f=new File("PLTeams");  
System.out.println(f.exists());//false  
  
f.mkdir();//Creates a new directory  
System.out.println(f.exists());//true  
}  
}  
1st run  
=====  
false  
true  
  
2nd run  
=====  
true  
true  
  
Note: In UNIX everything is a file, java "file IO" is based on UNIX operating system  
hence in java also we can represent both files and directories by File object only.
```



14



A screenshot of a Java code editor window titled "EditPlus - [D:\IOOperations\TestJava].java". The code demonstrates Java's File API for creating directories and files. A green annotation highlights the line "f.mkdir();". The code is as follows:

```
3 //java.io api classes are built using the standards of Unix os.
4 // In linux/unix os there is no difference b/w file and directory.
5 class Test
6 {
7     public static void main(String[] args) throws Exception
8     {
9         File f = new File("iNeuron");
10        f.mkdir();
11        System.out.println("Is f pointing to a directory :: "+f.isDirectory());
12
13        File f1 = new File(f,"abc.txt");
14        f1.createNewFile();
15        System.out.println("Is f1 pointing to a file :: "+f1.isFile());
16
17    }
18    //JVM shutdown now
19
20
21 }
```

The status bar at the bottom shows the following information: In 15 col 70 Z1 29 PC ANSI 20:47 02-09-2023. There is also a green hand-drawn annotation on the right side of the code area.

Editor - Test.java

File Edit View Search Document Project Tools Browser Ethernet Window Help

Directory Object Functions

[D] New Volume D:\ D:\Temp

```
1 import java.io.*;  
2  
3 // java.io api classes are built using the standards of Unix os.  
4 // In Linux/unix os there is no difference b/w file and directory.  
5 class Test  
6 {  
7     public static void main(String[] args) throws Exception  
8     {  
9         String location = "c:\\\\pwskills";  
10        File f =new File(location);  
11        f.mkdir();  
12  
13        File f1 =new File(f,"java.txt");  
14        f1.createNewFile();  
15  
16    }  
17  
18 }  
19  
//JVM shutdown now
```

Java (pws)

* Test.java

For Help press F1

20C Clear

Q. Search

In 3 col 4 19 6a PC ANSI | 100% (100%)

^ ⌂ ⌂ ENG ⌂ ⌂ IN 02-01-2023 2053 0



Editor - (D:\IOoperations\test.java)

File Edit View Search Document Project Tools Browser Ethernet Window Help

Directory Object Functions

[D] New Volume

D:\ IOoperations

```
1 import java.io.*;
```

2

3 // java.io api classes are built using the standards of Unix os.

4 // In linux/unix os there is no difference b/w file and directory.

```
5 class Test
```

6 {

```
7     public static void main(String[] args) throws Exception
```

8 {

```
9     public long length(); //no of characters present in file
```

```
10    public boolean delete(); //it deletes file or directory based on File Object
```

```
11    public String[] list(); //it returns the names of all files and directories under
```

```
12 }
```

```
13 //JVM shutdown now
```

```
14 }
```

```
15
```

```
16
```

Java (*.java)

* Test.java

For Help press F1

20C Clear

Q. Search

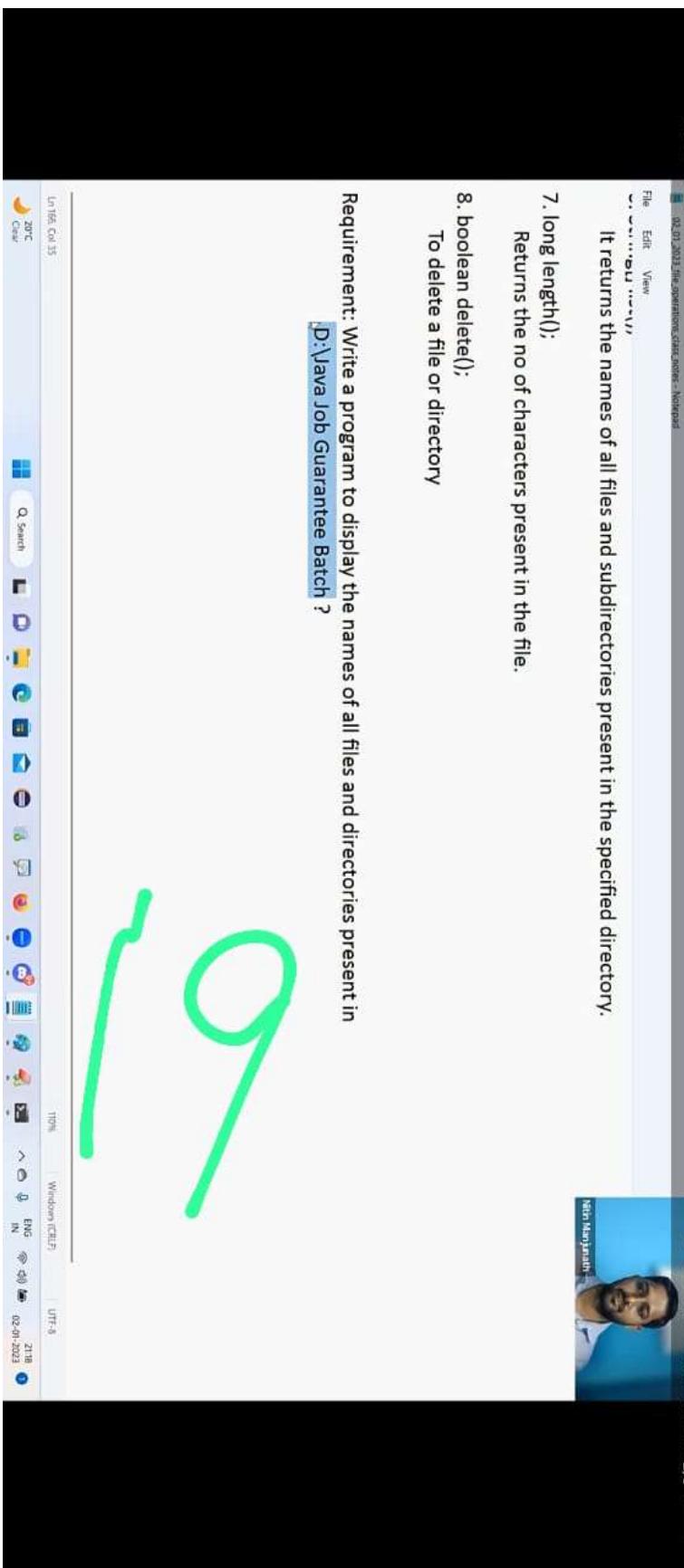
In 15 col 1 16 00 PC ANSI Evaluation

^ & ENG IN 02-01-2023 20:59



18

```
Editor - (D:\Operations\Java\)
File Edit View Search Document Project Tools Browser Ethernet Window Help
Directory Object Functions
[D] New Volume
D:\ Operations
1 3 4 5 6 7 8 9 0
2
3 ilt using the standards of Unix os.
4 no difference b/w file and directory.
5
6
7 tring[] args) throws Exception
8
9 //no of characters present in file
10 () ;//it deletes file or directory based on File Object
11 );//it returns the names of all files and directories under specified directory
12
13
14
15
16
```



```
Editor - (D:\Temp\TemporaryFile.java)
File Edit View Search Document Project Icons Browser Ethernet Window Help
Directory Object Functions
[D:\] New Volume
D:\ D:\Temp\TemporaryFile.java
10 */
11 class Test
12 {
13     public static void main(String[] args) throws Exception
14     {
15         int count = 0;
16
17         String location = "D:\\java Job Guarantee Batch";
18         File f= new File(location);
19
20         String[] names = f.list();
21
22         for(String name : names){
23             count++;
24             System.out.println(name);
25         }
26         System.out.println("Total no of files is :: "+count);
27
28
29
30 } //JVM shutdown now
```

Java (*.java)

* Test.java

For Help press F1

20C

Clear

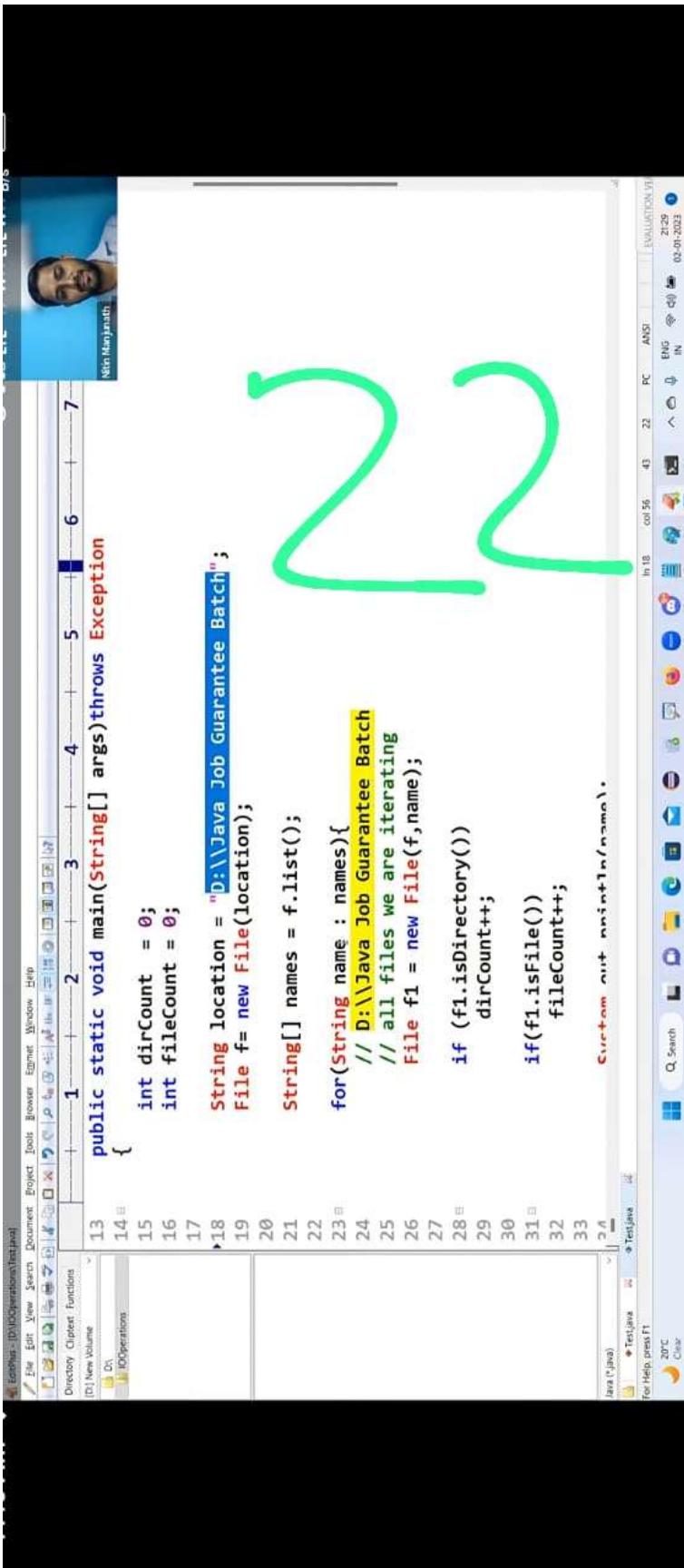
Q. Search

In 17 col 32 32 48 PC ANSI ↑ ↓ PAGE

ENG ⇔ CHS ↵ 02-01-2021

```
D:\IOOperations>javac Test.java
D:\IOOperations>java Test
01-PrimitiveValue-Injection
02-One-One-Association-mapping
02_01_2023_file_operations.png
02_01_2023_file_operations_class_notes.txt
03-One-MANY-Association-mapping
04-MANY-ONE-Association-mapping
05-MANY-MANY-Association-mapping
05_12_2022_threads_classnotes.txt
06_10_2022_QA_Discussion_notes.txt
07_10_2022_QA_Discussion_notes_job_guarantee.txt
07_11_2022_StringBuilder_StringBuffer_image.png
07_12_2022_multithreading_class_notes.txt
07_12_2022_multithreading_images.png
07_12_2022_snippets_session_images.png
07_12_2022_snippets_session_notes.txt
08_10_2022_QA_Discussion_notes_job_guarantee.txt
08_12_2022_deadlock_images.png
08_12_2022_multithreading_notes.txt
09_12_2022_snippets_session_images.png
09_12_2022_snippets_session_notes.txt
10_10_2022_github_notes.txt
10_11_2022_Snippets_images.png
10_11_2022_Snippets_session_classnotes.txt
11_10_2022_github_notes.txt
11_11_2022_Snippets_images.png
11_11_2022_Snippets_session_classnotes.txt
12_10_2022_class_notes.txt
12_10_2022_QA_Discussion.txt
```

21



```
File Edit View Search Document Project Tools Browser Ethernet Window Help  
Directory Clipboard Functions [D:] New Volume  
Do KOperations  
13 public static void main(String[] args) throws Exception  
14 {  
15     int dirCount = 0;  
16     int fileCount = 0;  
17  
18     String location = "D:\\Java Job Guarantee Batch";  
19     File f= new File(location);  
20  
21     String[] names = f.list();  
22  
23     for(String name : names){  
24         // D:\\Java Job Guarantee Batch  
25         // all files we are iterating  
26         File f1 = new File(f, name);  
27  
28         if (f1.isDirectory())  
29             dirCount++;  
30  
31         if(f1.isFile())  
32             fileCount++;  
33  
34     System.out.println("name\\");  
35 }  
Java(TM)  
TestJava  
For Help press F1  
20°C Clear  
In 18 °C 56 43 22 PC ANSI EVALUATION 04/01/2023 21:29 ENG ⇧ IN 02-01-2023
```

Editor - D:\Guarantee\Batch

File Edit View Search Document Project Tools Browser Format Window Help

Directory Object Functions

[D] New Volume

D:\ Operations

1 2 3 4 5 6 7

```
19
20
21
22
23 for(String name : names){
24     // D:\\Java Job Guarantee Batch
25     // all files we are iterating
26     File f1 = new File(f,name);
27
28 if (f1.isDirectory()){
29     dirCount++;
30
31 if(f1.isFile())
32     fileCount++;
33
34 System.out.println(name);
35
36 System.out.println("Total no of files is :: "+fileCount);
37 System.out.println("Total no of Directory is :: "+dirCount);
38
39 }
40
41 // JVM shutdown now!
```

Java (*.java)

* TestJava

For Help press F1

20C

Clear

Q. Search

ln 31 col 28 43 00 PC ANSI 21:29 02-01-2021

Nitin Manjrekar

2

File Edit View Document Project Tools Window Help

Director/Client Functions

[D] New Volume [D] Operations

15 int dirCount = 0;

16 int jpgFileCount = 0;

17 int txtFileCount = 0;

18

19 String location = "D:\\Java Job Guarantee Batch";

20 File f = new File(location);

21

22 String[] names = f.list();

23

24 for(String name : names){

25 // D:\\Java Job Guarantee Batch

26 // all files we are iterating

27 File f1 = new File(f, name);

28

29 if (f1.isDirectory()) {

30 dirCount++;

31

32 if(f1.isFile()){

33

34 if (name.endsWith(".jpg")){

35 jpgFileCount++;

45 // name ends with ".txt" + v+ "

46

47

24

Nitin Raghavath

1:49 AM

File Edit View Search Document Tool Flower Ermine Window Help

Directory Object Functions

[D] New Volume

[D] Operations

12 {

13 public static void main(String[] args) throws Exception

14 {

15 int dirCount = 0;

16 int jpgFileCount = 0;

17 int txtFileCount = 0;

18 int zipFileCount = 0;

19

20 String location = "D:\\Java Job Guarantee Batch";

21 File f= new File(location);

22

23 String[] names = f.list();

24

25 for(String name : names){

26 // D:\\Java Job Guarantee Batch

27 // all files we are iterating

28 File f1 = new File(f,name);

29

30 if (f1.isDirectory())

31 dirCount++;

32

33 System.out.println(f1.getName());

34 }

Java (Java)

* TestJava

* TestJava

For Help press F1

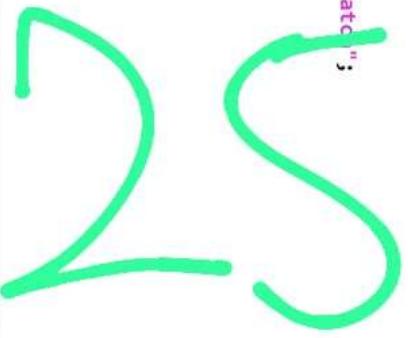
20C

Clear

Q. Search

In 18 col 30 55 00 PC ANSI

EN IN 02-01-2021 21:32



```
Editor - D:\JavaOperations\test.java
```

```
File Edit View Search Document Project Icons Browser Ethernet Window Help
D:\ New Volume Directory Object Functions
D:\ Operations
1 2 3 4 5 6 7
2.9
30 if (f1.isDirectory())
31     dirCount++;
32
33 if(f1.isFile()){
34
35     if (name.endsWith(".jpg"))
36         jpgFileCount++;
37     if(name.endsWith(".txt"))
38         txtFileCount++;
39     if(name.endsWith(".zip"))
40         zipFileCount++;
41
42
43
44 }
45
46 System.out.println("Total no of JPGfiles is :: "+jpgFileCount);
47 System.out.println("Total no of txtfiles is :: "+txtFileCount);
48 System.out.println("Total no of Zipfiles is :: "+zipFileCount);
49 System.out.println("Total no of Directory is :: "+dirCount);

System.out.println(name);

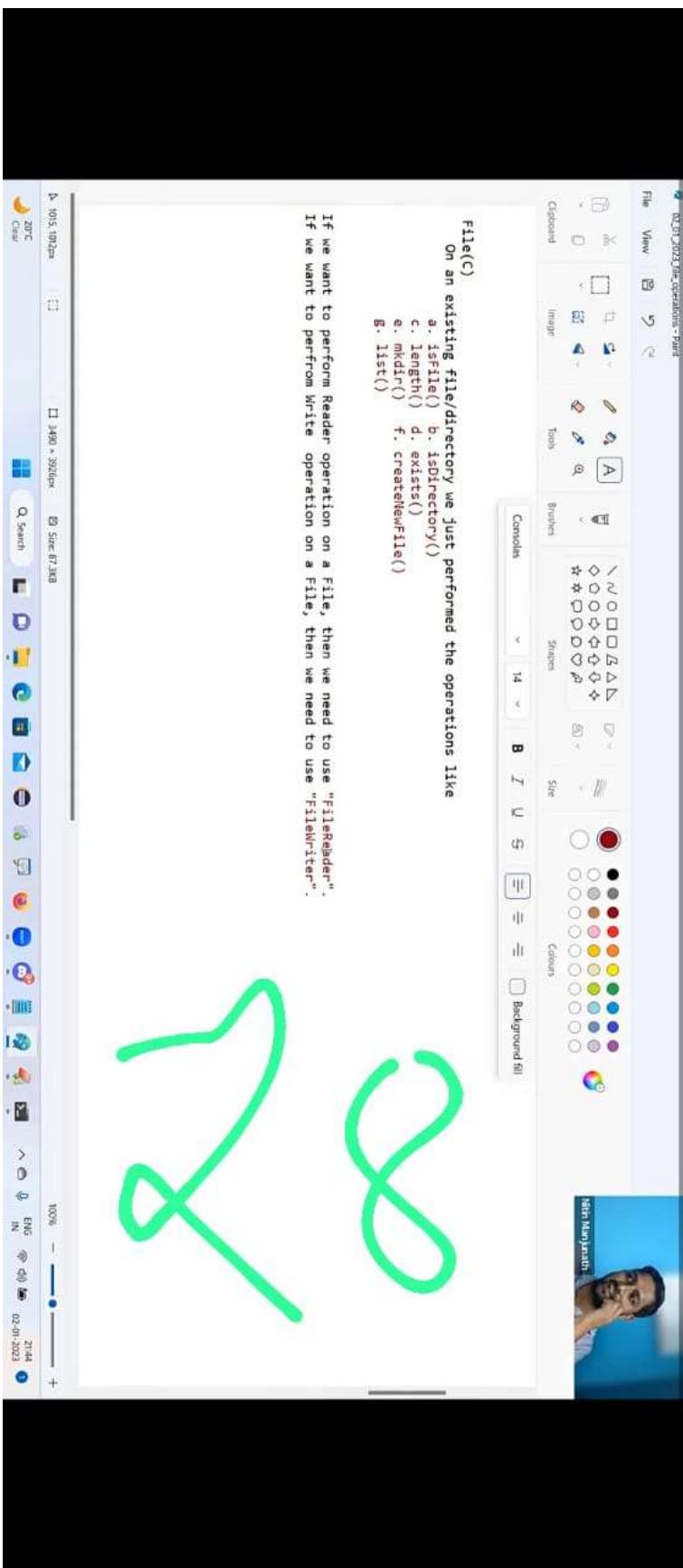
}
}

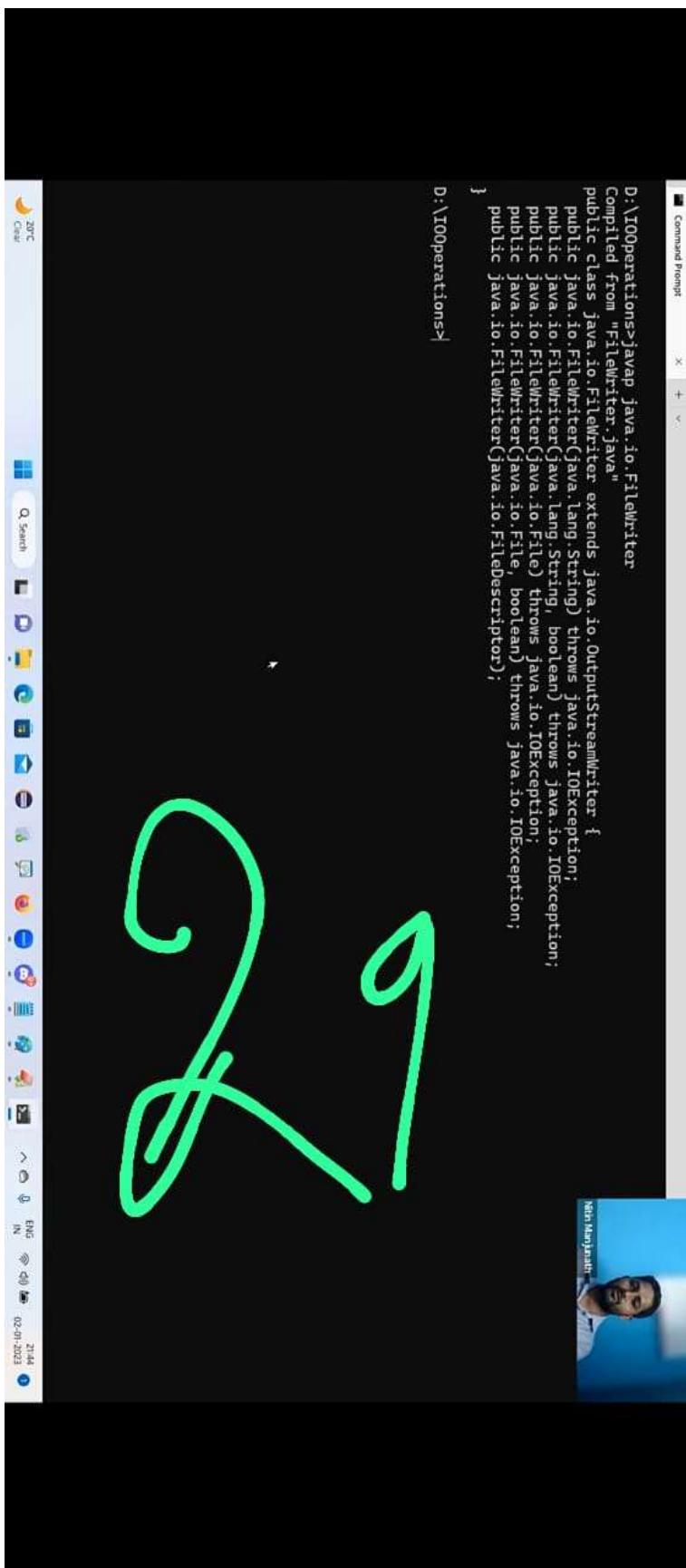
for Help press F1
Java (*.java)
TestJava
TestJava
20C
Clear
Q. Search
In 41 col 13 55 00 PC ANSI 100% DATE
ENG ⇧ ⌂ 21:32 02-01-2023
```

26

```
Command Prompt x + v  
6-12-2022_multithreading_class_notes.txt  
6-12-2022_multithreading_images.png  
6-12-2022_snippets_session_notes.txt  
7-11-2022_Snippets_session_classnotes.txt  
7-11-2022_StringBuilder_StringBuffer_session_classnotes.txt  
7-11-2022_String_snippets_session.png  
8-11-2022_snippets_image.png  
8-11-2022_snippets_session_classnotes.txt  
9-11-2022_QA_discussion_session_classnotes.txt  
9-11-2022_Snippets_images.png  
9-11-2022_Snippets_session_classnotes.txt  
code_snippets.png  
Datatypes.png  
Git_Architecture.png  
github_introduction_with_eclipse.txt  
Github_introduction.png  
GitServer_Architecture.png  
indentification_of_reservewords.png  
JVM_Architecture.png  
Local_Version_Control_System.png  
Object_creation.png  
personal_access_token.txt  
Reservewordchart.png  
snippets_solving_question.png  
Source_Code.zip  
SVN_Vs_Git.png  
Total no of JPGfiles is :: 0  
Total no of txtfiles is :: 86  
Total no of Zipfiles is :: 1  
Total no of Directory is :: 5  
D:\IOoperations>
```

A large, handwritten number '27' is drawn in green ink over the terminal window, with the top of the '2' pointing towards the top-left corner of the window.





D:\IOoperations>javap java.io.FileWriter

Compiled from "FileWriter.java"

```
public class java.io.FileWriter extends java.io.OutputStreamWriter {
    public java.io.FileWriter(java.lang.String) throws java.io.IOException {
        public java.io.FileWriter(java.lang.String, boolean) throws java.io.IOException;
        public java.io.FileWriter(java.io.File, boolean) throws java.io.IOException;
        public java.io.FileWriter(java.io.FileDescriptor);
    }

D:\IOoperations>javap java.io.OutputStreamWriter
Compiled from "OutputStreamWriter.java"
public class java.io.OutputStreamWriter extends java.io.Writer {
    public java.io.OutputStreamWriter(java.io.OutputStream, java.lang.String) throws java.io.UnsupportedEncodingException;
    public java.io.OutputStreamWriter(java.io.OutputStream);
    public java.io.OutputStreamWriter(java.io.OutputStream, java.nio.charset.Charset);
    public java.io.OutputStreamWriter(java.io.OutputStream, java.nio.charset.CharsetEncoder);
    public void flushBuffer() throws java.io.IOException;
    public void write(int) throws java.io.IOException;
    public void write(char[], int, int) throws java.io.IOException;
    public void write(java.lang.String, int, int) throws java.io.IOException;
    public void flush() throws java.io.IOException;
    public void close() throws java.io.IOException;
}
```

30

```
D:\IOoperations>javadoc java.io.Writer
Compiled from "Writer.java"
public abstract class java.io.Writer implements java.lang.Appendable,java.io.Closeable,java.io.Flushable {
    protected java.lang.Object lock;
    protected java.io.Writer();
    protected java.io.Writer(java.lang.Object);
    public void write(int) throws java.io.IOException;
    public void write(char[], int, int) throws java.io.IOException;
    public void flush() throws java.io.IOException;
    public void close() throws java.io.IOException;
}

D:\IOoperations>
```

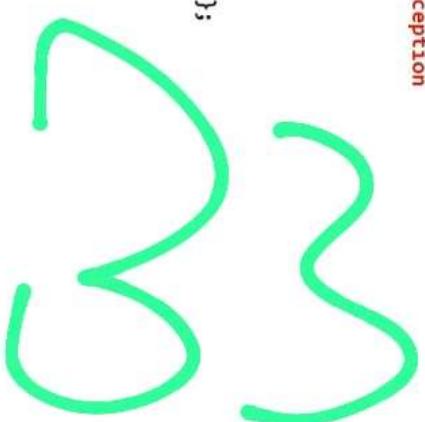


```
D:\IOoperations>javac Test.java
D:\IOoperations>java Test
Compiled from "FileWriter.java"
public class java.io.FileWriter extends java.io.OutputStreamWriter {
    public java.io.FileWriter(java.lang.String) throws java.io.IOException;
    public java.io.FileWriter(java.lang.String, boolean) throws java.io.IOException;
    public java.io.FileWriter(java.io.File) throws java.io.IOException;
    public java.io.FileWriter(java.io.File, boolean) throws java.io.IOException;
    public java.io.FileWriter(java.io.FileDescriptor);
}

D:\IOoperations>javap java.io.OutputStreamWriter
Compiled from "OutputStreamWriter.java"
public class java.io.OutputStreamWriter extends java.io.Writer {
    public java.io.OutputStreamWriter(java.io.OutputStream, java.lang.String) throws java.io.UnsupportedEncodingException;
    public java.io.OutputStreamWriter(java.io.OutputStream, java.nio.charset.Charset);
    public java.io.OutputStreamWriter(java.io.OutputStream, java.nio.charset.CharsetEncoder);
    public java.lang.String getEncoding();
    void flushBuffer() throws java.io.IOException;
    public void write(int) throws java.io.IOException;
    public void write(char[], int, int) throws java.io.IOException;
    public void write(java.lang.String, int, int) throws java.io.IOException;
    public void flush() throws java.io.IOException;
    public void close() throws java.io.IOException;
}

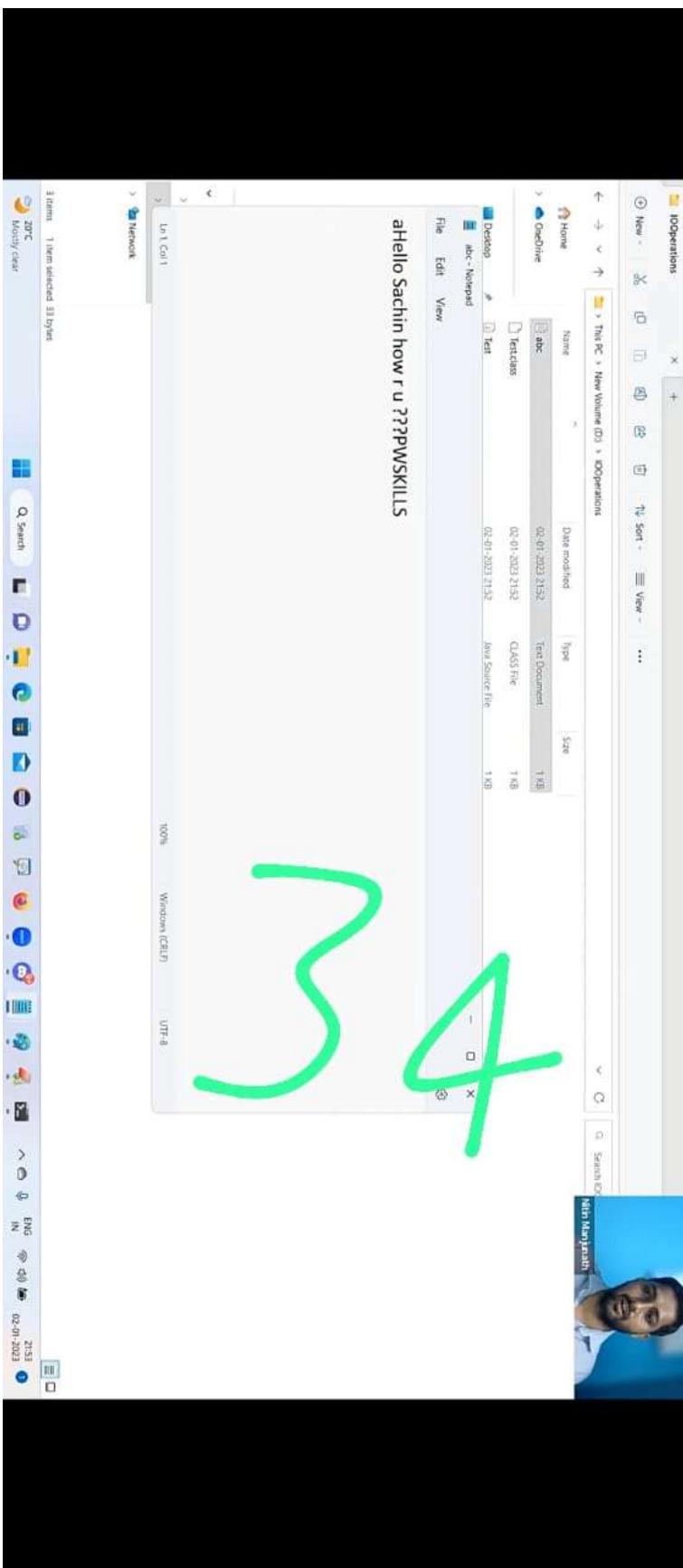
D:\IOoperations>
```



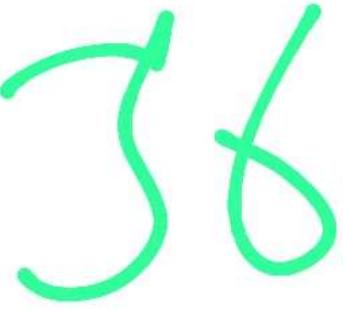


```
Editor - D:\Work\Java\HelloWorld
File Edit View Search Document Project Tools Browser Ethernet Window Help
Directory Object Functions
[D:\] New Volume
D:\ Operations
1 import java.io.*;
2
3
4 class Test
5 {
6     public static void main(String[] args) throws Exception
7     {
8         File f1 = new File("abc.txt");
9         FileWriter f2 = new FileWriter(f1);
10
11     //performing write operation on a file
12     f2.write(97);
13     f2.write("Hello Sachin how r u ???");
14     char[] ch={'P','W','S','K','I','L','L','S'};
15     f2.write(ch);
16
17     //closing the resource
18
19     f2.close();
20
21 } //JVM shutdown now
```

Java (Java) In 11 00:47 23 00 PC ANSI 100% (100%)
* TestJava 20C Mostly clear
For Help press F1



33



```
Editor - (D:\Operations\test.java)
File Edit View Search Document Project Tools Browser Ethernet Window Help
K ↵ F10 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
Directory Object Functions
[D:\] New Volume
D:\ D:\Operations
1 2 3 4 5 6 7
File f1 = new File("abc.txt");
FileWriter f2 = new FileWriter(f1,true);
//performing write operation on a file
f2.write("\n");
f2.write(g7);//a
f2.write("\n");
f2.write("Hello Sachin how r u ???");
f2.write("\n");
char[] ch={'p','W','S','K','I','L','L','S'};
f2.write(ch);
f2.close();
//making the data to write to the File
f2.flush();
f2;
System.out.println("open abc.txt to see the output");
}
//JVM shutdown now
}

Java (Java) 10 In 8 col 19 29 20 PC ANSI 100% Resolution
* TestJava TestJava 20C 20C
For Help press F1 20C
Mon 01:45 02-01-2021
```

Editor - D:\IOoperations\test.java

File Edit View Search Document Project Tools Browser Ethernet Window Help

Directory Object Functions

[D] New Volume

D:\ IOoperations

3 class Test

4 {

5 public static void main(String[] args) throws Exception

6 {

7 File f1 = new File("abc.txt");

8 //performing write operation on a file

9 FileWriter f2 = new FileWriter(f1,true);

10 f2.write("\n");

11 f2.write("\n");

12 f2.write("Hello Sachin how r u ???");

13 f2.write("\n");

14 f2.write("Hello Sachin how r u ???");

15 f2.write("\n");

16 char[] ch ={'P','W','S','K','I','L','L','S'};

17 f2.write(ch);

18

19 //making the data to write to the File

20

21 f2.flush();

22

23 } ~~~~~.

Java (javac) 1.7 TestJava TestJava

For Help press F1 20C Mostly clear

Q. Search

In 8 col 47 29 29 PC ANSI Evaluation Date: 02-01-2021 22:14 ENG ⇧ ⇩ ⇨ ⇪ IN

38

```
D:\IOoperations>javap java.io.FileReader
Compiled from "FileReader.java"
public class java.io.FileReader extends java.io.InputStreamReader {
    public java.io.FileReader(java.lang.String) throws java.io.FileNotFoundException;
    public java.io.FileReader(java.io.File) throws java.io.FileNotFoundException;
    public java.io.FileReader(java.io.FileDescriptor);
}

D:\IOoperations>javap java.io.InputStreamReader
Compiled from "InputStreamReader.java"
public class java.io.InputStreamReader extends java.io.Reader {
    public java.io.InputStreamReader(java.io.InputStream);
    public java.io.InputStreamReader(java.io.InputStream, java.lang.String) throws java.io.UnsupportedEncodingException;
    public java.io.InputStreamReader(java.io.InputStream, java.nio.charset.Charset);
    public java.io.InputStreamReader(java.io.InputStream, java.nio.charset.CharsetDecoder);
    public java.lang.String getEncoding();
    public int read() throws java.io.IOException;
    public int read(char[], int, int) throws java.io.IOException;
    public boolean ready() throws java.io.IOException;
    public void close() throws java.io.IOException;
}

D:\IOoperations>
```

Command Prompt

```
Compiled from "InputStreamReader.java"
public class java.io.InputStreamReader extends java.io.Reader {
    public java.io.InputStreamReader(java.io.InputStream);
    public java.io.InputStreamReader(java.io.InputStream, java.lang.String) throws java.io.UnsupportedEncodingException;
    public java.io.InputStreamReader(java.io.InputStream, java.nio.charset.Charset);
    public java.io.InputStreamReader(java.io.InputStream, java.nio.charset.CharsetDecoder);
    public java.lang.String getEncoding();
    public int read() throws java.io.IOException;
    public int read(char[], int, int) throws java.io.IOException;
    public boolean ready() throws java.io.IOException;
    public void close() throws java.io.IOException;
}

Compiled from "Reader.java"
public abstract class java.io.Reader implements java.lang.Readable,java.io.Closeable {
    protected java.lang.Object lock;
    protected java.io.Reader();
    protected java.io.Reader(java.lang.Object);
    public int read(java.nio.CharBuffer) throws java.io.IOException;
    public int read() throws java.io.IOException;
    public int read(char[]) throws java.io.IOException;
    public abstract int read(char[], int, int) throws java.io.IOException;
    public long skip(long) throws java.io.IOException;
    public boolean ready();
    public boolean markSupported();
    public void mark(int) throws java.io.IOException;
    public void reset() throws java.io.IOException;
    public abstract void close() throws java.io.IOException;
}

D:\IOoperations>javap java.io.Reader
```

20C Mailbox Clear

Q. Search

^ < > ENG IN 02-01-2021 22:18

Editor - D:\DOC\operations\Java\

File Edit View Search Document Project Tools Browser Ethernet Window Help

Directory Object Functions

[D] New Volume

D:\ Operations

1 import java.io.*;

2

3 class Test

4 {

5 public static void main(String[] args) throws Exception

6 {

7 FileReader f1 =new FileReader("sachin.txt");

8 int i = f1.read();

9 while (i!= -1)

10 {

11 System.out.print((char)i+"====>");

12 i= f1.read();

13 System.out.println(i);

14 }

15 }

16 }

17 }

18 }

19 }

20 }

21 //JVM shutdown now

Java (jps) Java (jstat) TestJava TestJava

For Help press F1

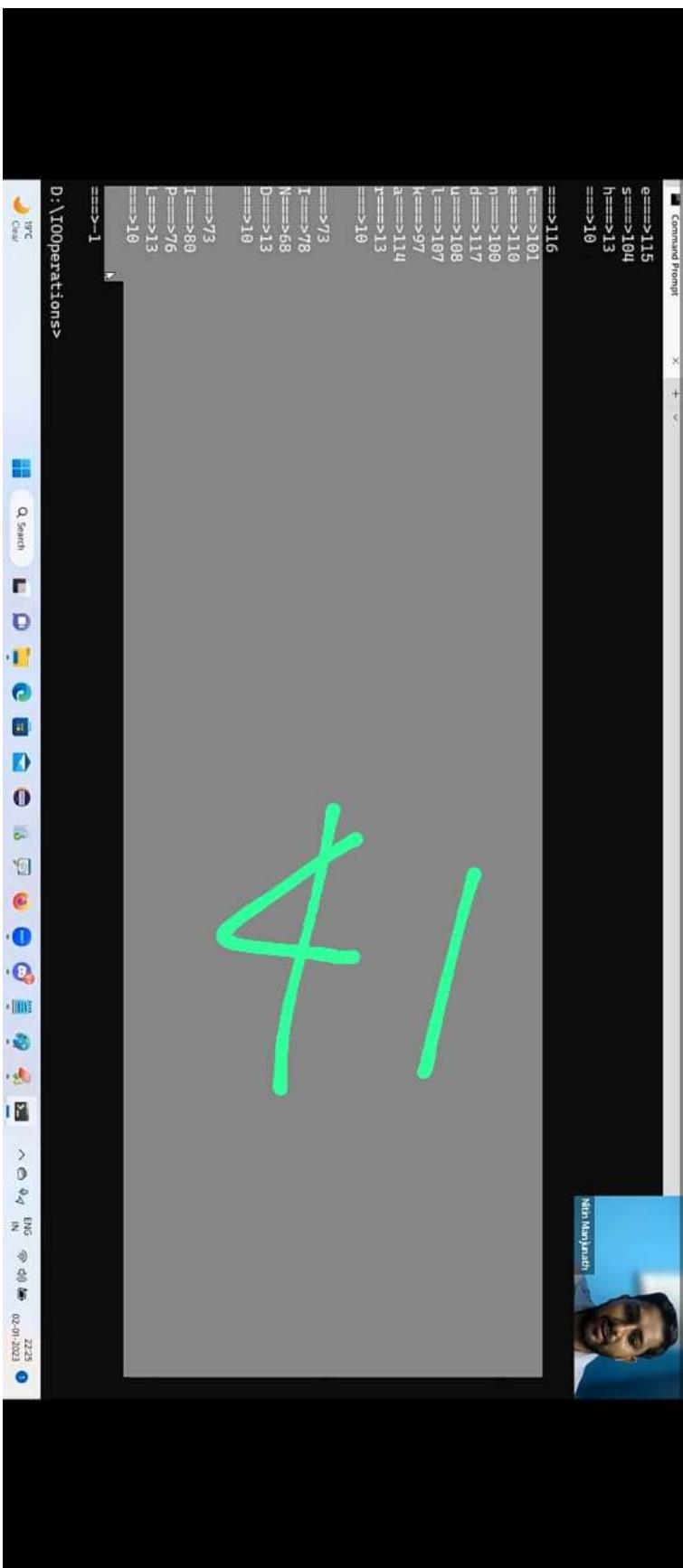
20°C Mostly clear

Q. Search

In 12 col 48 22 28 PC ANSI | EVALUATION DATE 02-01-2021 22:24

ENG → (d) 22:24

40



```
Editor - (FileOperations.java)
File Edit View Search Document Project Tools Browser Ethernet Window Help
Directory Object Functions
[D] New Volume
[DA]
IDOperations
1 import java.io.*;
2
3 class Test
4 {
5     public static void main(String[] args) throws Exception
6     {
7         FileReader f1 =new FileReader("sachin.txt");
8         int i = f1.read();
9
10        while (i!=1)
11        {
12            System.out.print(i+"====>");
13            System.out.println((char)i);
14            i= f1.read();
15
16        }
17    }
18 }
//JVM shutd 02-01-2023 file_operations.java
FileOperations - Notepad
FileOperationsSupports - Notepad
abc - Notepad
section - Notepad
Java (Java)
TelJava TelJava
For Help press F1
192PC
Clear
Q. Search
In 16 00:14 22 09 PC ANSI EDITION/DAVE
ENG ⇧ ⇩ 22:30 02-01-2023
```