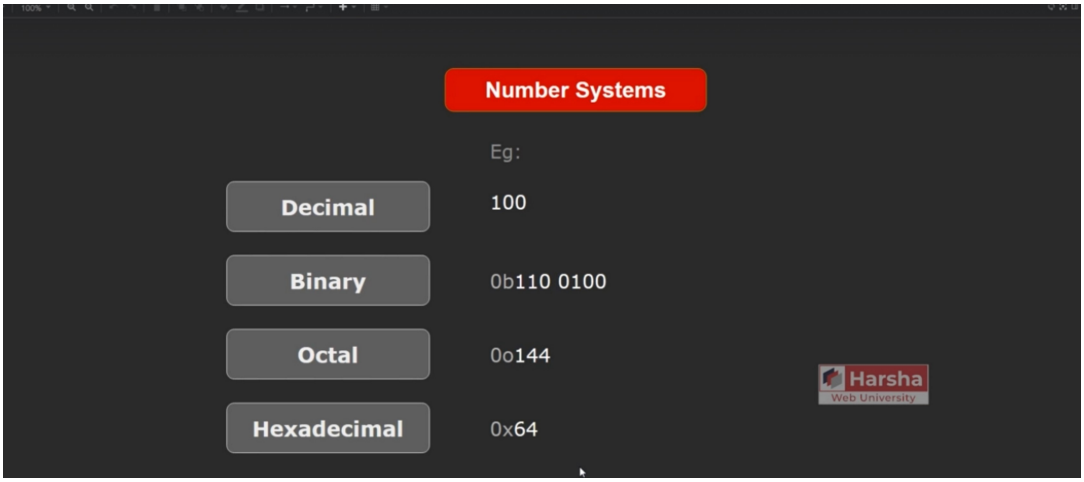
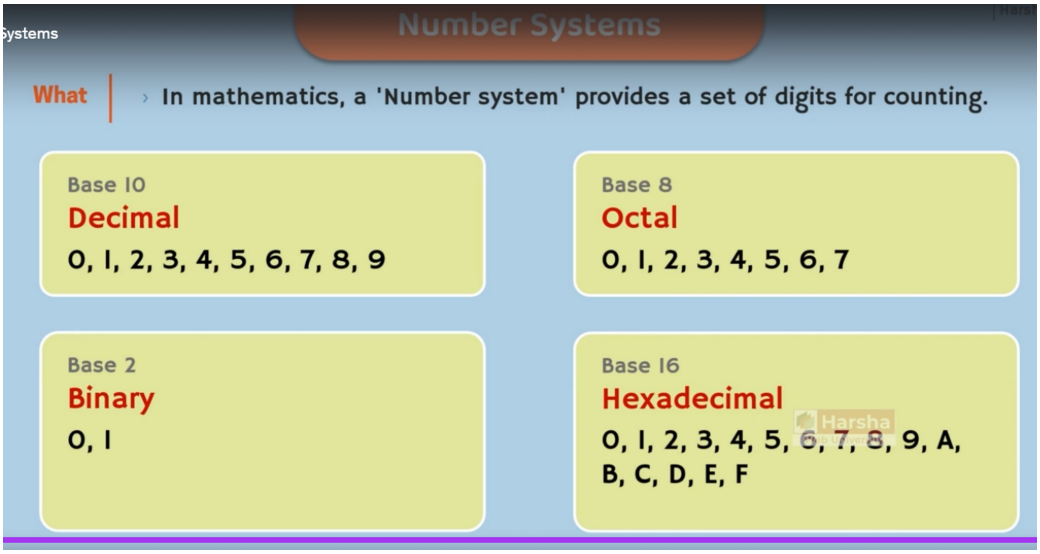
Section 27: IO, Serialization, Encoding

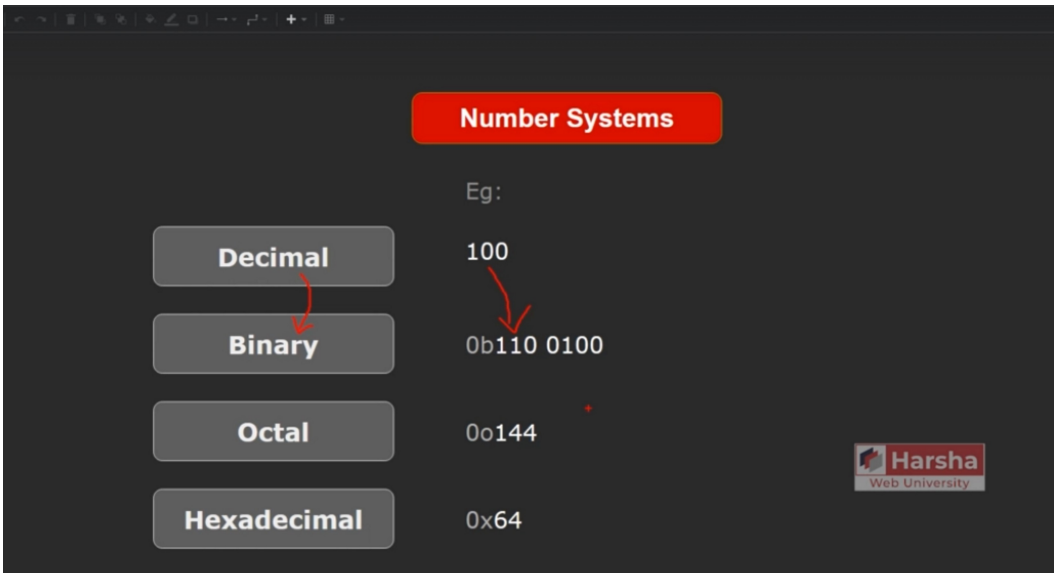
30 July 2024 16:45

Introduction to Number Systems





Binary Number System

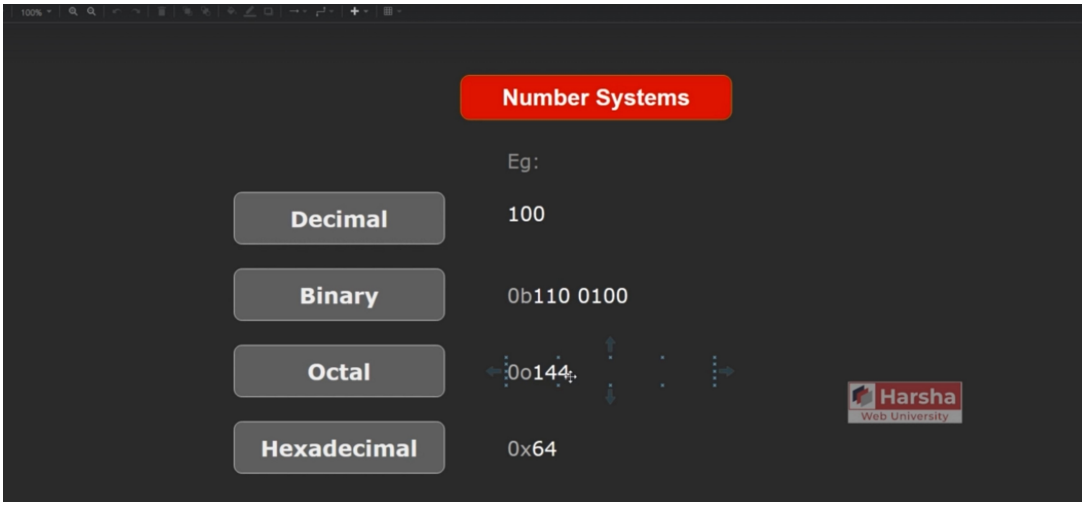


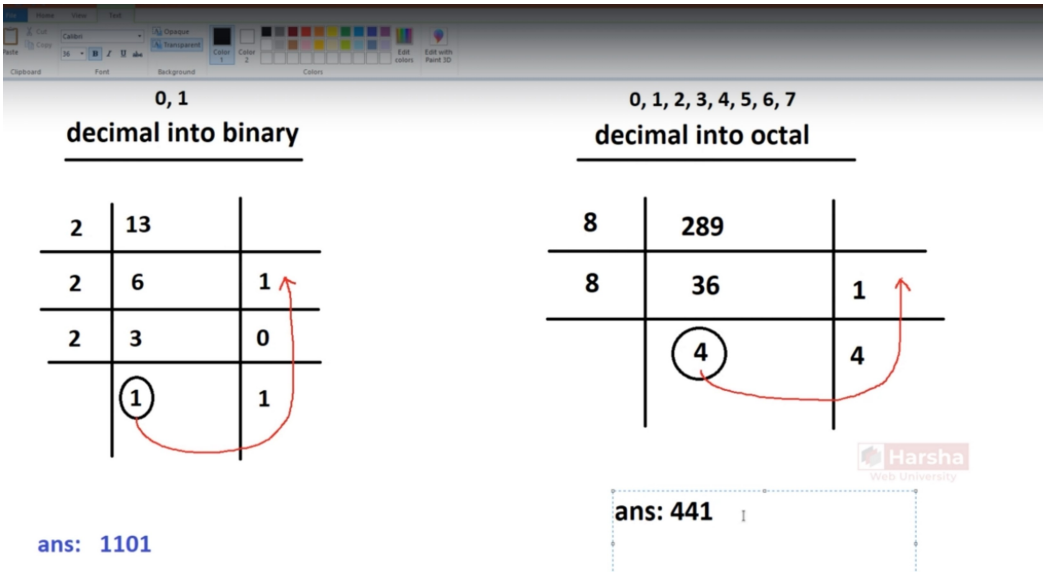
decimal into binary

2	13	
2	6	1
2	3	0
	1	1

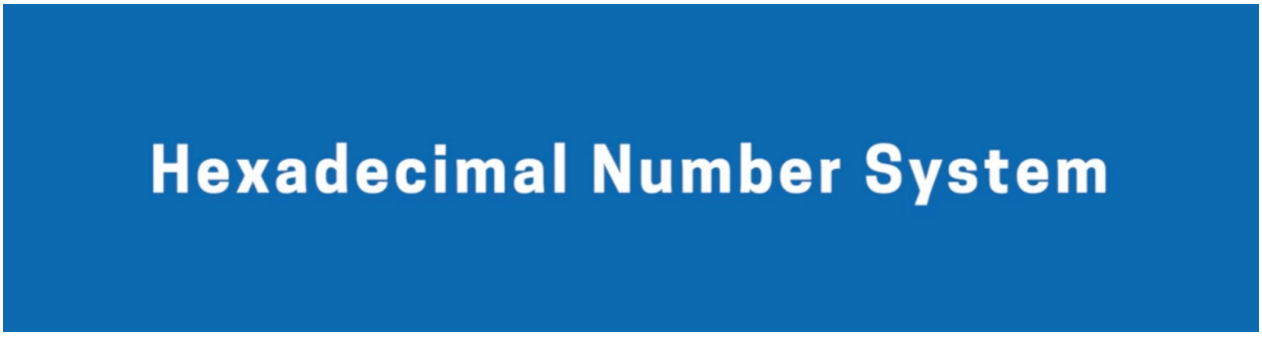
ans: 1101

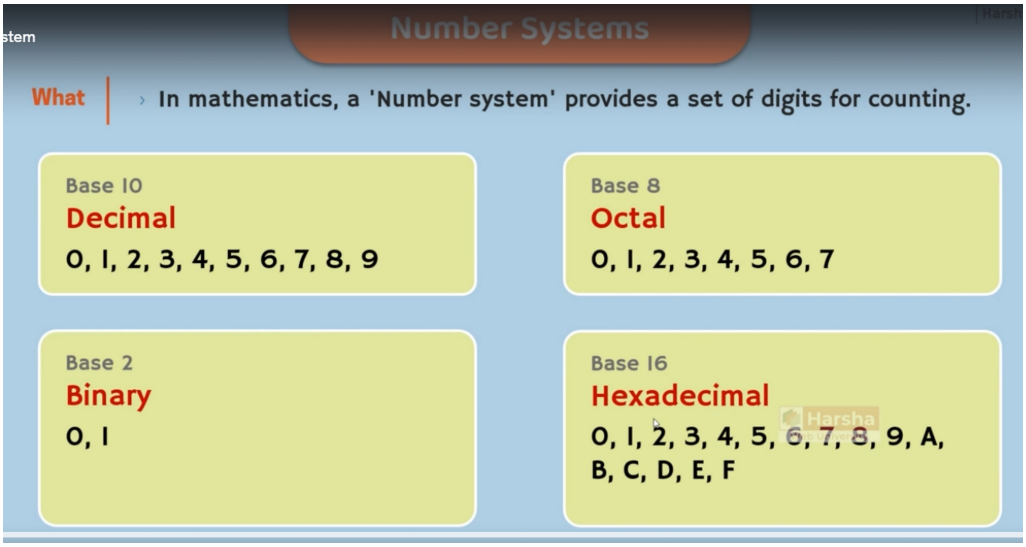
Octal Number System



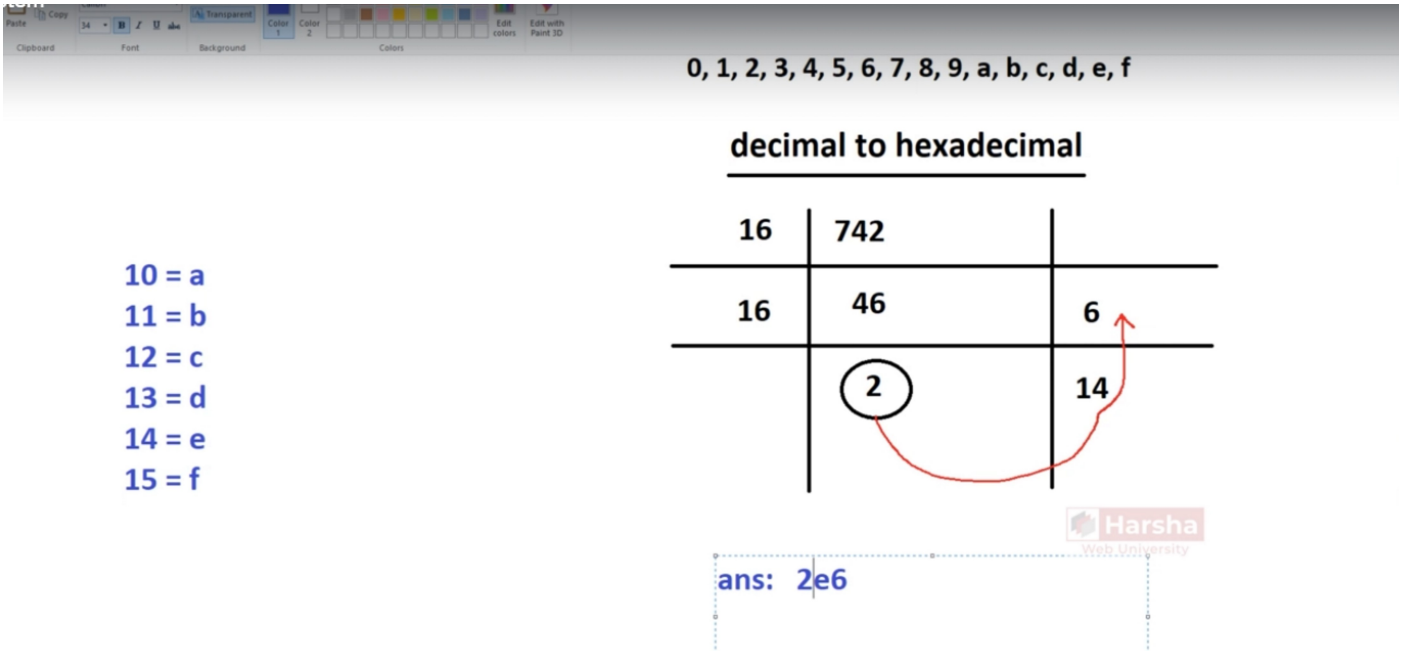


There is not possibility in octal literals in C#





For representation of larger number, hashing and encrypted data, also for representation of color code, in these case you require to use hexadecimal number system in the real life.



Introduction to Character Encoding

Introduction to Encoding

In general encoding means writing the characters in the specific format based on certain rules
in other words the process of converting the set of characters into another different format is called as encoding and convert
original format is called as decoding.

but here we are trying to understand the character encoding in specific that means
converting the characters into numerical format so here we have to assume a number for each character in the older compute
character encoding format by default.

ASCII stands for American standard code for information interchange according to which by default it supports 128 charact
according to ascii each character is represented as corresponding number so totally for 128 characters

we have 128 numbers that is 0 to 127 and each character occupies exactly 7 bits but in general it is considered as 1 byte.

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

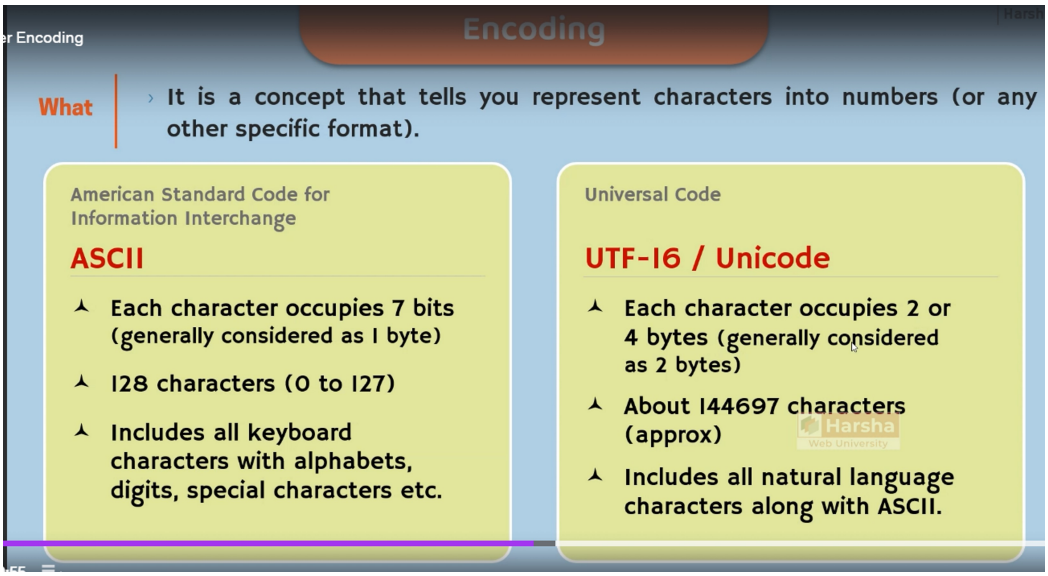
These are the character that are represented by ASCII. but the problem with ASCII
is that, is only supports english language. It does not support other natural language character such as arabic, bengali etc. In
UniCode characters are introduced.

ing those characters back to the

er systems ascii was used as

ters that is 0 to 127 in fact

order to solve this problem,



List of Uni-Code Characters

https://en.wikipedia.org/wiki/List_of_Unicode_characters

Bengali and Assamese [edit]

Main article: Bengali (Unicode block)

Bengali[1][2]
Official Unicode Consortium code chart (PDF)

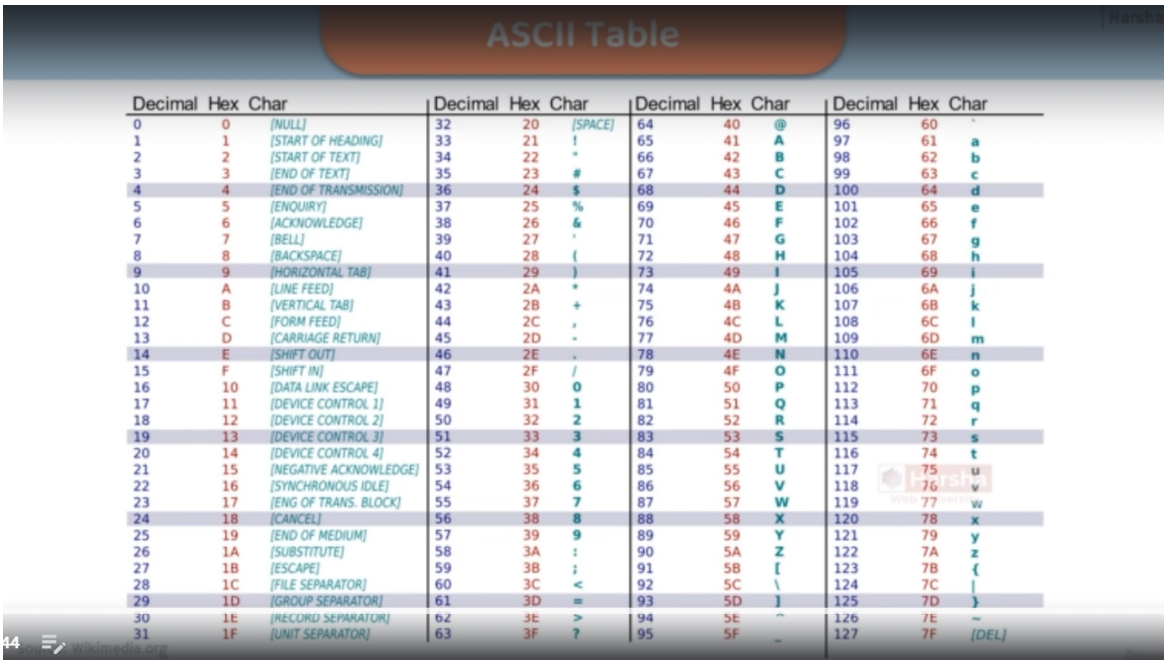
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+098x	ঀ	ঁ	ং	ঃ		অ	আ	ই	ঈ	উ	ঊ	ঋ	ঌ			এ
U+099x	ঐ			ও	ঔ	ক	খ	গ	ঘ	ঙ	চ	ছ	জ	ঝ	ঞ	ট
U+09Ax	ঠ	ড	ঢ	ণ	ত	থ	দ	ধ	ন		প	ফ	ব	ভ	ম	য
U+09Bx	র		ল				শ	ষ	স	হ			়	ঽ	া	ি
U+09Cx	ী	ু	ূ	ৃ	ৄ			ে	ৈ			ো	ৌ	্	ৎ	
U+09Dx								ৗ					ড়	ঢ়		য়
U+09Ex	ৠ	ৡ	ৢ	ৣ			০	১	২	৩	৪	৫	৬	৭	৮	৯
U+09Fx	ৰ	ৱ	৲	৳	৴	৵	৶	৷	৸	৹	৺	৻	ৼ	৽	৾	

Notes

1. ^ As of Unicode version 15.1
2. ^ Grey areas indicate non-assigned code points

ASCII Character Encoding

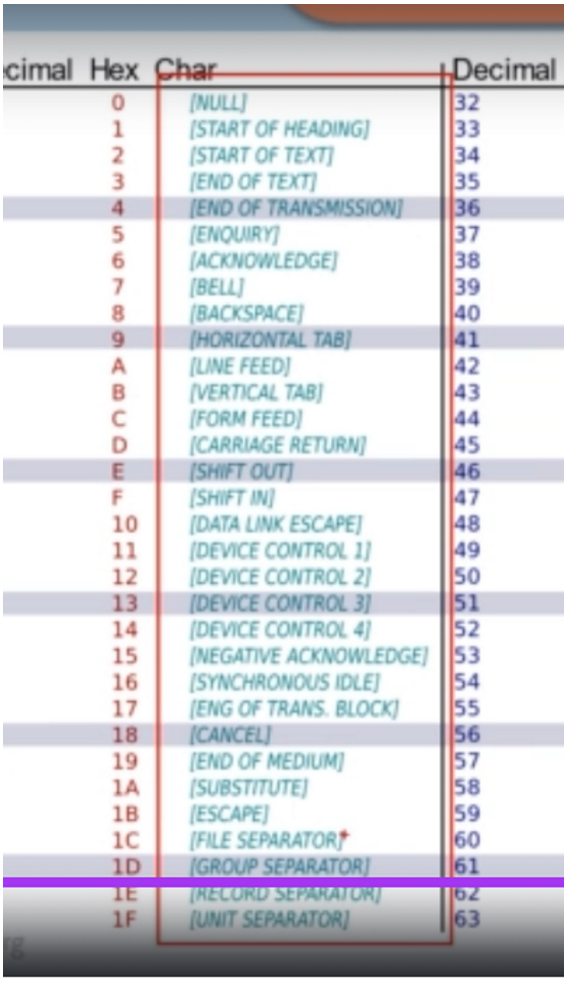
How do you convert Character into ASCII Code and Vice-versa in the C# Program?



ASCII Table

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

by default, ASCII code can be represented as byte data type. Because byte data type supports upto 256 character.
and ASCII code are available upto 127. so byte data type is more than enough for representation of ASCII code.



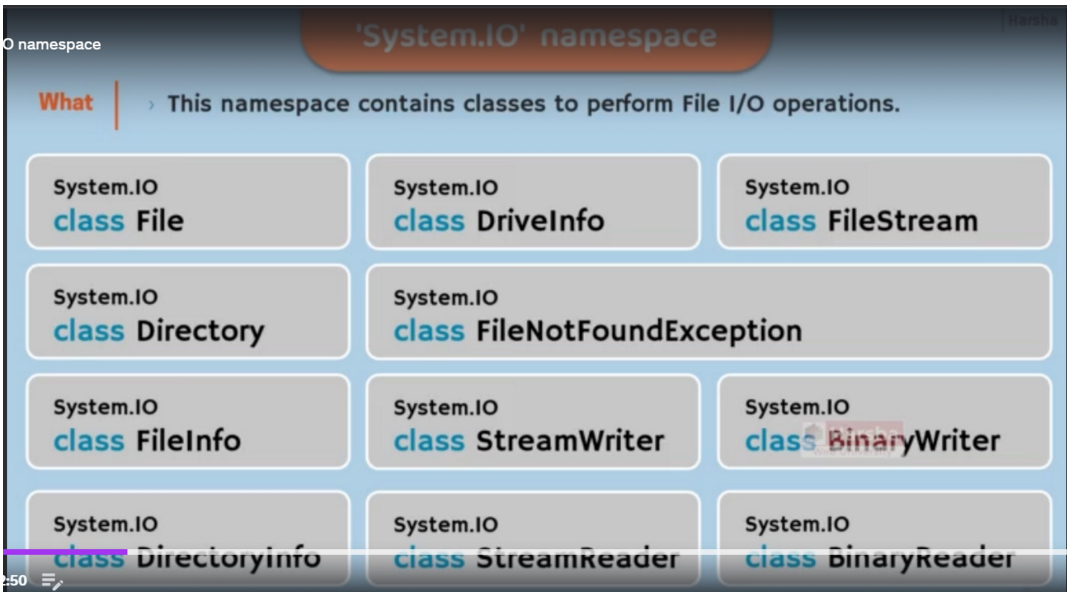
Decimal	Hex	Char
0	0	[NULL]
1	1	[START OF HEADING]
2	2	[START OF TEXT]
3	3	[END OF TEXT]
4	4	[END OF TRANSMISSION]
5	5	[ENQUIRY]
6	6	[ACKNOWLEDGE]
7	7	[BELL]
8	8	[BACKSPACE]
9	9	[HORIZONTAL TAB]
10	A	[LINE FEED]
11	B	[VERTICAL TAB]
12	C	[FORM FEED]
13	D	[CARRIAGE RETURN]
14	E	[SHIFT OUT]
15	F	[SHIFT IN]
16	10	[DATA LINK ESCAPE]
17	11	[DEVICE CONTROL 1]
18	12	[DEVICE CONTROL 2]
19	13	[DEVICE CONTROL 3]
20	14	[DEVICE CONTROL 4]
21	15	[NEGATIVE ACKNOWLEDGE]
22	16	[SYNCHRONOUS IDLE]
23	17	[ENG OF TRANS. BLOCK]
24	18	[CANCEL]
25	19	[END OF MEDIUM]
26	1A	[SUBSTITUTE]
27	1B	[ESCAPE]
28	1C	[FILE SEPARATOR]
29	1D	[GROUP SEPARATOR]
30	1E	[RECORD SEPARATOR]
31	1F	[UNIT SEPARATOR]
32	20	[SPACE]
33	21	!
34	22	"
35	23	#
36	24	$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	[DEL]

These are the characters which will not print in the console. but you must be able to see all these following character

UniCode Character Encoding

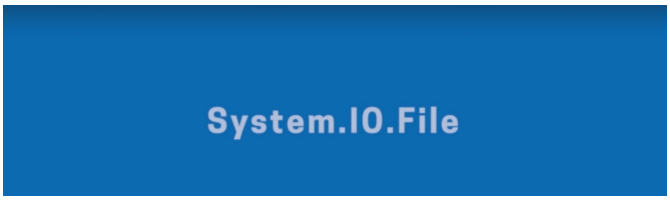
C# by default supports unicode character.

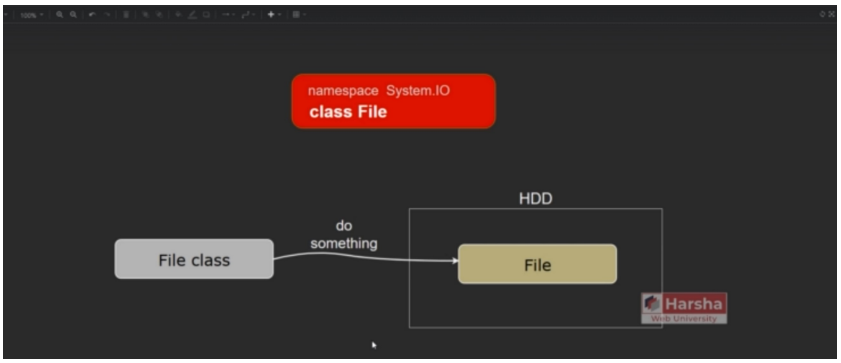
Introduction to System.IO namespace

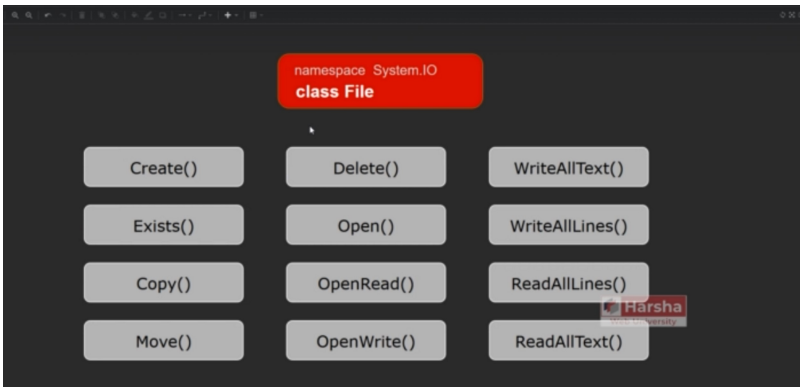


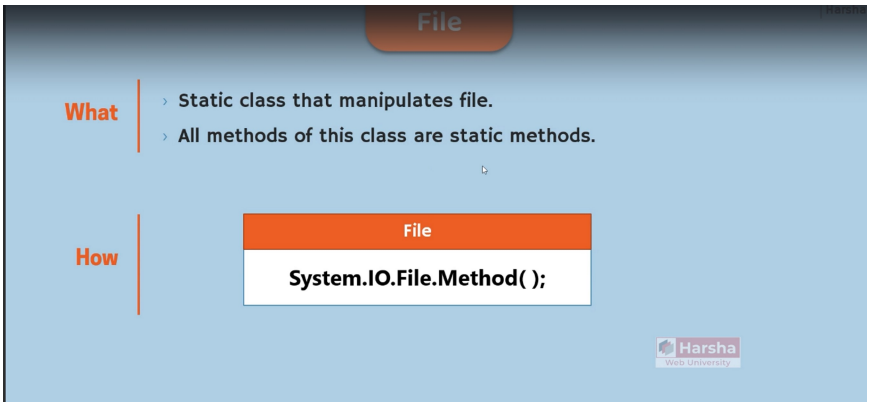
In dot net we have a predefined namespace called system.io
it contains several predefined classes as you can see which are used to perform various manipulations on files in the hard disk

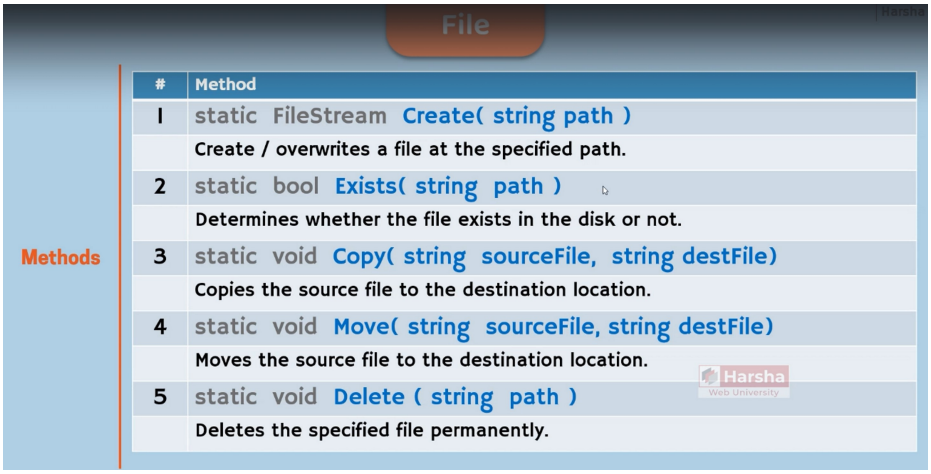
'File' class


System.IO.File

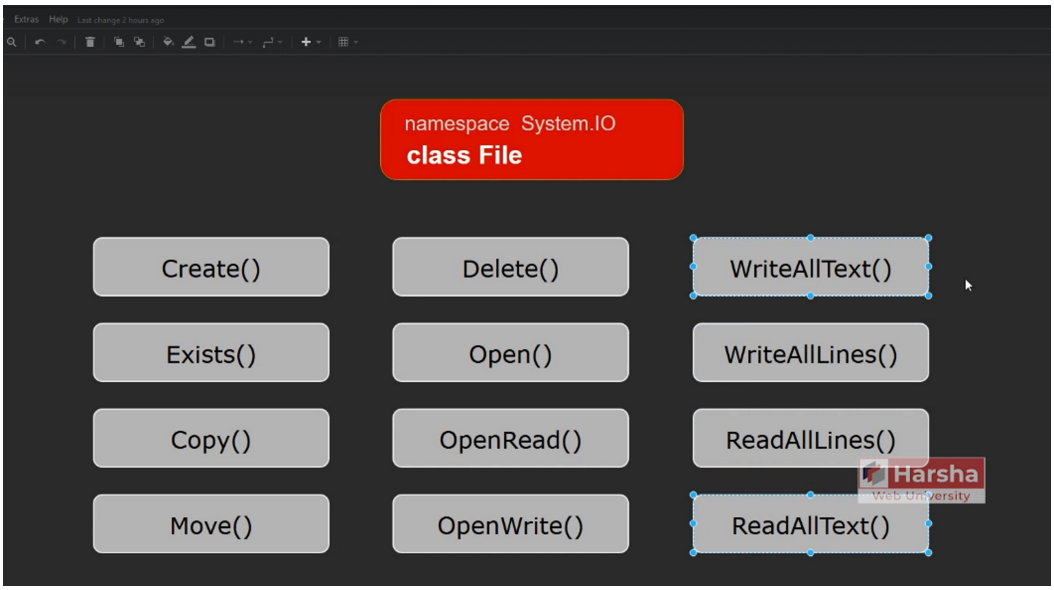

namespace System.IO
class File
HDD
do
something
File class
File
Harsha


namespace System.IO
class File
Create()
Delete()
WriteAllText()
Exists()
Open()
WriteAllLines()
Copy()
OpenRead()
ReadAllLines()
Move()
OpenWrite()
ReadAllText()
Harsha

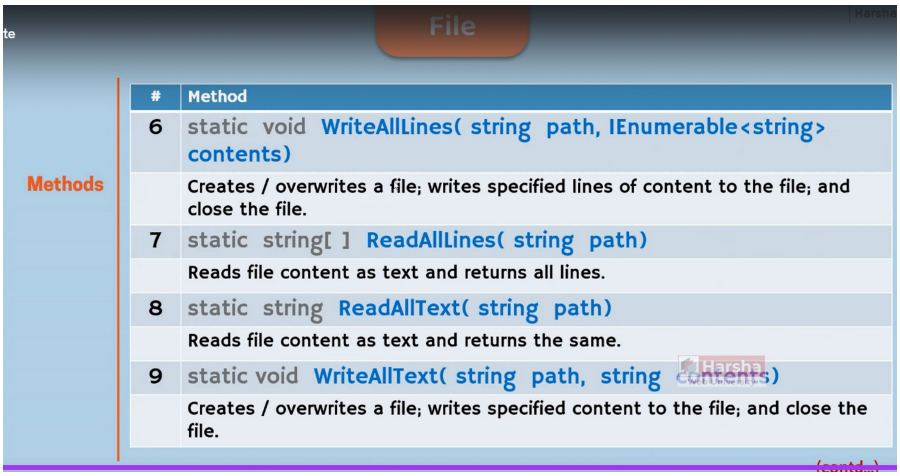

File
What
Static class that manipulates file.
All methods of this class are static methods.
How
File
System.IO.File.Method( );
Harsha



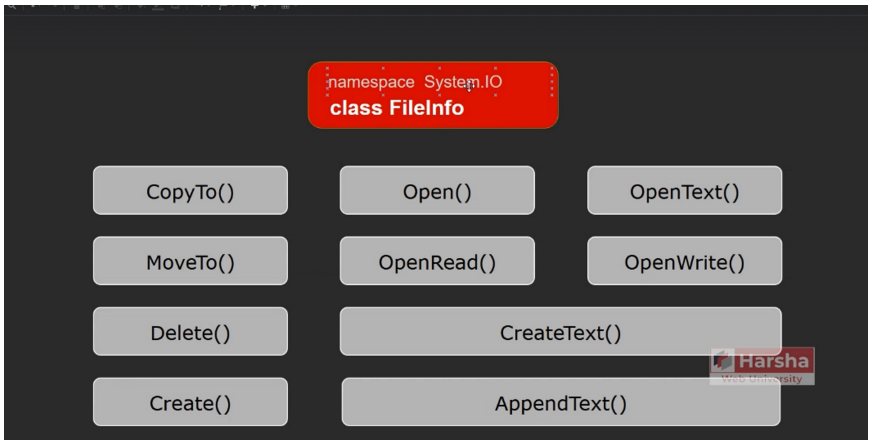
'File' class - Read and Write



For single string, WriteAllText() and ReadAllText() is best. But for a list of string, WriteAllLines() and ReadAllLines() is best.



FileInfo Class



FileInfo

What

› Represents a file and provides methods to manipulate the file.

How

FileInfo

```
FileInfo referenceVariable = new FileInfo( "Your File Path Here");
```

Constructors

Constructor	Description
FileInfo(string filePath)	It initializes the file path which needs to be manipulated.

File

Methods

#	Method
1	static FileStream Create(string path)
	Create / overwrites a file at the specified path.
2	static bool Exists(string path)
	Determines whether the file exists in the disk or not.
3	static void Copy(string sourceFile, string destFile)
	Copies the source file to the destination location.
4	static void Move(string sourceFile, string destFile)
	Moves the source file to the destination location.
5	static void Delete (string path)
	Deletes the specified file permanently.

FileInfo

Methods

#	Method
1	FileInfo CopyTo(string destFilePath, bool overwrite)
	Copies the file into the new destination path. It returns an object of FileInfo class, that represents the newly created file.
2	void MoveTo(string destFilePath)
	Moves the file into the new destination path (should be in the same drive).
3	void Delete( )
	Deletes the file permanently.

(contd...)

File class is good for single operation; here you do not have to create an instance time you have to supply the FilePath,

if you have to perform multiple operation, then use FileInfo class...

The file info class can provide additional information or details about the file for example you have a hard disk and it has a file you would like to read it as full path file extension length of the file and also the date on time of file creation etc you will be able to read all the file details in the form of these properti of the file info class.

e but every

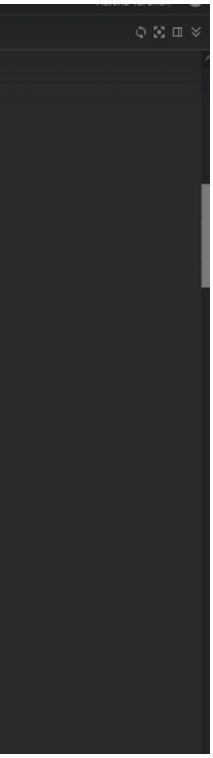
s details such

ies

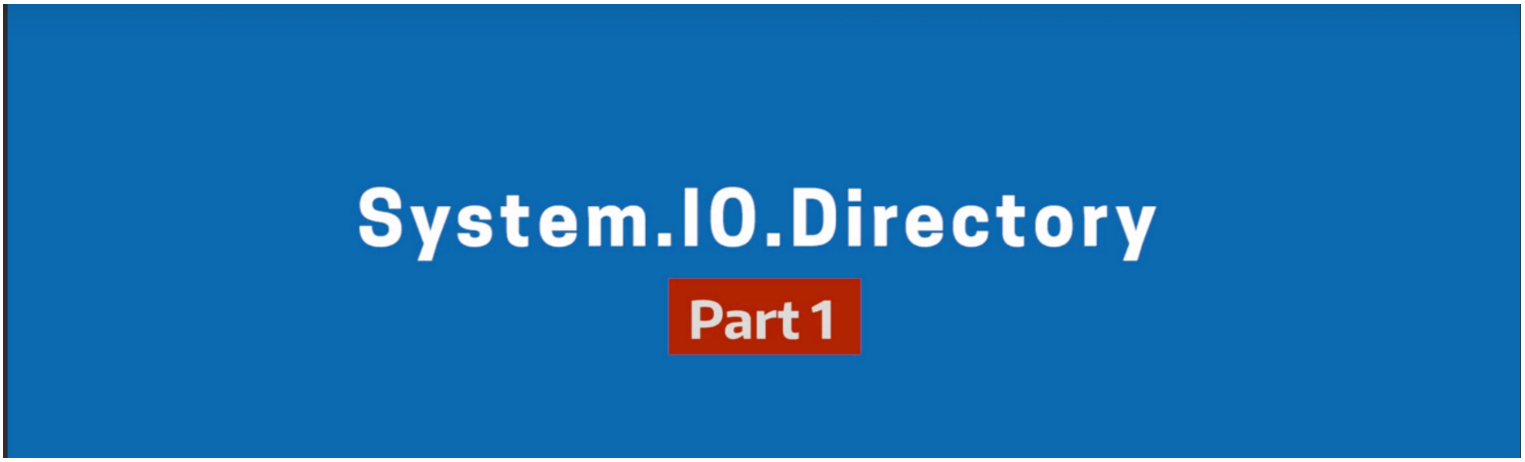
FileInfo

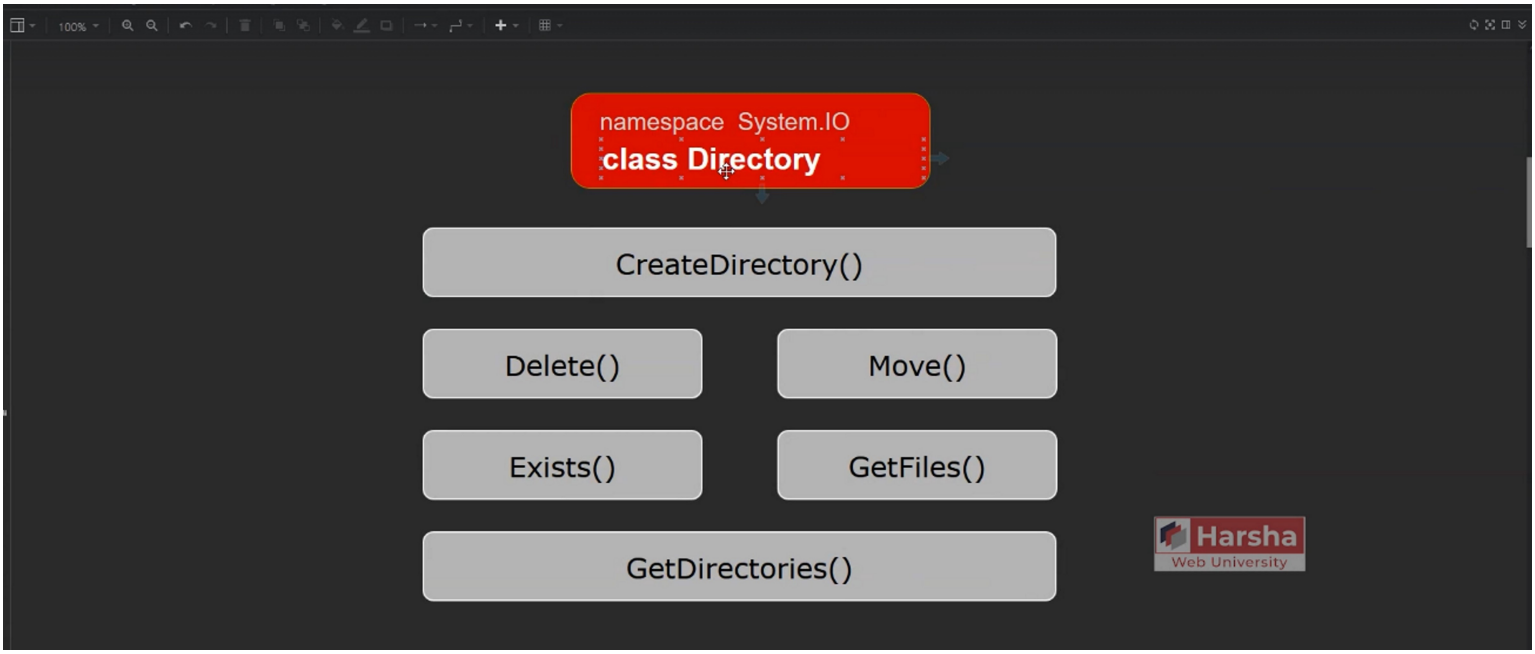
Properties

Property	Description
bool **Exists**	Determines whether the file exists in the disk or not.
string **FullName**	Represents full path (including file name and extension) of the file.
string **Name**	Represents only name of the file (without path).
string **DirectoryName**	Represents only path of the file (without file name).
string **Extension**	Represents only file extension (without file name).
DateTime **CreationTime**	Represents date and time of file creation.
DateTime **LastWriteTime**	Represents date and time of last modification of the file.
DateTime **LastAccessTime**	Represents date and time of last access of the file.
long **Length**	Represents file size (in the form of no. of bytes).



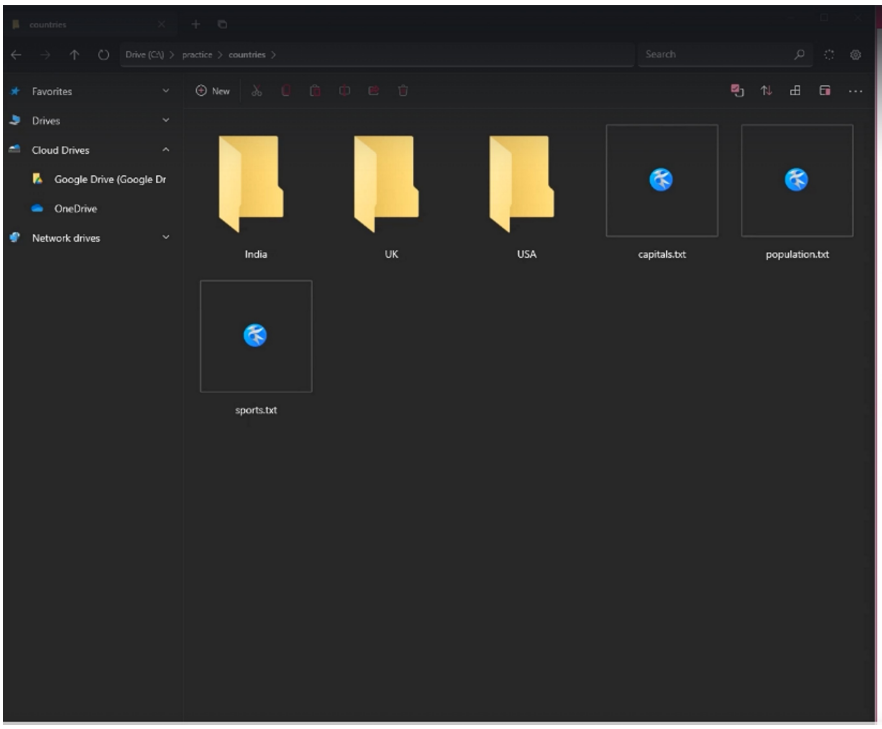
'Directory' class - Part 1

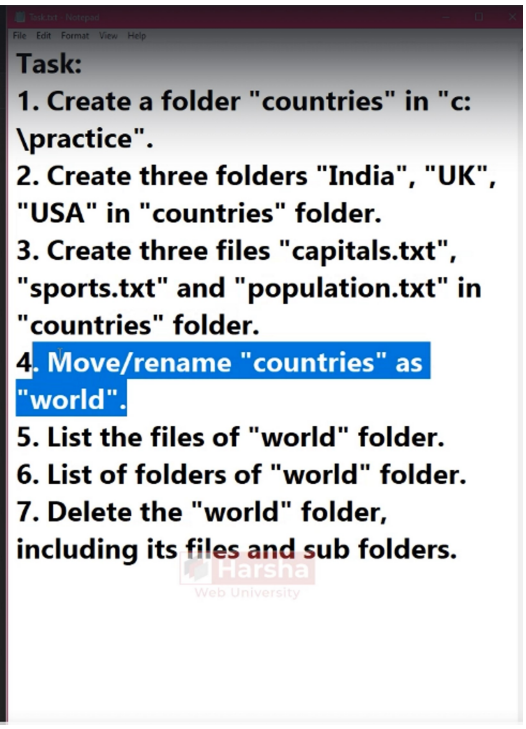





```
            new FileStream()
    File.Create(capitalsFilePath).Close();
    File.Create(sportsFilePath).Close();
    File.Create(populationFilePath).Close();
    Console.ReadKey();
}
```

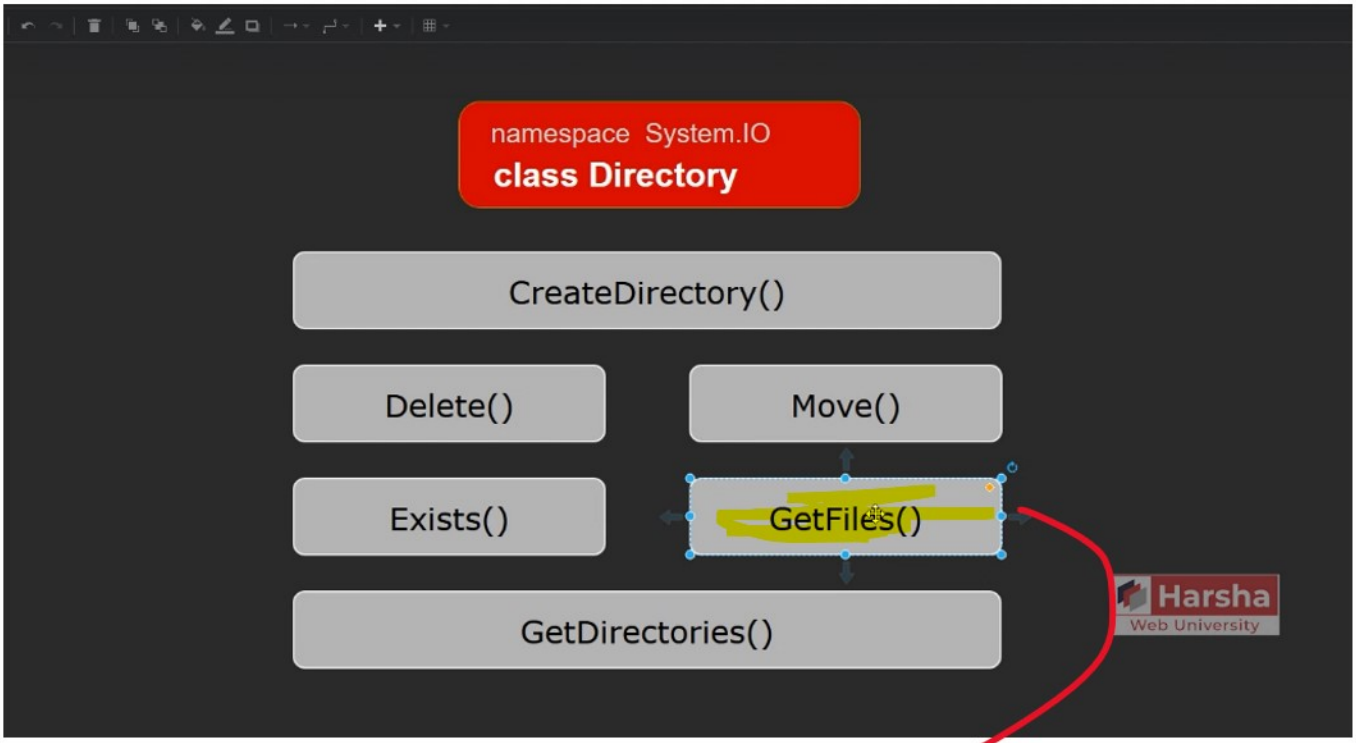
'Directory' class - Part 2

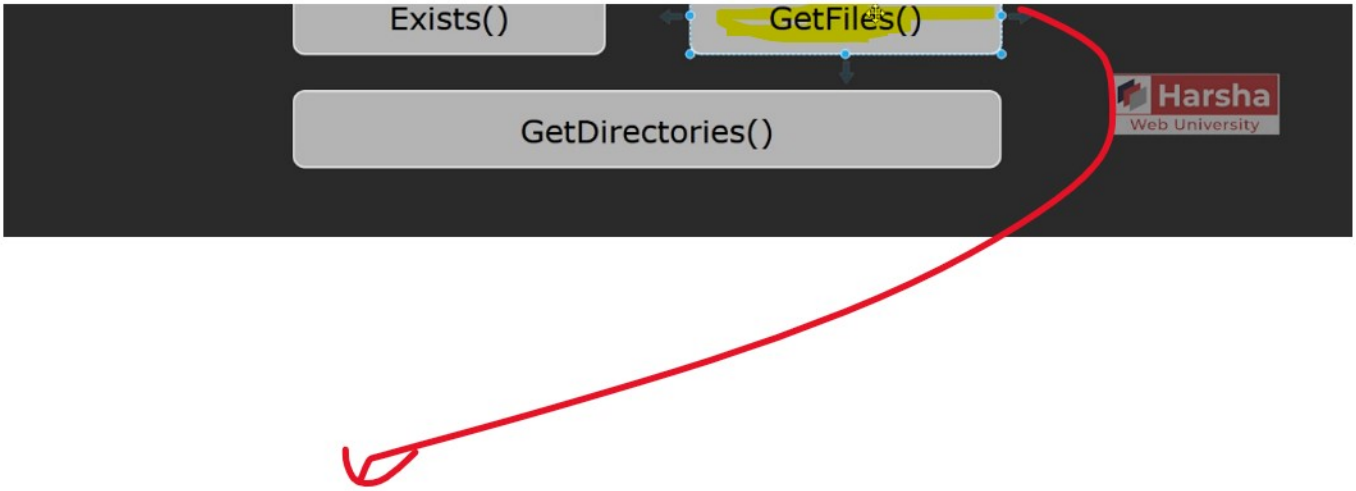




Task:

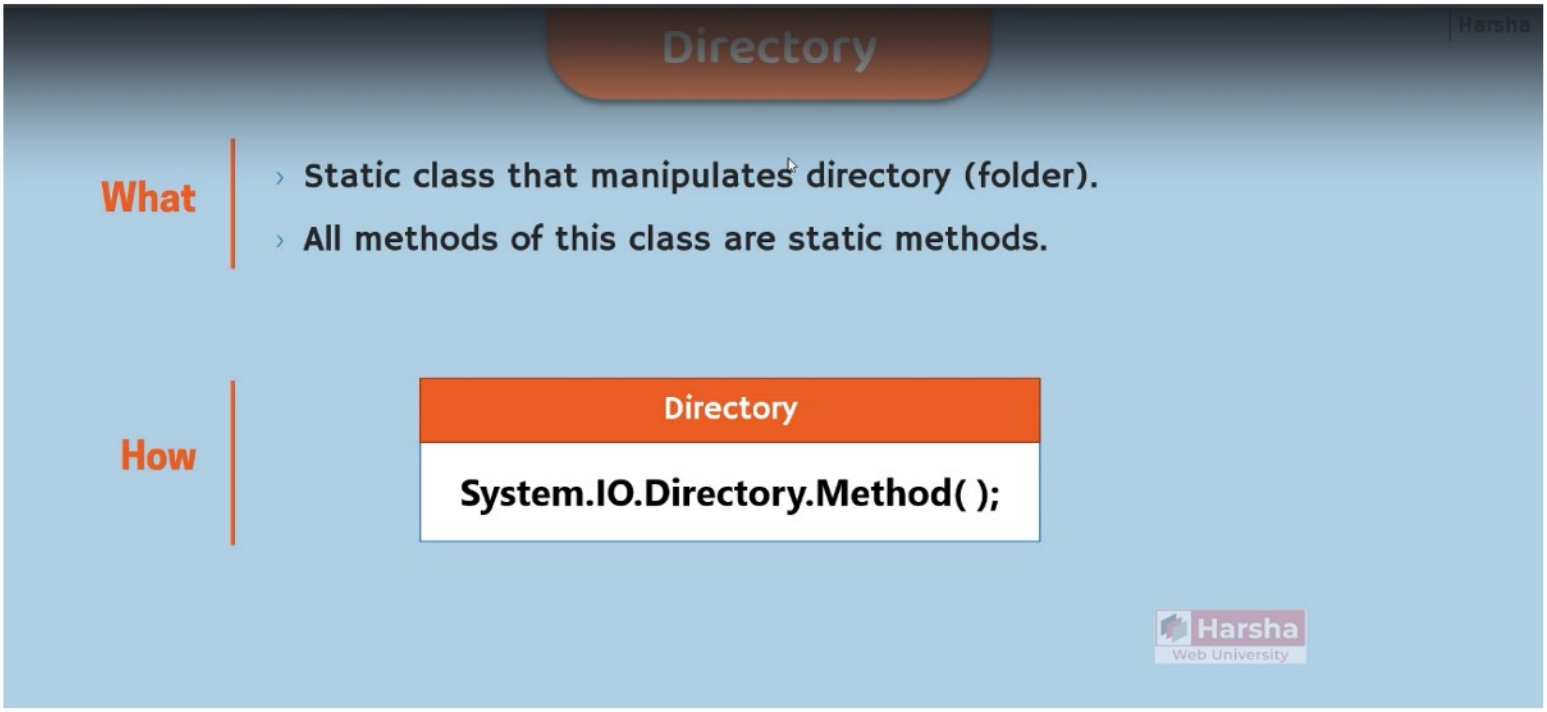
1. Create a folder "countries" in "c:\practice".
2. Create three folders "India", "UK", "USA" in "countries" folder.
3. Create three files "capitals.txt", "sports.txt" and "population.txt" in "countries" folder.
4. Move/rename "countries" as "world".
5. List the files of "world" folder.
6. List of folders of "world" folder.
7. Delete the "world" folder, including its files and sub folders.





In Order to get files from a specific folder, we have GetFiles() fol

But the problem with Directory class is that we have to supply the folderPath every
But if we want to do multiple operation with the same file, use 'DirectoryInfo' clas



der

ytime.
s.

Directory

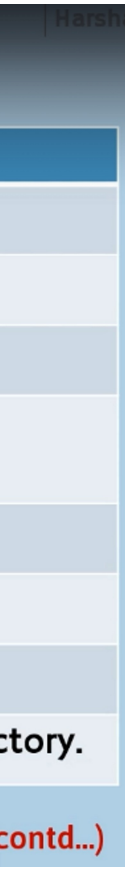
Methods

#	Method
1	static DirectoryInfo CreateDirectory(string path)
	Create a directory at the specified path.
2	static void Delete (string path, bool recursive)
	true: Deletes the directory including sub directories and all files. false: Deletes the directory only if it is empty.
3	static bool Exists(string path)
	Determines whether the directory exists in the disk or not.
4	static string[ ] GetDirectories(string path)
	Returns string[ ] that contains paths of sub directories of specified direc

Directory

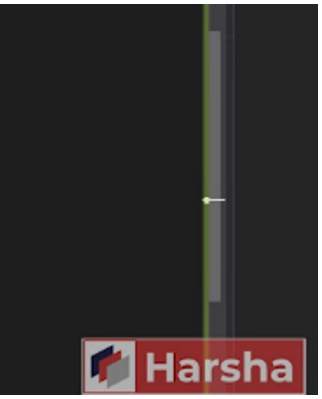
Methods

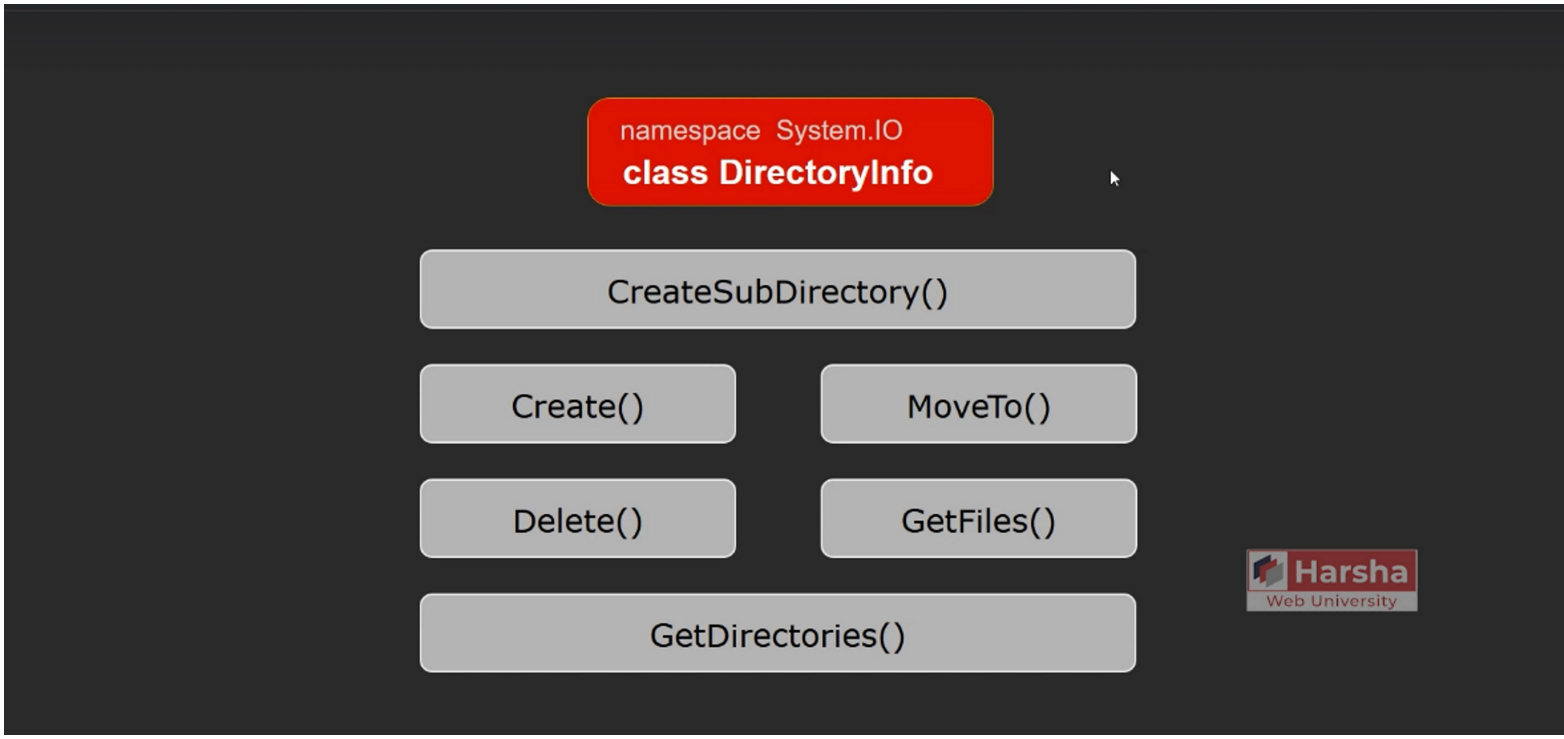
#	Method
5	static string[ ] GetDirectories(string path, string searchPattern)
	Returns string[ ] that contains paths of sub directories that matches with specified search pattern.
6	static string[ ] GetFiles(string path)
	Returns string[ ] that contains paths of files of specified directory.
7	static string[ ] GetFiles(string path, string searchPattern)
	Returns string[ ] that contains paths of files that matches with specified search pattern.
8	static void Move(string sourceDir, string destDir)
	Moves the source directory to the specified destination location in the same drive.


ctory.
contd...)

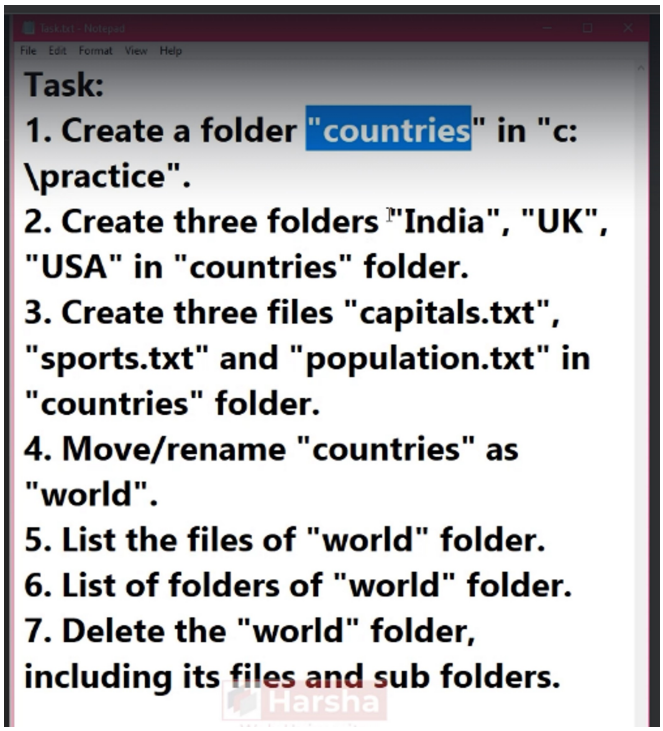
```
38
39      //GetFiles
40      string[] files = Directory.GetFiles(worldFolderPath, "c*.txt");
41      Console.WriteLine("\nFiles:");
42      foreach (string file in files)
43      {
```

'DirectoryInfo' Class


Harsha



Let's do the same task using FileInfo Class



DirectoryInfo

What

- Class that represents a directory (folder) on the disk and performs manipulations on directories.

How

DirectoryInfo

```
DirectoryInfo referenceVariable = new DirectoryInfo( "Directory Path Here");
```

Constructors

Constructor	Description
DirectoryInfo(string path)	It initializes the directory path which needs to be manipulated.

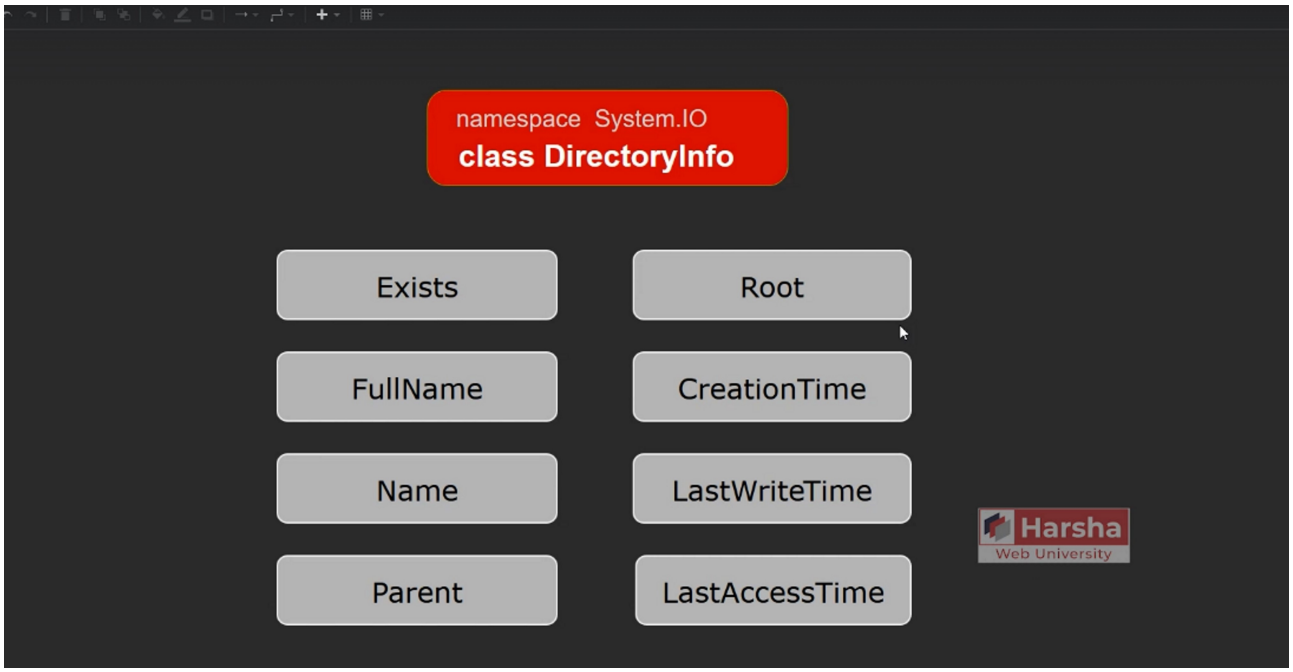
DirectoryInfo

Methods

#	Method
1	void Create( )
	Create the directory at the current path.
2	DirectoryInfo CreateSubDirectory (string path)
	Creates a sub directory at the current path with specified name.
3	void Delete(bool recursive)
	true: Deletes the directory including sub directories. false: Deletes the directory only if it is empty.
4	DirectoryInfo[ ] GetDirectories()
	Returns DirectoryInfo[ ] that represents sub directories of current directory.

(contd...)

'DirectoryInfo' class - Properties



When you create an object of this directoryinfo class it will automatically read so information regarding that particular folder for example there is a folder called sa have created an object of directory info class and specified the path of the same fo reads the details of that sample folder and those details are available in the form o

DirectoryInfo

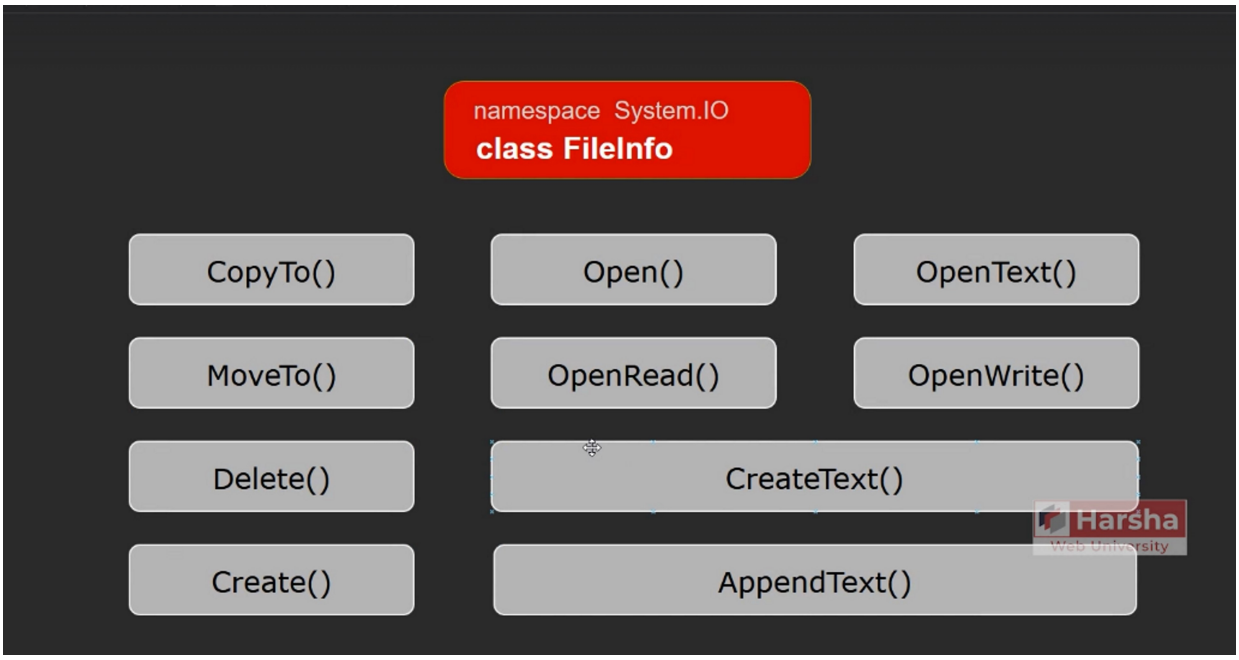
Properties

Property	Description
bool Exists	Determines whether the directory exists in the disk or not.
string FullName	Represents full path of the directory
string Name	Represents only name of the directory (without path).
DirectoryInfo Parent	Represents parent directory of the current directory.
DirectoryInfo Root	Represents root (drive) of the directory.
DateTime CreationTime	Represents date and time of directory creation.
DateTime LastWriteTime	Represents date and time of last modification of the directory.
DateTime LastAccessTime	Represents date and time of last access of the directory.

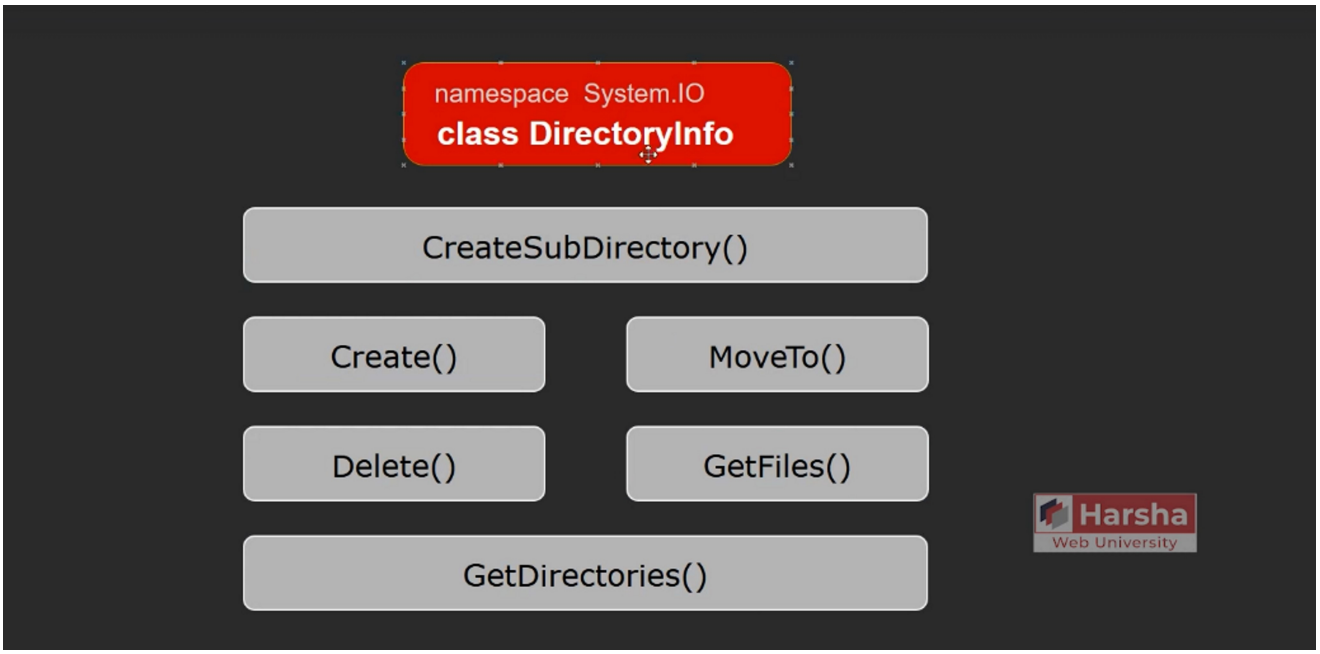
me additional details or
mple in c drive and you
lder then it automatically
f these properties.

'DriveInfo' class

In order to manipulate the file, we have FileInfo class,

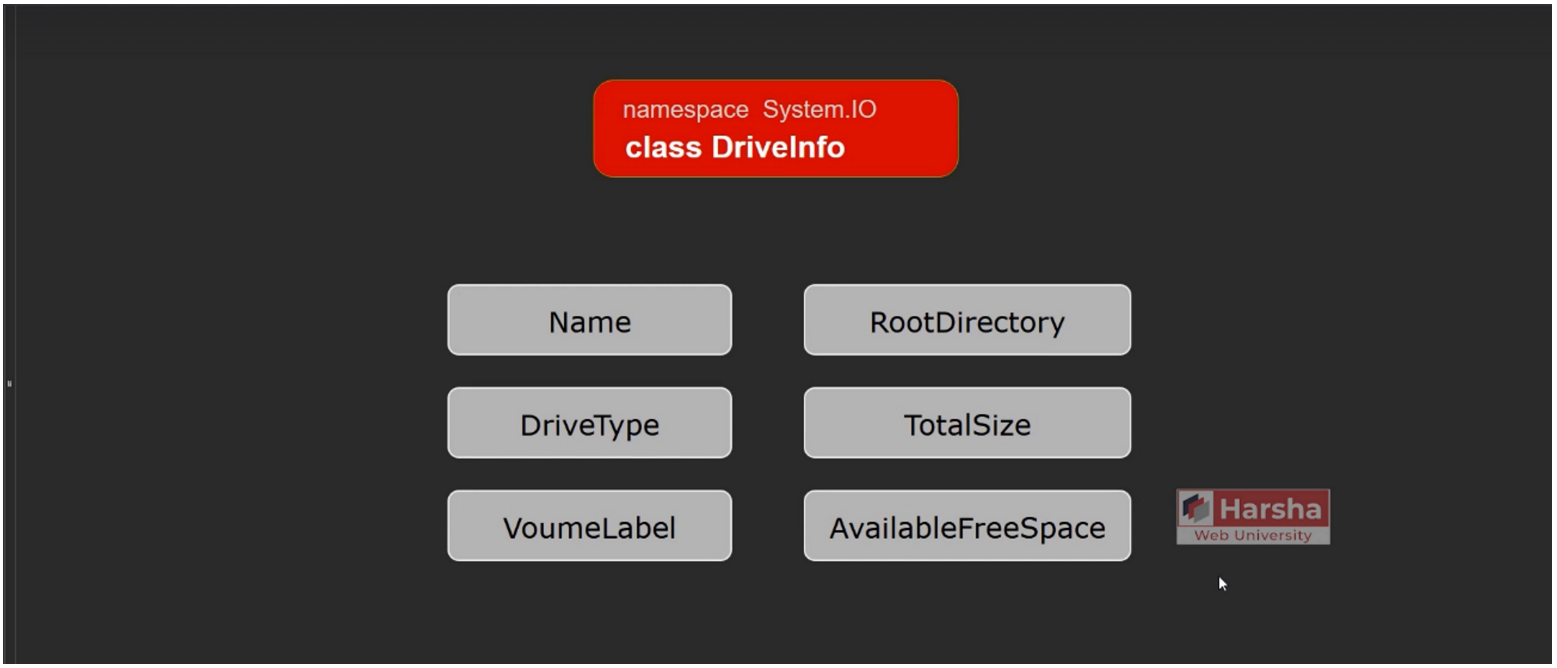


In order to directory, we have DirectoryInfo class.



and in order to manipulate the Drive I mean get details about a particular drive, w
'DriveInfo' class.

ve have



DriveInfo

What

- Class that represents a drive and performs manipulations on drives.
- You can read both fixed / removable drives.

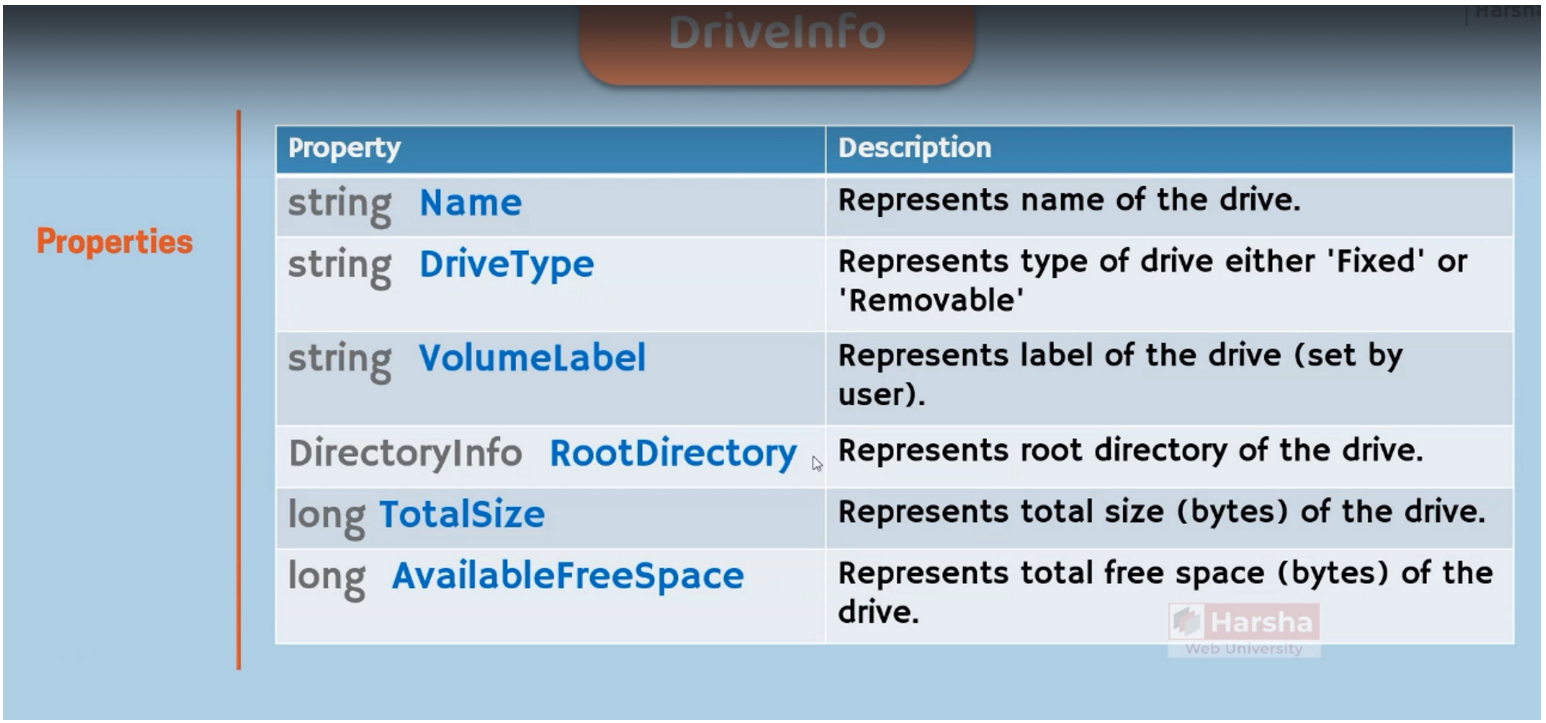
How

DriveInfo

```
DriveInfo referenceVariable = new DriveInfo( "Your Drive Name Here");
```

Constructors

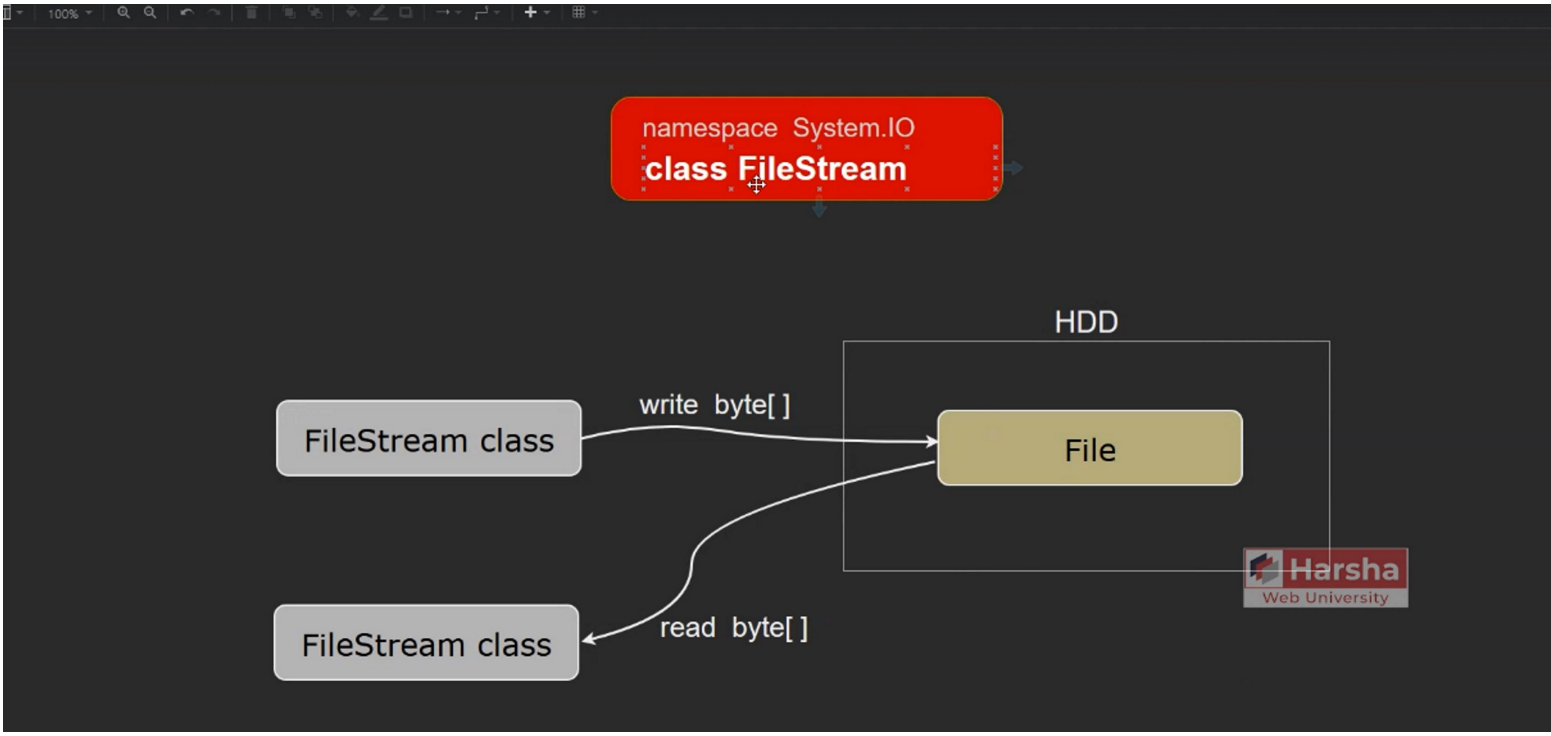
Constructor	Description
DriveInfo(string path)	It initializes the drive name which needs to be manipulated.

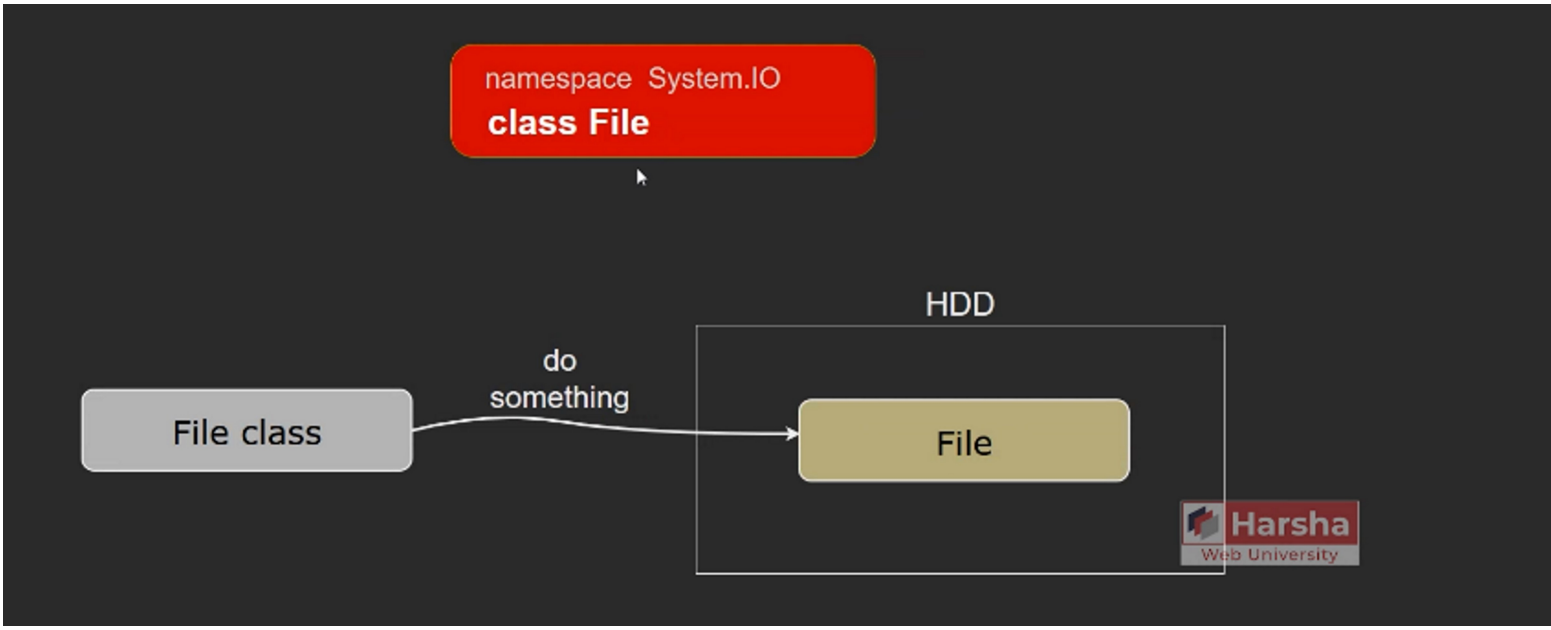


FileStream Class - Part 1

The file stream is such a class using which you can write content into the file or r
from the file but wait

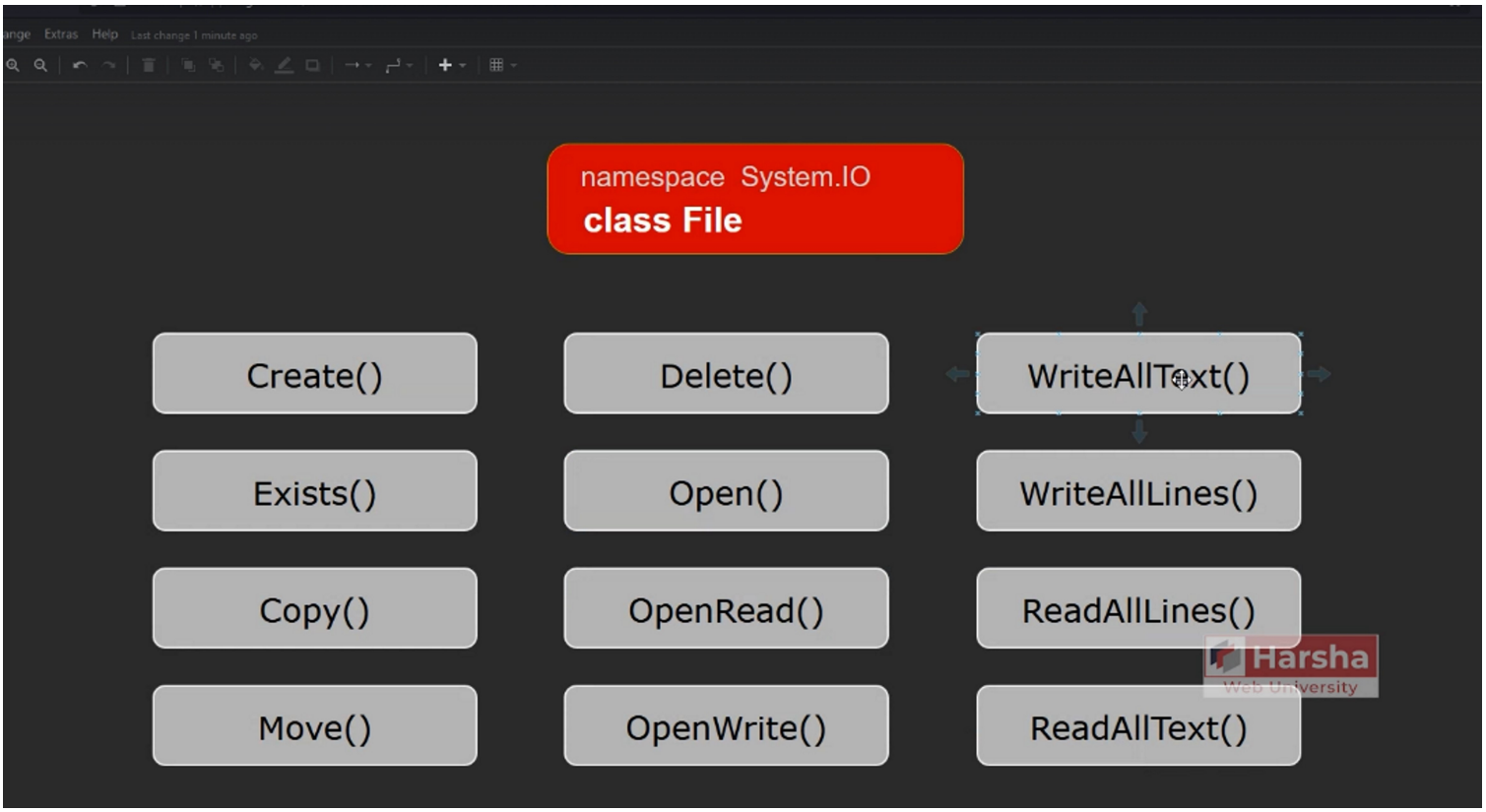
read existing content





Already we are able to perform file IO operations that is writing and reading ope
by using the predefined class called File. the file class has methods like readallte
writealltext etc in order to perform writing and reading operations so what is the
stream class.

rations
xt
e necessity of file



The file class is helpful when you want to perform a single operation with the
file that means you have a string value with you at a single shot you would like to write some content into that particular
perform multiple operations to the file then you can quickly use file dot write alt text.

FileStream: you would like to write some text into the file and then after that execute some other code again you want
into the file again execute some other code like this in the middle of the big program at different stages you would like to
into the file.

file that's it you don't want to

to write some more information
keep writing some information

File

Methods

#	Method
1	static FileStream Create(string path)
	Create / overwrites a file at the specified path.
2	static bool Exists(string path)
	Determines whether the file exists in the disk or not.
3	static void Copy(string sourceFile, string destFile)
	Copies the source file to the destination location.
4	static void Move(string sourceFile, string destFile)
	Moves the source file to the destination location.
5	static void Delete (string path)
	Deletes the specified file permanently.

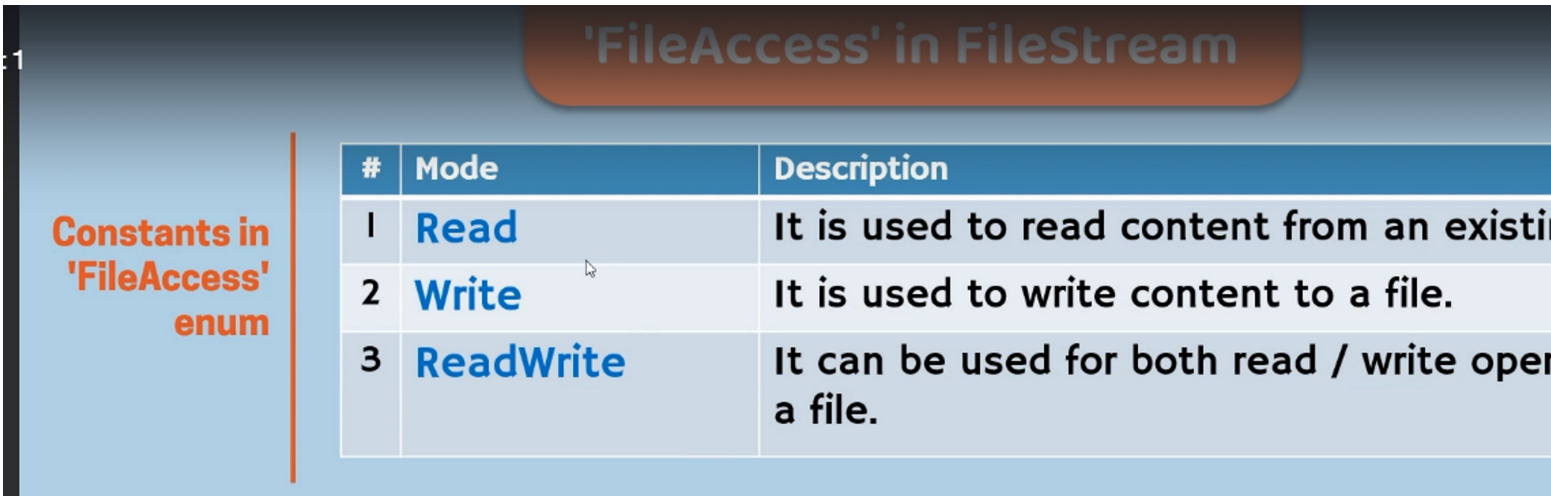
File

Methods

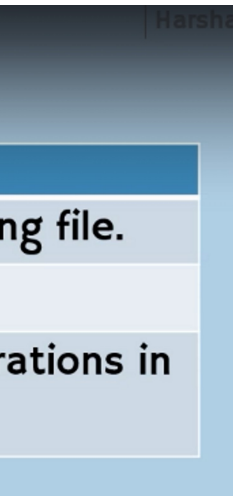
#	Method
10	static FileStream Open(string path, FileMode mode, FileAccess access)
	Opens the file in specified file mode with specified file access permission and returns a new object of FileStream type.
11	static FileStream OpenRead(string path)
	Opens the file in 'Open' mode and returns a new object of FileStream type.
12	static FileStream OpenWrite(string path)
	Opens the file in 'OpenOrCreate' mode and returns an object of FileStream type.

'FileMode' in FileStream

#	Mode	Description
1	CreateNew	It specifies that the o/s should create a new file. If the file already exists an IOException will be thrown.
2	Create	It specifies that the o/s should create a new file. If the file already exists, it will be overwritten.
3	Open	It specifies that the o/s should open an existing file. If the file doesn't exist, a FileNotFoundException will be thrown.
4	OpenOrCreate	It specifies that the o/s should open an existing file. If the file doesn't exist, a new file will be created. It is useful for reading content of the file.
5	Append	It specifies that the o/s should open an existing file; and seek to end of the file, in order to write content at the end. It is useful with FileAccess.Write



FileStream class - Part 2


ng file.
rations in

```
        //create content
        string content = "Dog is one of the domestic animal";
        byte[] bytes = System.Text.Encoding.ASCII.GetBytes(content);

        //Write
        fileStream.Write(bytes, )
    }
```

Offset

System.IO.FileStream

Part 3


sha
rsity

FileStream

Constructors

#	Constructor
1	FileStream(string filePath, FileMode mode, FileAcc access)
	It initializes opens the file at specified mode and specified a permission. FileMode: CreateNew, Create, Open, OpenOrCreate, Appen FileAccess: Read, Write, ReadWrite

'FileMode' in FileStream

#	Mode	Description
1	CreateNew	It specifies that the o/s should create a new file. If the file already exists an IOException will be thrown.
2	Create	It specifies that the o/s should create a new file. If the file already exists, it will be overwritten.
3	Open	It specifies that the o/s should open an existing file. If the file doesn't exist, a FileNotFoundException will be thrown.
4	OpenOrCreate	It specifies that the o/s should open an existing file. If the file doesn't exist, a new file will be created. It is useful for reading content of the file.
5	Append	It specifies that the o/s should open an existing file; and seek to end of the file, in order to write content at the end. It is useful with FileAccess.Write


Harsha
cess
ccess
nd

'FileAccess' in FileStream

Constants in 'FileAccess' enum

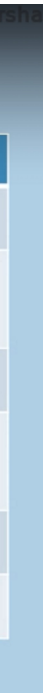
#	Mode	Description
1	Read	It is used to read content from an existing file.
2	Write	It is used to write content to a file.
3	ReadWrite	It can be used for both read / write operations in a file.

FileStream

Methods

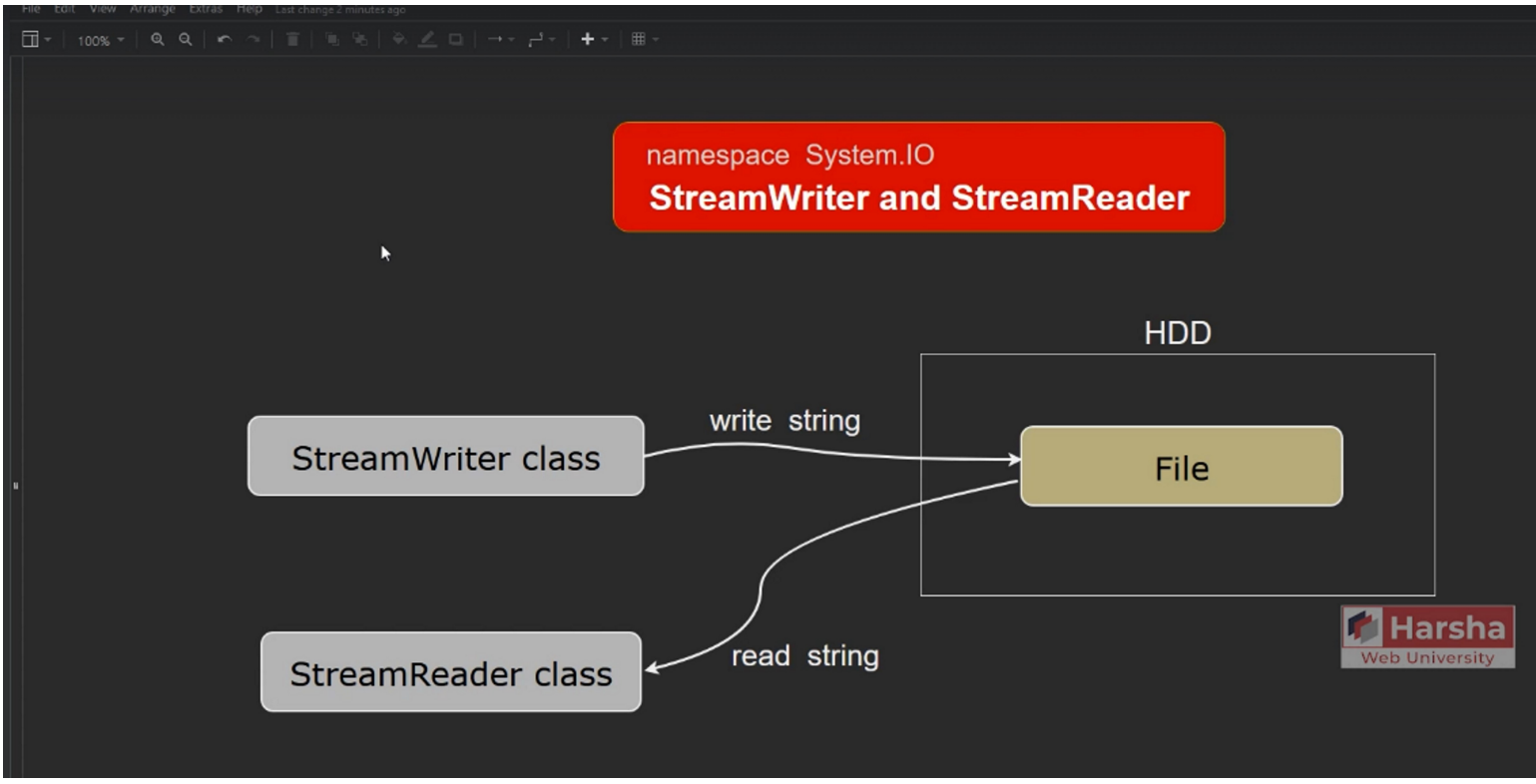
#	Method
1	void Write(byte[] array, int offset, int count)
	Writes the specified no. of bytes specified by count, based on the byte[] specified by 'array' into the file, after the no. of bytes specified by 'offset'.
2	int Read(byte[] array, int offset, int count)
	Read the specified no. of bytes specified by count, into the byte[] specified by 'array' from the file, after the no. of bytes specified by 'offset'.
3	void Close( )
	Closes the file.

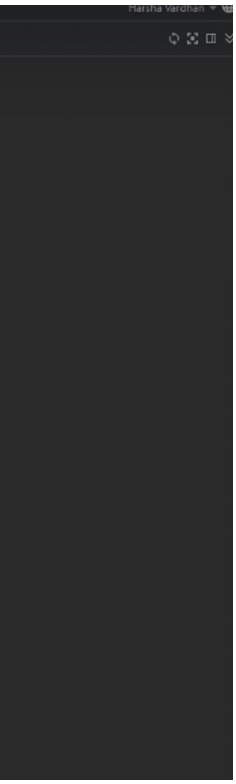
Only the problem with this file stream is that every time for writing as well as re
you have to convert the string into battery and vice versa every time. So this pro
lengthy and confusing sometimes in order to perform the easier operation on the
and stream reader is better.



eading
oblem makes the code
e same stream writer

StreamWriter and StreamReader - Part 1





When you want to perform multiple operations on the same file, the File Stream class is better
but of course when you want to perform single operation like only you would like to write a single content into the
file or single reading operation in that case the File class always the best because it has the predefined methods
which can be used directly like write all text read all text we have already seen those methods in
the file class in the previous lectures but in case of multiple operations like for example at the beginning of the
program you would like to write line number 1 into the file then after that after execution of some particular code
you would like to write another line into the file again after some piece of code you would like to write the third line int
to keep performing multiple operations like multiple times writing
or multiple times reading if this is the case the file stream is the best so far that we know but what's the problem with the
perform the writing and reading operation you require to manually convert your string value into byte array and byte arr

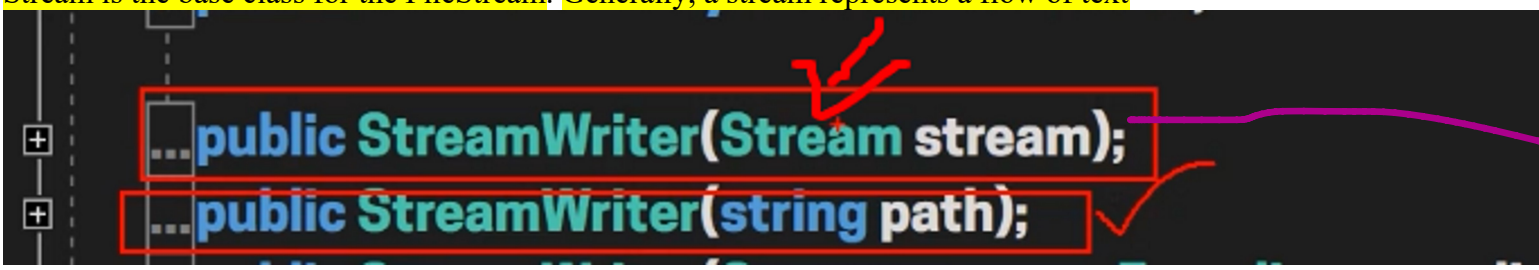
so this makes the code lengthy and cumbersome"
to overcome this problem if there is a class that allows you to write the string values into the file directly and also can re
that would be better such classes are stream writer and stream reader the stream writer allows you to write a string direct
not manually convert that string into byte array but of course it does the same automatically and internally without
your effort and in the same way it allows you to read the existing content from the file and gives you the same as a string
reads the content as a byte array and automatically converts the same into
string and it gives you the string value directly so these allow you to operate on strings but internally they will handle th
that is the beauty of these classes but the limitation for these classes is
it works with only text information means text files they cannot work on other than text formats
such as for example they cannot work on images audio files video files and also documents or pdf files they
cannot work on that in case if you want to work on those files again you have to go back to the files frame
but of course working with the audio files video files and pdf files etc require you some additional plugins so

it is out of context at this moment

so let's stick to the text files now

that is stream writer and stream reader

Stream is the base class for the FileStream. Generally, a stream represents a flow of text



For example i am giving you the first byte and the next byte and so on so as a flow I keep on giving the data
to you and you are keep on receiving the same this flow of data is called as a stream.

Stream is a parent class which can represent any type of stream. Stream of information. The FileStream class is the chil

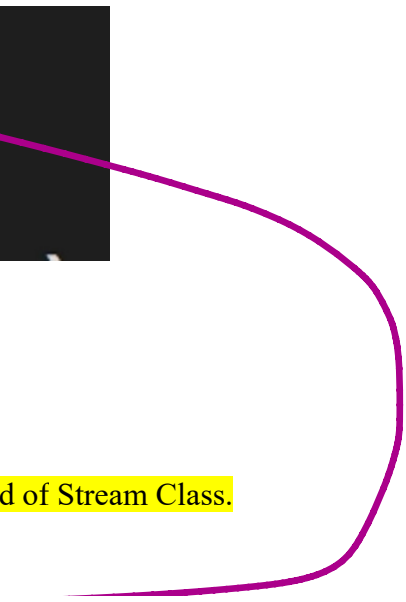
o the file like that you would like

e file stream every time when you
ay into string while reading

ad the strings from the file so
tly into the file you need

g value directly but internally it

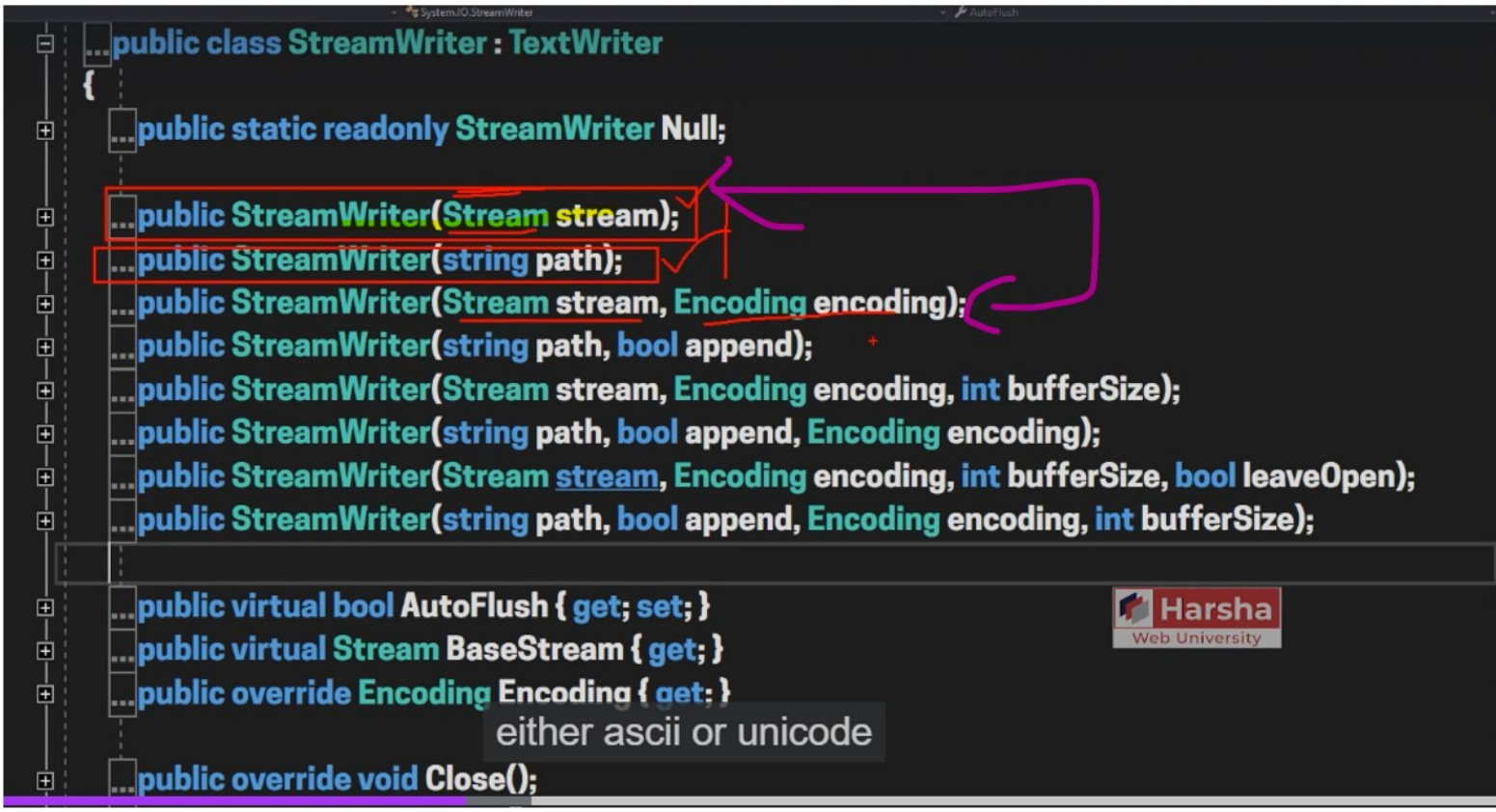
e conversions into byte array so



d of Stream Class.

Stream is a parent class which can represent any type of stream. Stream of information. The FileStream class is the chi

you can supply the child of FileStream class here



d of Stream Class.

```
class Program
{
    0 references
    static void Main()
    {
        string filePath = @"c:\practice\europe.txt";
        //StreamWriter streamWriter = new StreamWriter(filePath);
        FileStream fileStream = new FileStream(filePath, FileMode.Create, FileAccess.Write);
        StreamWriter streamWriter = new StreamWriter(fileStream);
        streamWriter.WriteLine("Hello")
    }
}
```

StreamWriter

FileStream Class

StreamWriter Class is a wrapper of FileStream Class.

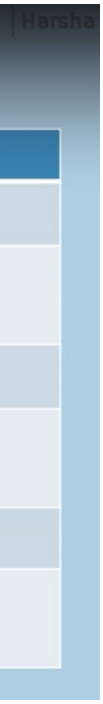
FileInfo

Methods

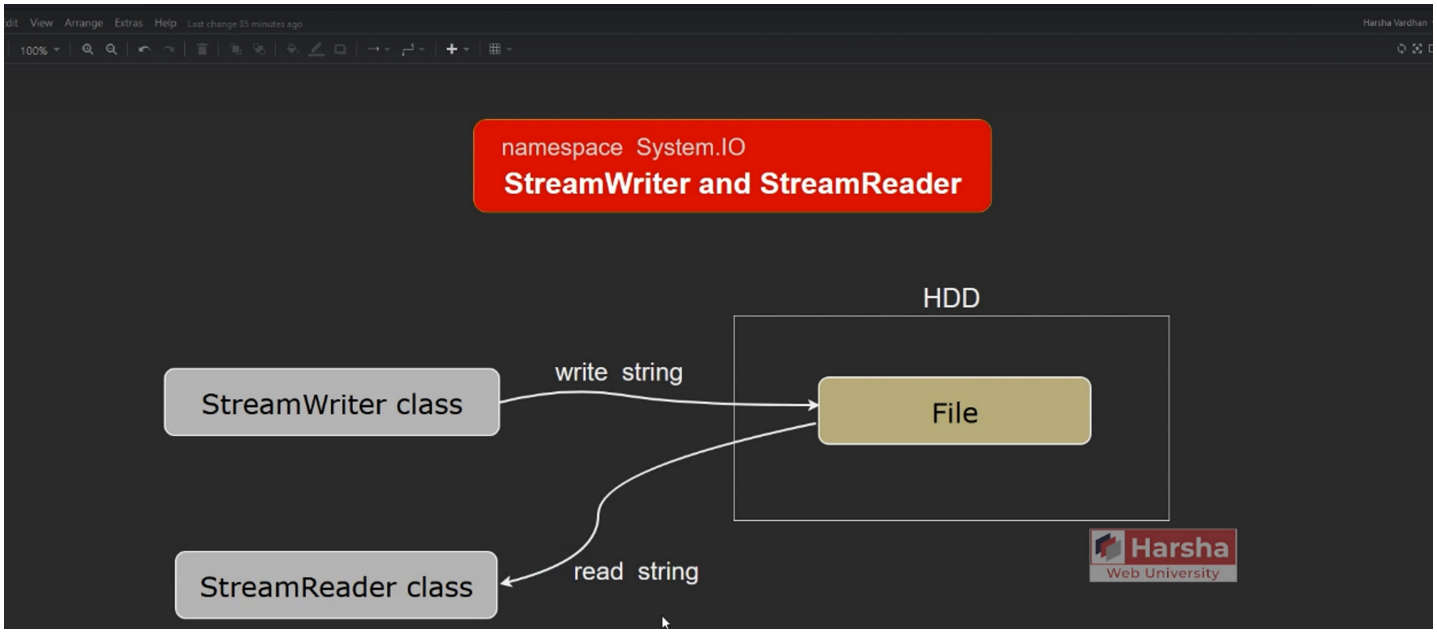
#	Method
8	StreamWriter CreateText( )
	Opens the file in 'Create' mode & creates and returns an object of StreamWriter class that can write text to the file.
9	StreamWriter AppendText( )
	Opens the file in 'Append' mode & creates and returns an object of StreamWriter class that can write the appended text to the file.
10	StreamReader OpenText( )
	Opens the file in 'Open' mode & creates and returns an object of StreamReader class that can read text from the file.

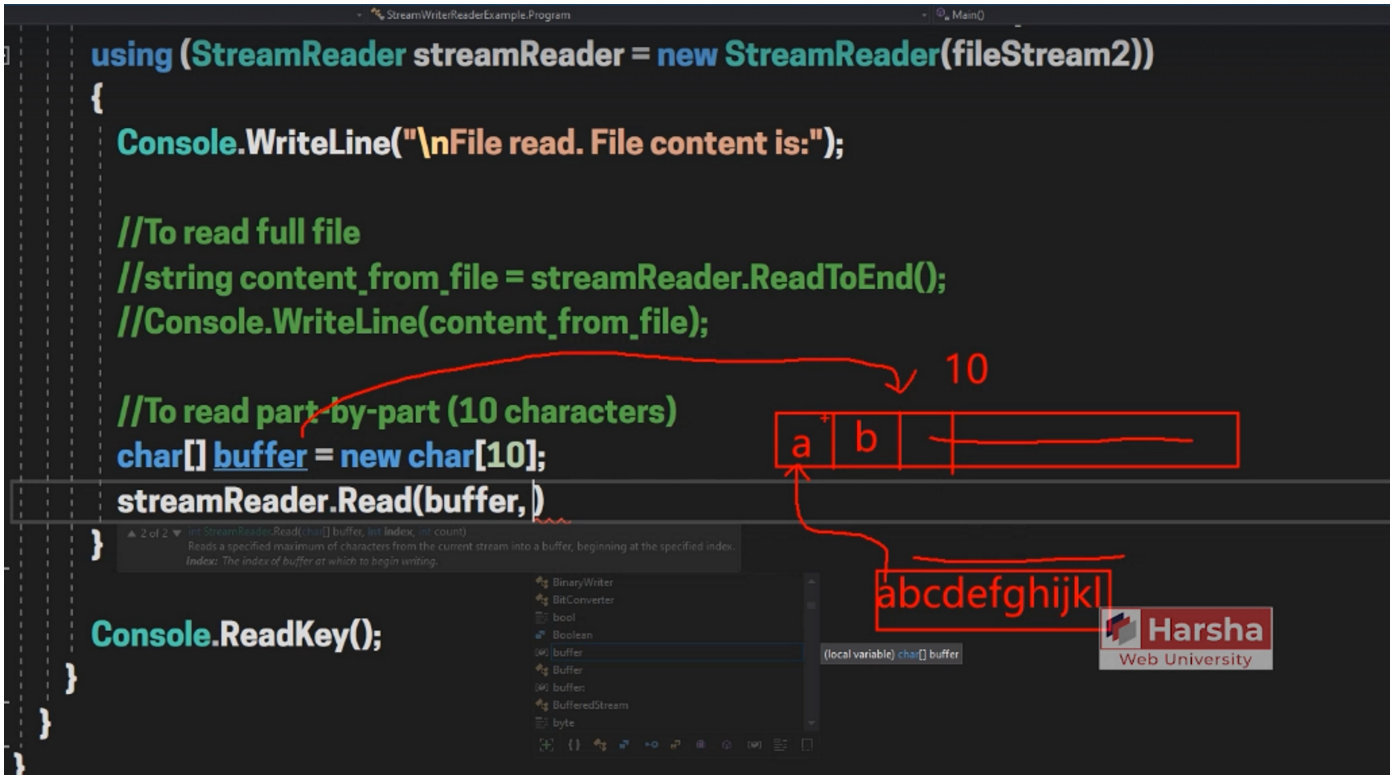

```
class Program
{
    0 references
    static void Main()
    {
        string filePath = @"c:\practice\europe.txt";
        FileInfo fileInfo = new FileInfo(filePath);
        //StreamWriter streamWriter = new StreamWriter(filePath);
        FileStream fileStream = new FileStream(filePath, FileMode.Create, FileAccess.Write);
        //StreamWriter streamWriter = new StreamWriter(fileStream);
        using (StreamWriter streamWriter = fileInfo.CreateText())
        {
            streamWriter.WriteLine("Hello");
            //streamWriter.Close(); //optional
        }

    }
}
```


this line is equivalent to 'StreamWriter streamWriter = new StreamWriter(fileStream);


Harsha

- **These are the 3 ways of creating StreamWriter Object.**


namespace System.IO
StreamWriter and StreamReader
HDD
StreamWriter class
write string
File
StreamReader class
read string
Harsha
Web University


using (StreamReader streamReader = new StreamReader(fileStream2))
{
Console.WriteLine("\nFile read. File content is:");
//To read full file
//string content_from_file = streamReader.ReadToEnd();
//Console.WriteLine(content_from_file);
//To read part-by-part (10 characters)
char[] buffer = new char[10];
streamReader.Read(buffer,)
}
Console.ReadKey();
}
}
}
10
a
b
abcdefghijkl
Harsha
Web University

StreamWriter

What

- Class that writes text data into the file.
- Internally uses FileStream.

How

StreamWriter

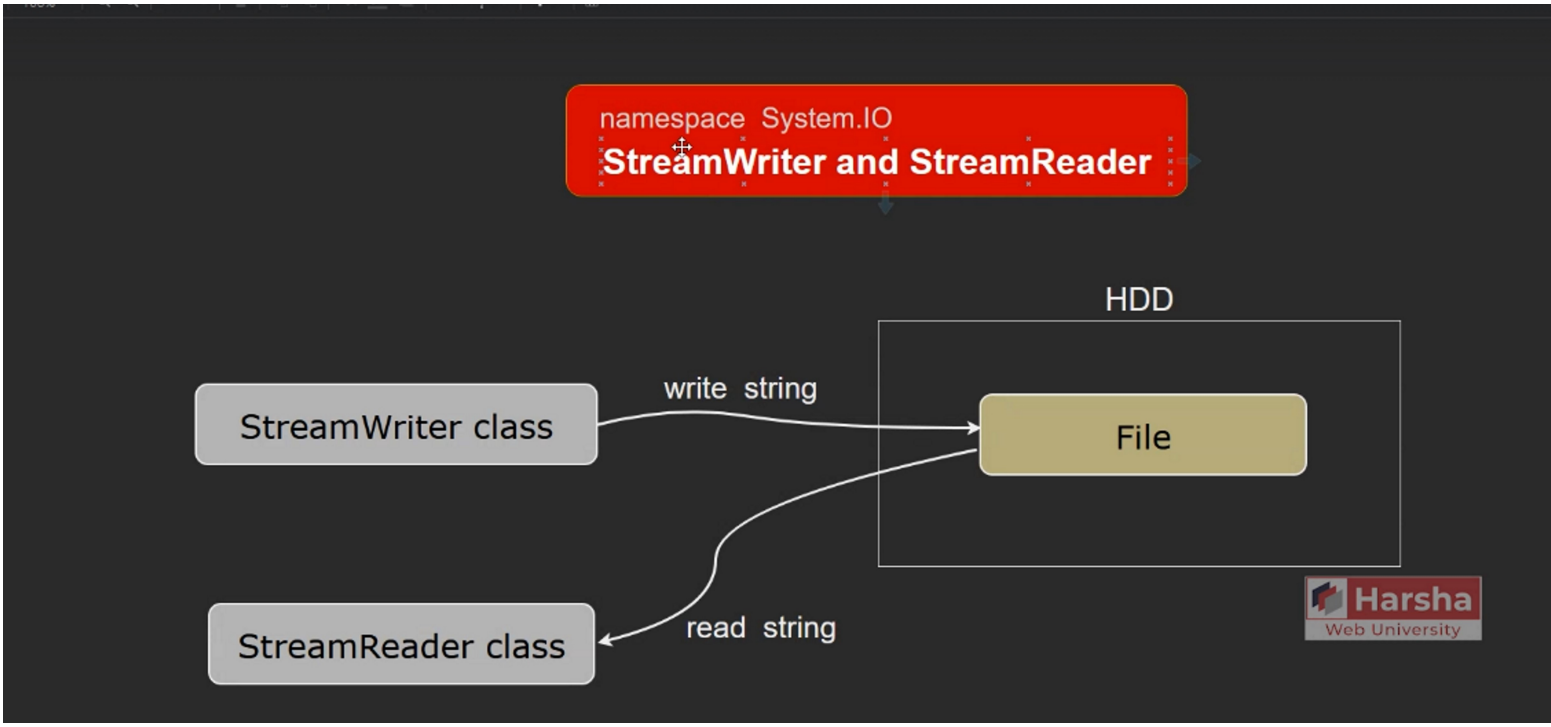
```
StreamWriter referenceVariable = new StreamWriter( "File Path" );
```

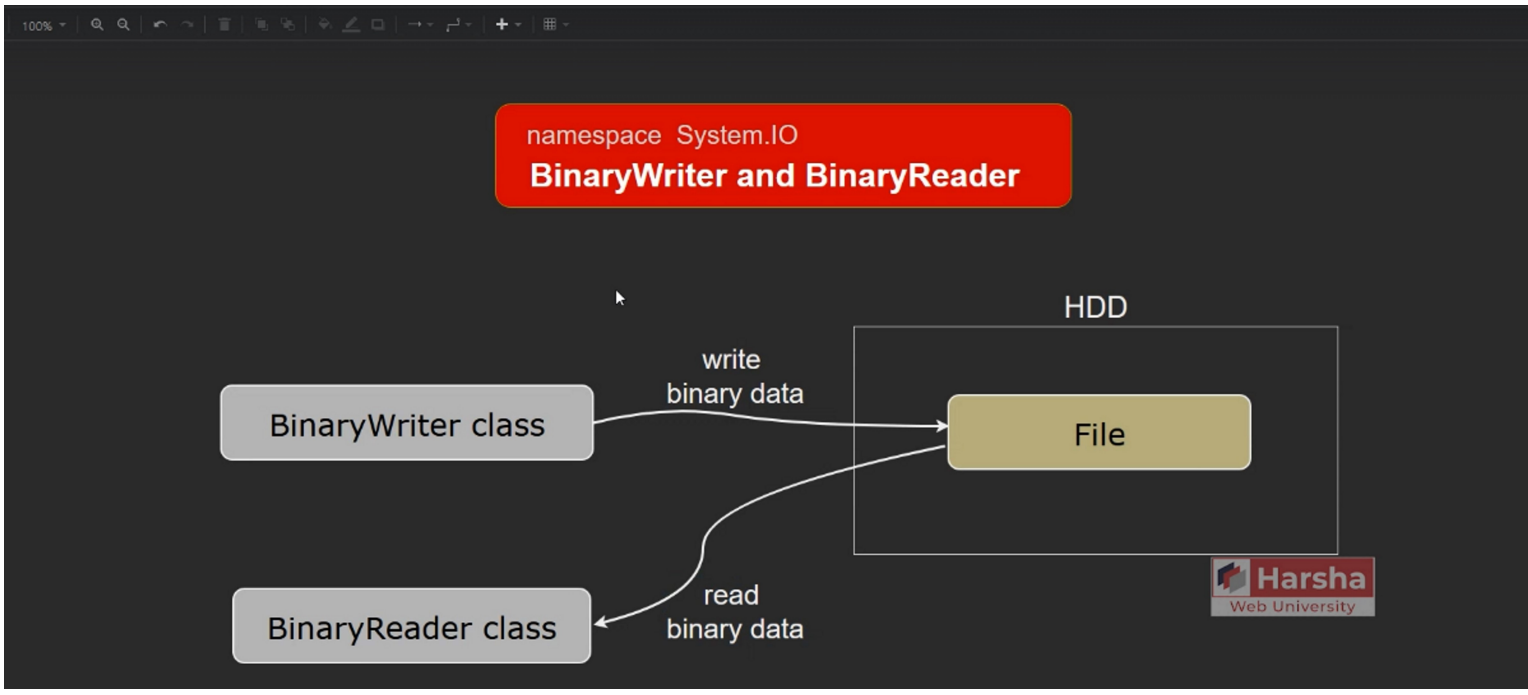
Constructors

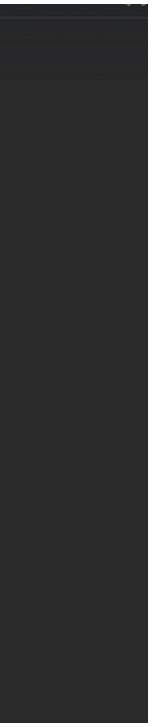
Constructor	Description
StreamWriter(FileStream stream)	It initializes the StreamWriter based on the specified FileStream.
StreamWriter(string path)	It initializes the StreamWriter for the specified file path.

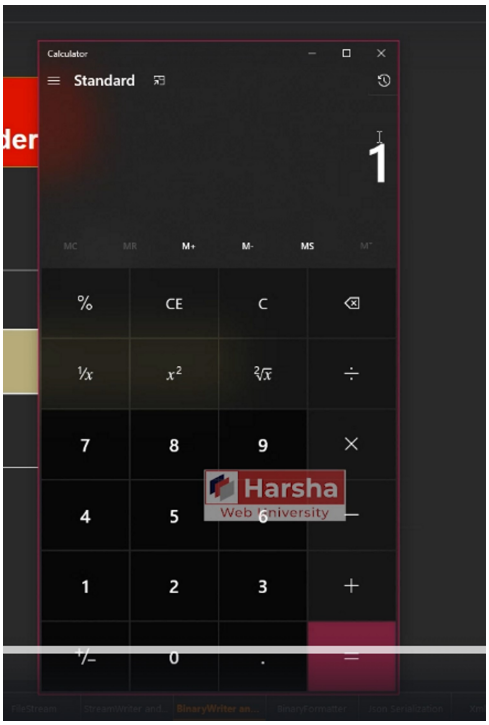
BinaryWriter and BinaryReader

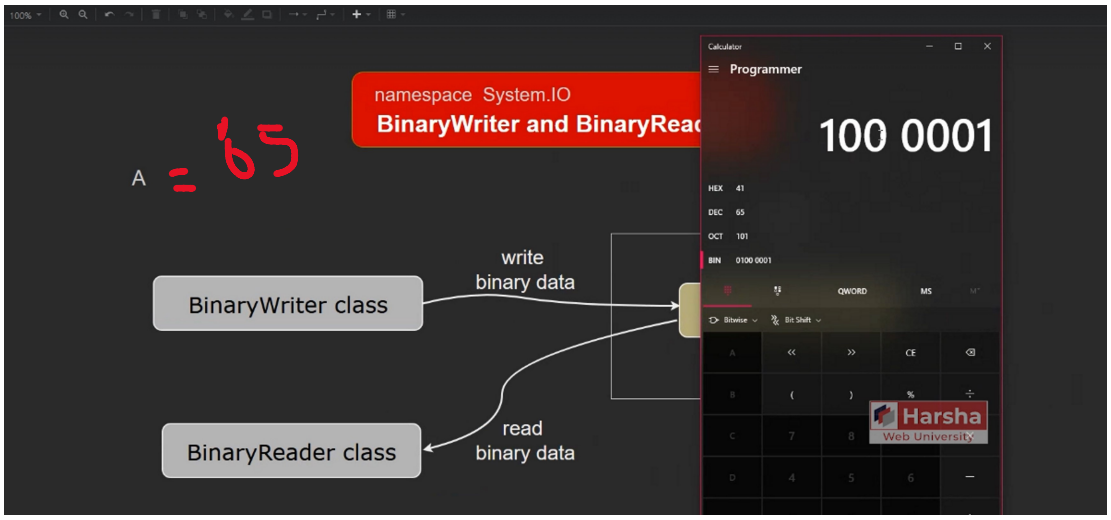
Part 1


namespace System.IO
StreamWriter and StreamReader
HDD
write string
StreamWriter class
File
read string
StreamReader class
Harsha
Web University


namespace System.IO
BinaryWriter and BinaryReader
HDD
write
binary data
BinaryWriter class
File
read
binary data
BinaryReader class
Harsha
Web University







You can also write information into the file by using binarywriter . so what is the difference between streamwriter an
binarywriter ? The stream writer can write the information only as a string even though you might be writing number
and other data types. They are first converted into string and finally that string internally
converted as a byte array and that byte array will be stored in the file but whereas the binary writer works in a little bi
different way for example you have a number that is for example 1

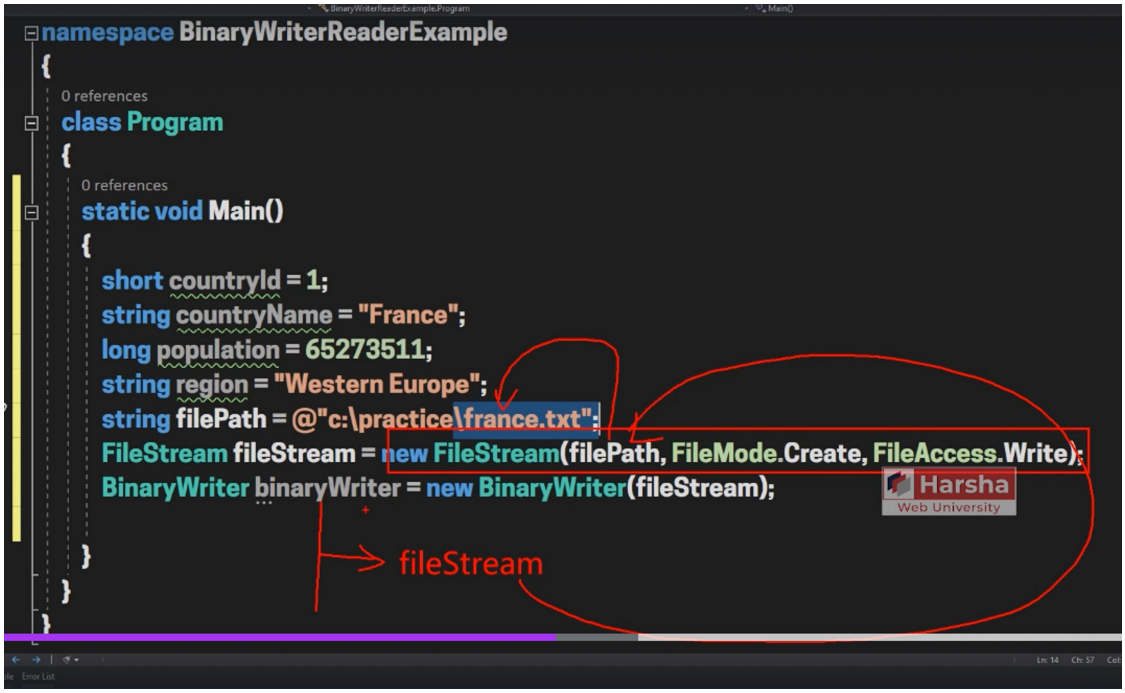
at first this number will be converted as a binary format that is zeros and ones format
already earlier we have understood about number systems and we know how do you convert the number from decima
number system into binary number system in the calculator in windows.

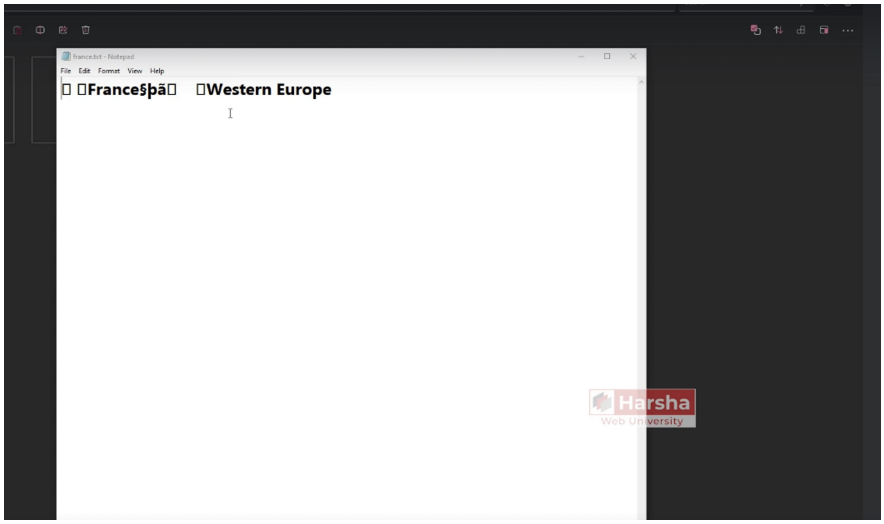
Open the Windows Calculator and select programmer and once you selected programmer if you give a number here f
example 1 the same will be printed in all the other number systems .So the decimal number one is equal to the
binary number zero zero zero one for example we have to write digit one by using the binarywriter at first this will be

converted into binary format and that binary numbers are written into the file that's the point about binarywriter

But what about the characters and strings? For example as a part of this string you have a character called a alphabet what is the equivalent ascii code of the character 'A'

For example if it is uppercase a it will be equivalent to 65 right so that 65 will be converted into binary format and it is 0 1 0 0 0 0 000 1 so this binary number will be stored in the file in that way the binder rater works and in the same way the binary reader also works means the existing information which is in binary format will be read and eventually you can convert the same into the dot net data type such as carr string int bool etc

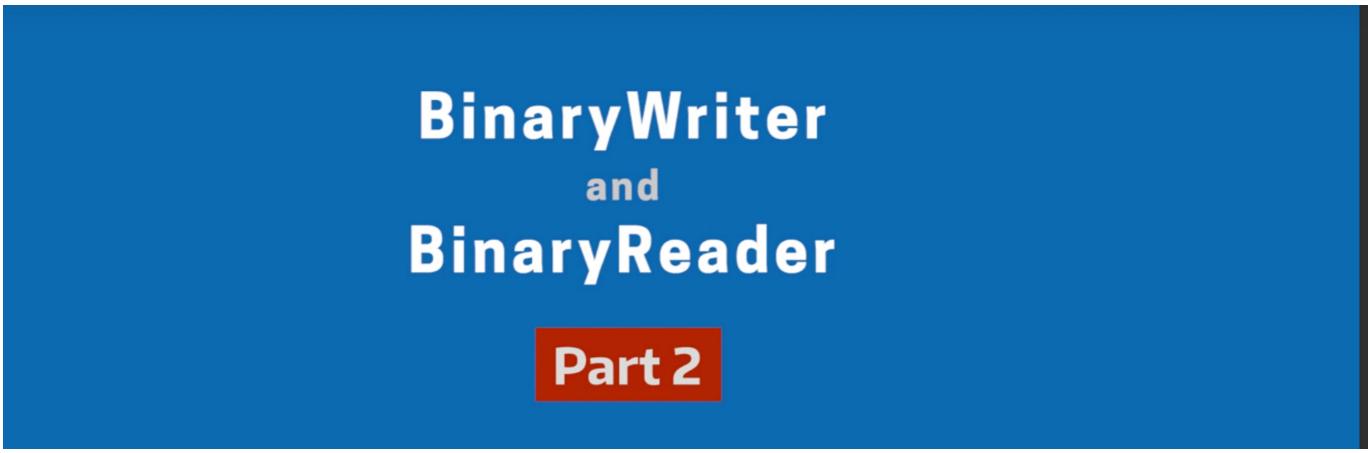




a

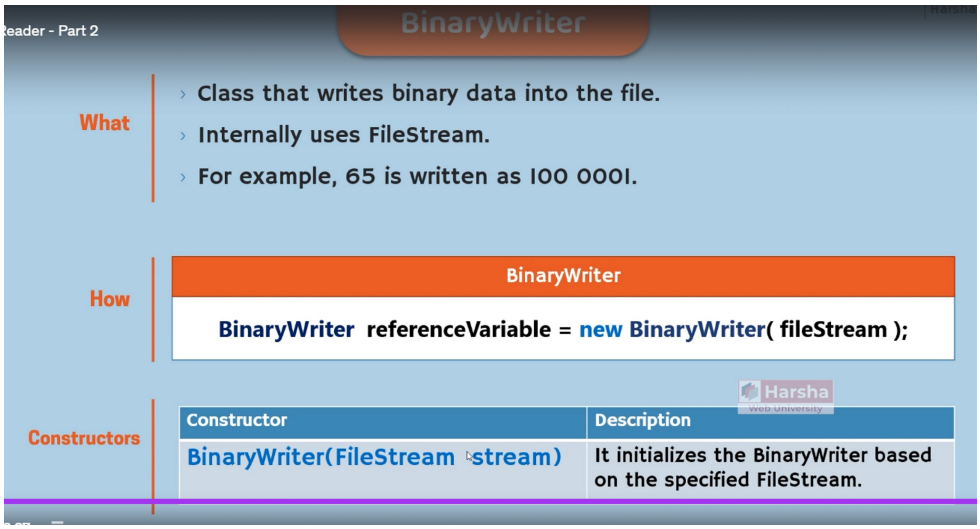
es

All the information are stored in binary format. But when you try to open the file in notepad, notepad try to convert the binary file to string format. But it can not convert the number to string.



Let us see how do you read existing binary file by using this Binary Reader
if you write information through Binary Writer then only you are able to use the Binary Reader in order to read the same.

It's general opinion that, binary file written by binary writer takes less amount of memory.

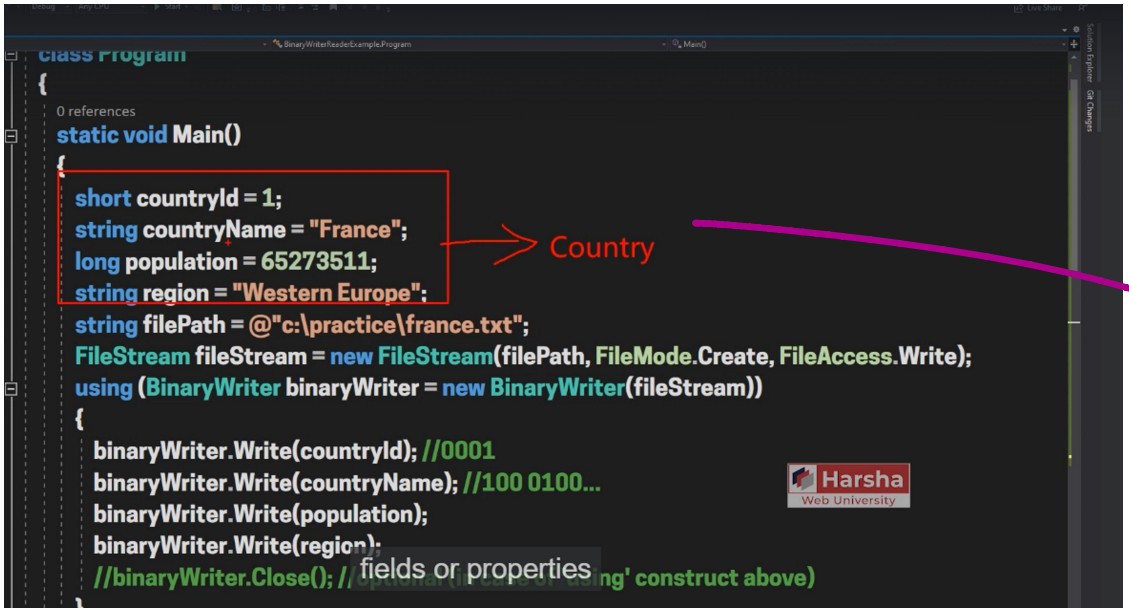


eader - Part 2

BinaryWriter

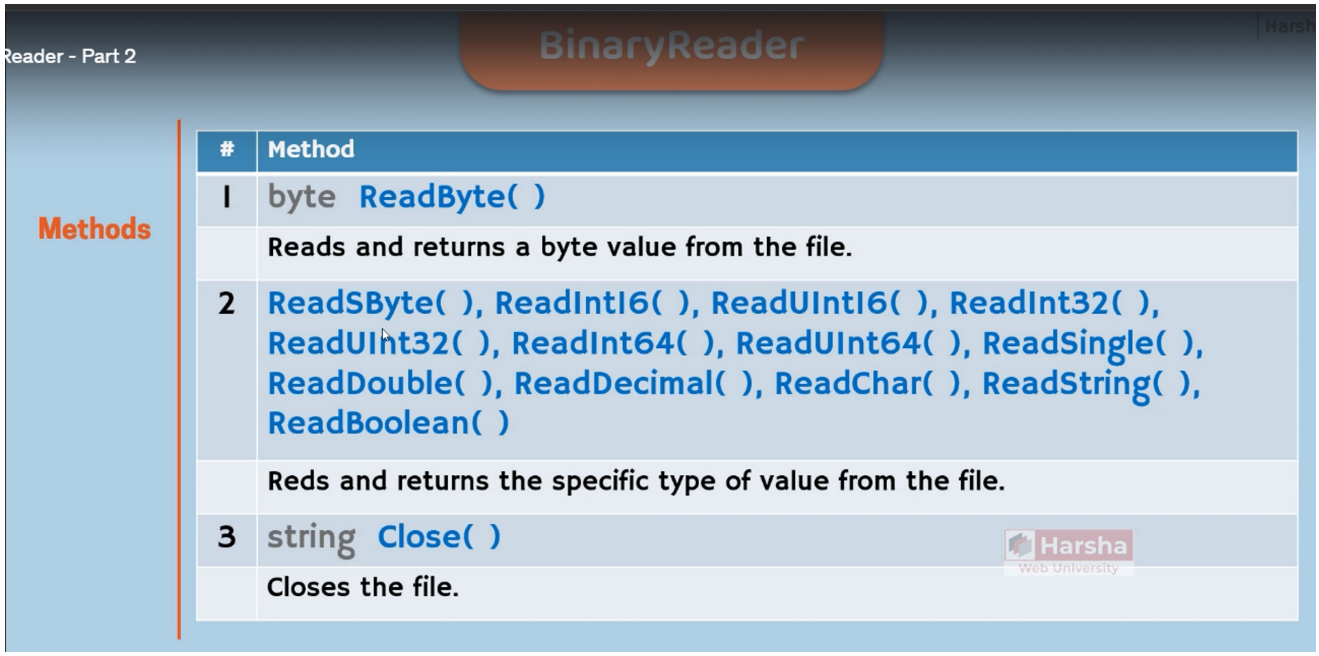
Methods

#	Method
1	void Write(...)
	Writes any type of value (only primitive types / string) into the file.
2	void Close( )
	Close the file.



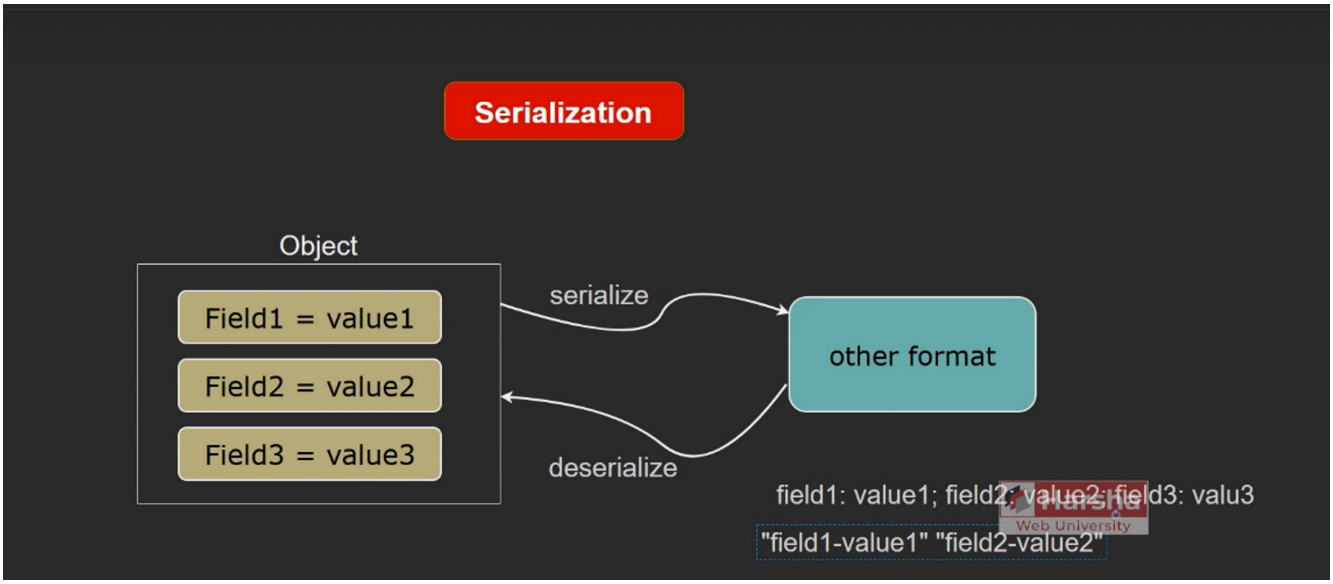
"If we had used an object, we wouldn't have needed to write multiple read and write methods."

This is called as Serialization.

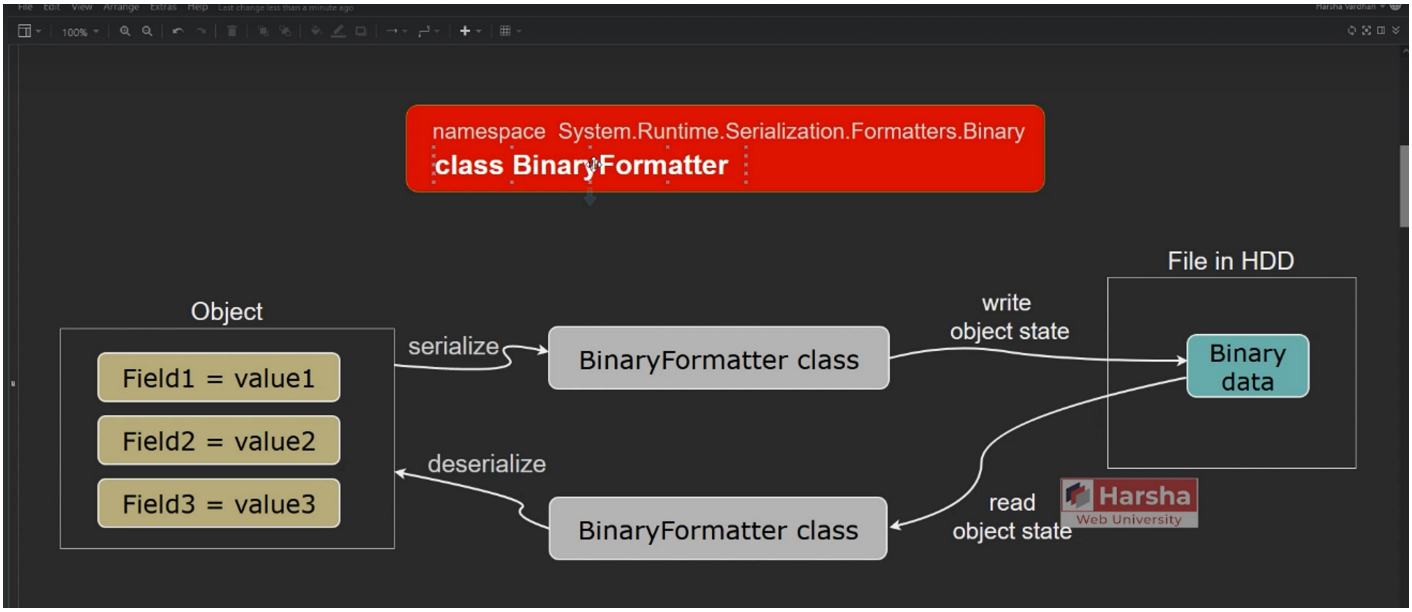


In c-sharp you have a concept of Serialization that means converting an object from one state to another state means from one format to another format. suppose for example you have converted your object from heap into a diffe maybe for example binary format or xml format or something else so that process of conversion is called a serializati

erent format
on



In c sharp mainly and importantly we have three predefined serializations that is binary serialization json serialization and xml serialization. In this lecture let us focus on binary serialization



So in this case through serialization concept you are serializing a specific state of the object into binary formatso that it ca saved in a file and later maybe from same application or from a different application you can rewrite means get back the s data into the object in this way two different applications can be interacted each other

for example you have created an application where you have an object with some fields with values
for example there is a customer object with customer id name email etc so you have serialized that object
and you have saved that serialized data in a file and you have given that file to me

and i can deserialize myself right so in my application i can read the content of your file

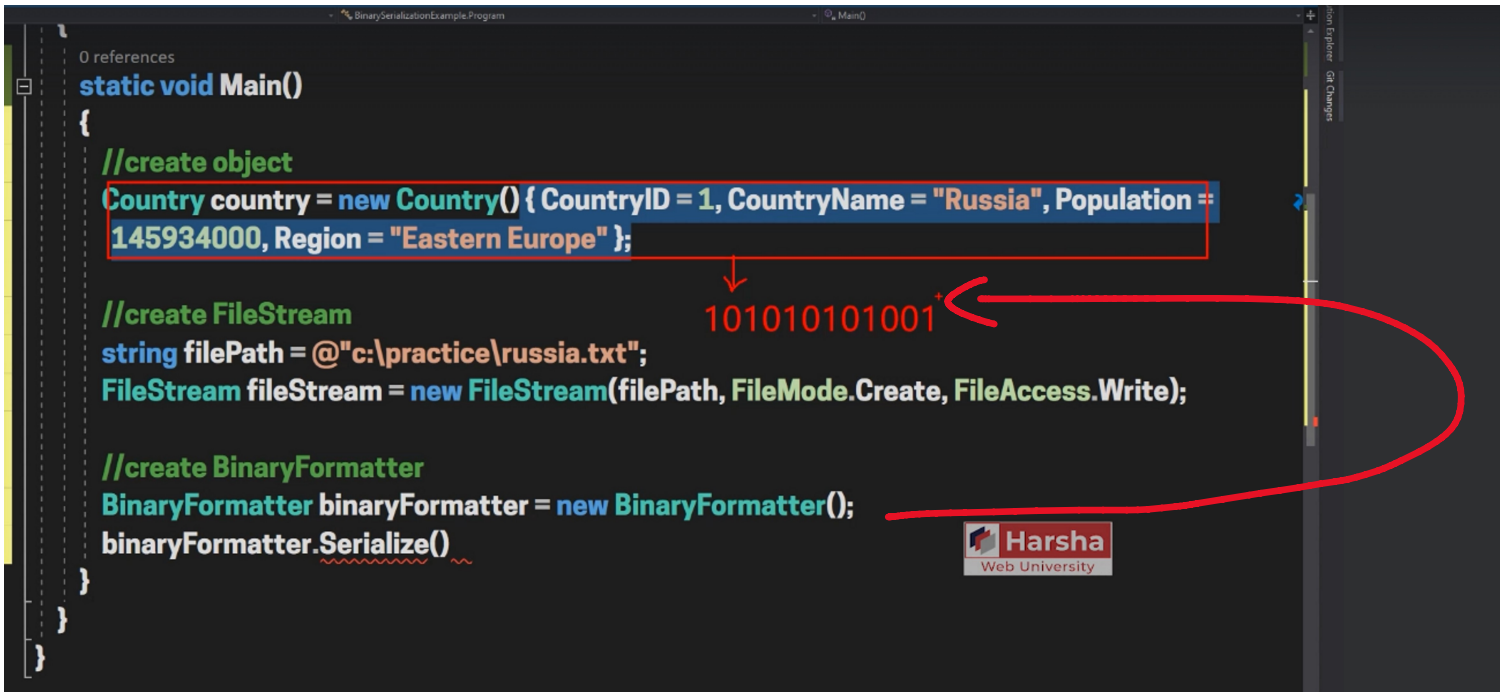
n be
ame

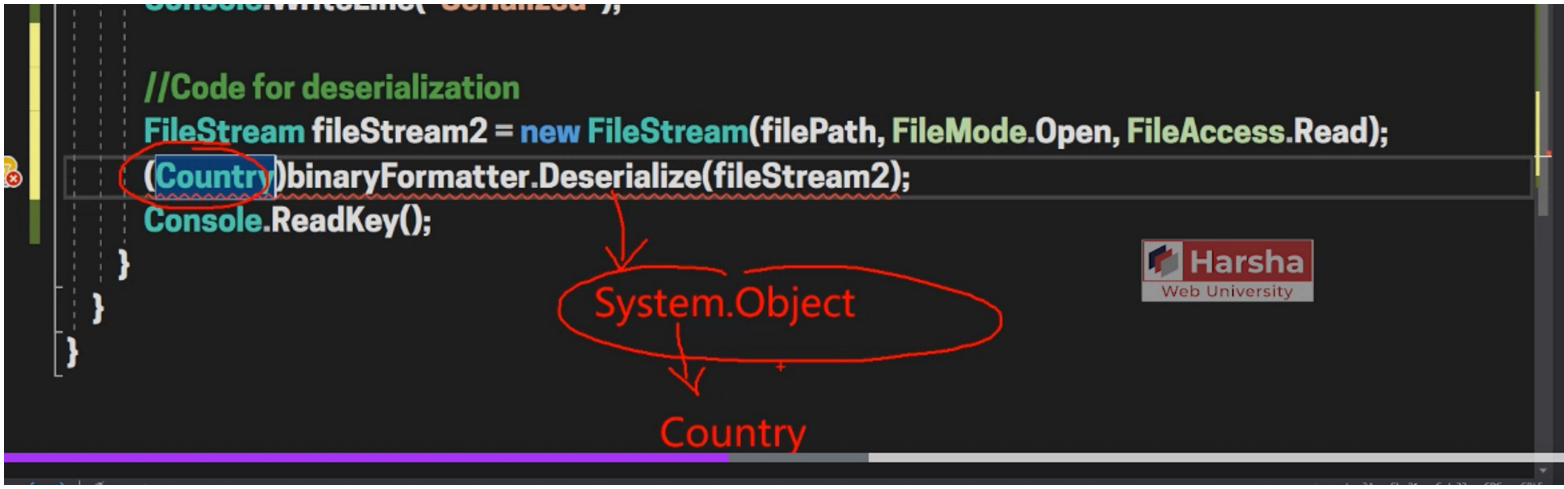
so that i can load this exactly same data into another object in my application
in that way your application can send the data to my application
not directly but through a file.

so for saving the information permanently in a file otherwise communicating the data from one application to another application we can use this serialization concept in the real-world applications

of course most of the cases in the real life we use databases instead so you can consider that.

This might be a or technique where you cannot use the databases





t 2

BinaryFormatter

What

- Class that serializes (converts) an object-state into binary format into and stores in binary file.
- Serialization is a process of converting an object from one format to another format.
- It can also read existing object-state from the binary file.
- Full Path: System.Runtime.Serialization.Formatters.Binary.BinaryFormatter

How

BinaryFormatter

```
BinaryFormatter referenceVariable = new BinaryFormatter( );
```

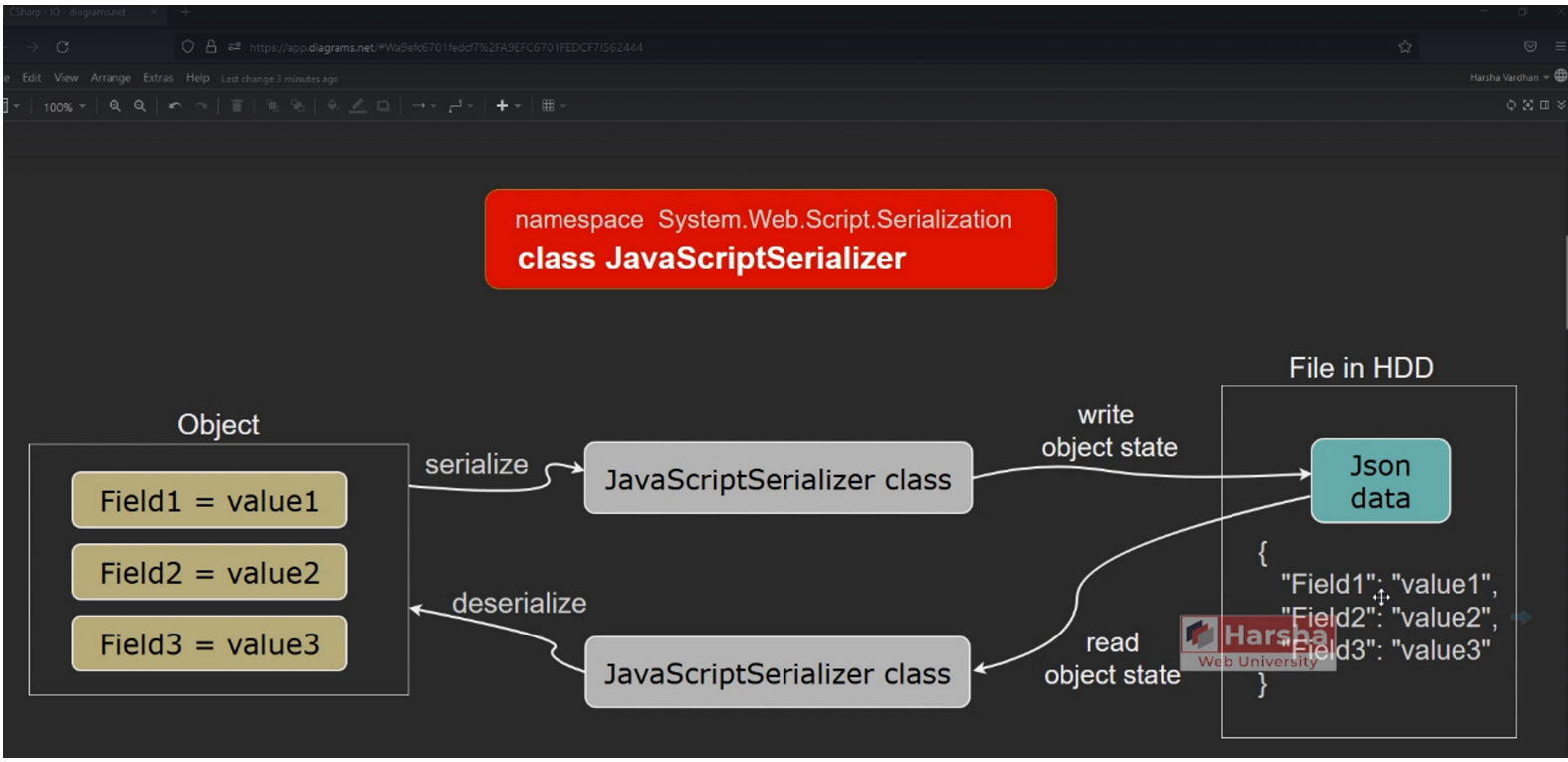
t 2

BinaryFormatter

Methods

#	Method
1	void Serialize(FileStream FileStream, object data)
	Converts the object-state into the binary file, using the specified FileStream (with Write access).
2	object Deserialize(FileStream FileStream)
	Reads the existing object-state from the binary file, using the specified FileStream (with Read access).

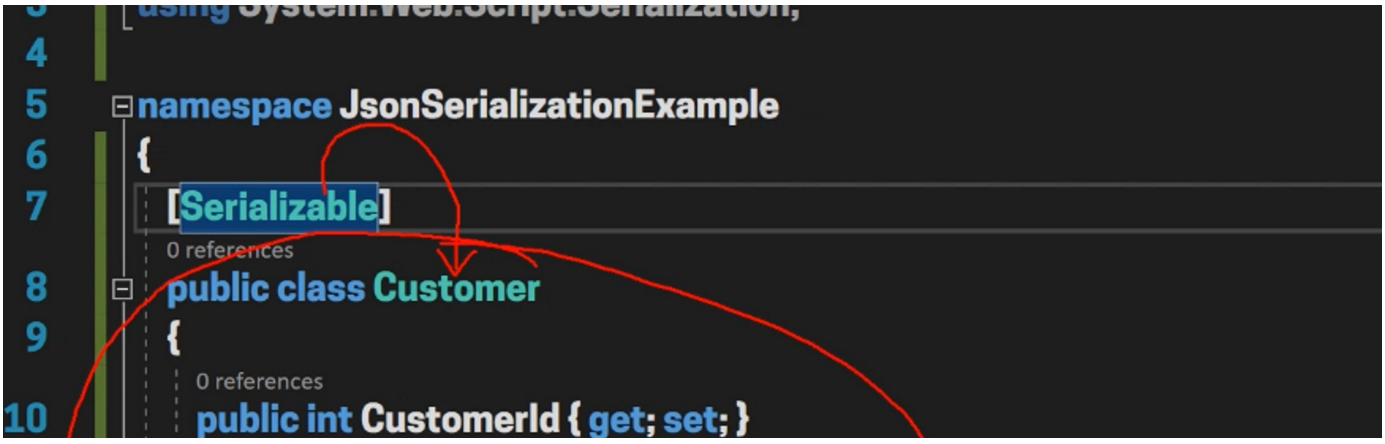
Json Serialization



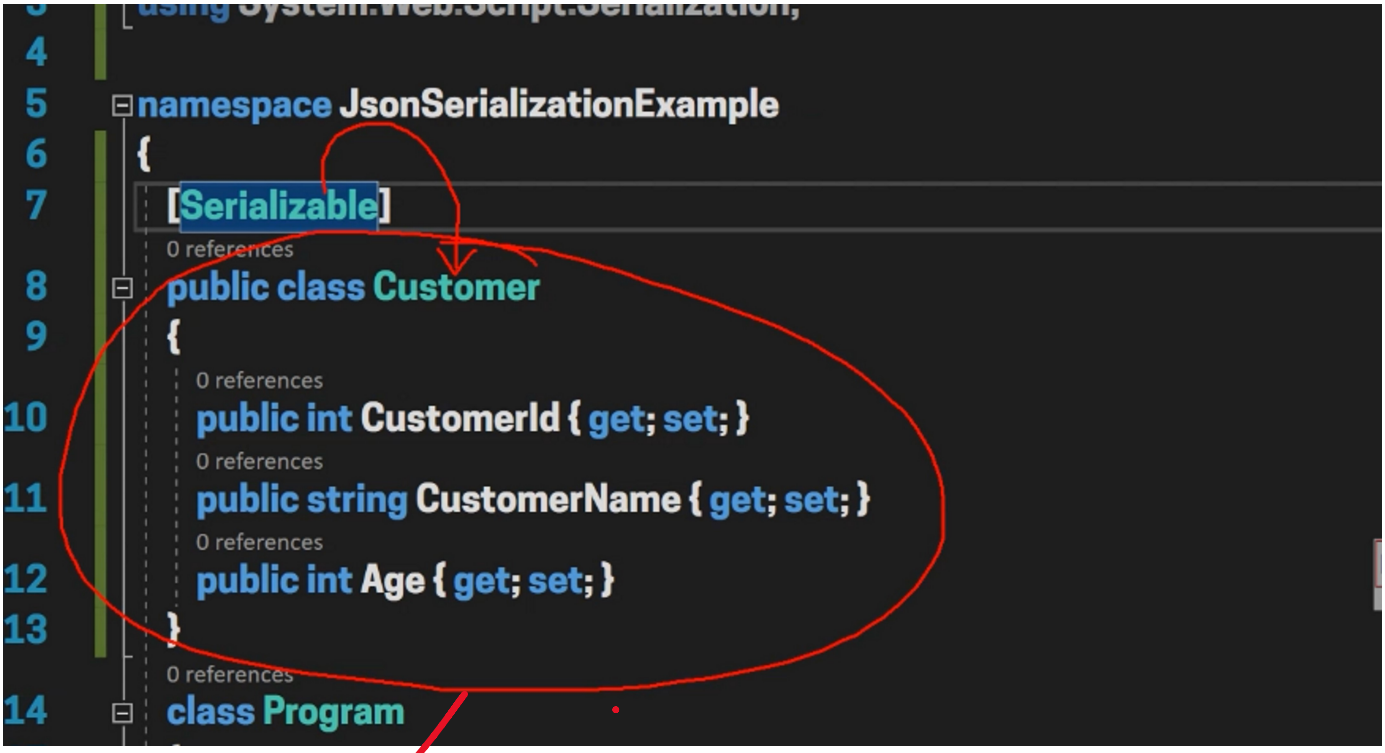
We have understood the purpose of the serialization and we have performed the binary serialization in the last lecture but the limitation of this binary serialization is that you can de-serialize means get back the same information into any dot.net application only. you cannot read the same information from other language applications

that means for example you cannot deserialize the same information from an application that was developed in java or python so you can serialize and deserialize only from dotnet application only.
that is the problem with the binary serialization.

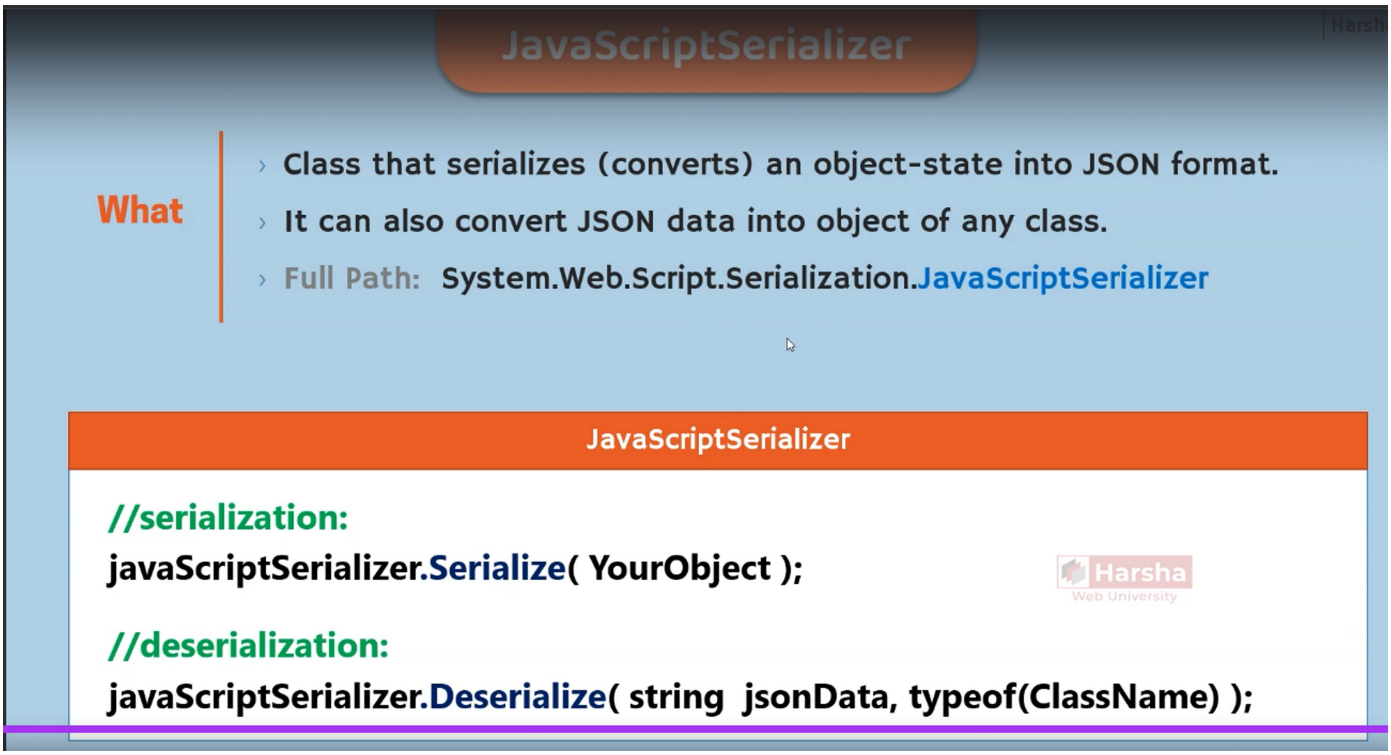
But in case if you want to write the content in a universally acceptable and understandable format which can be read from any type of application from a java application python application of course from the c-sharp application or from any other language application you will be able to read the information from application, if this is the requirement for you then you can think about json serialization which can be implemented by javas serializer.



any other
script

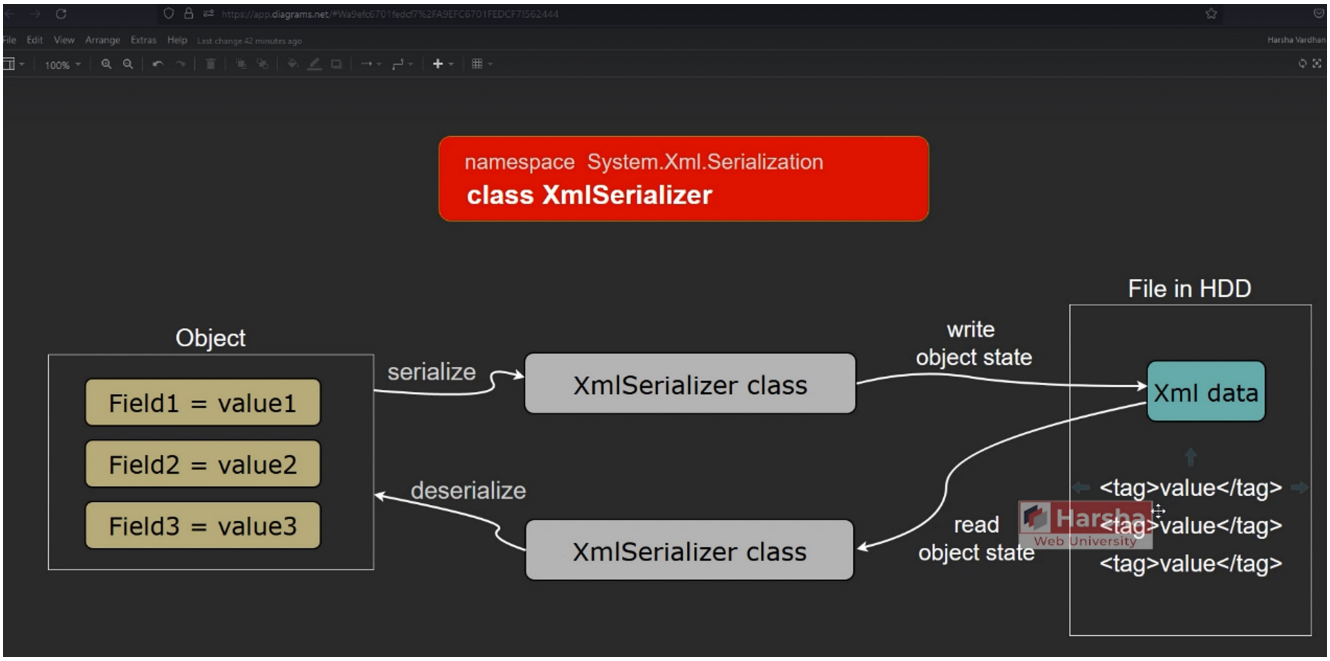


It indicates to C# that, this class objects can be converted into any other format such as binary format or JSON format or XML format or any other different format.



For example you can serialize in c sharp application here and Deserialize the same
in a java application or javascript application or something else and in the same way it can be reversed
for example some other java developer has serialized the information in java application
and you can perform deserialization here so across applications you can transfer
the data that is the advantage or purpose of this aim

Not only data sharing but also you can save the data permanently in the form of
a file with the help of this deserialization concept.



XmlSerializer

Harsha

What

- Class that serializes (converts) an object-state into XML format and stores in XML file.
- It can also read existing object-state from the XML file.
- Full Path: System.Xml.Serialization.XmlSerializer

How

XmlSerializer

```
XmlSerializer referenceVariable = new XmlSerializer( typeof(ClassName) );
```

Harsha
Web University

XmlSerializer

Methods

#	Method
1	void Serialize(FileStream FileStream, object data)
	Converts the object-state into the xml file, using the specified FileStream (with Write access).
2	object Deserialize(FileStream fileStream)
	Reads the existing object-state from the xml file, using the specified FileStream (with Read access).