**File IO – Part-01**

* **Agenda:**

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

* **File:**

File f = new File(“abc.txt”);

This line won’t create any physical file. First it will check is there any physical file named with abc.txt is available or not. If it is available then “f” simply refers that file.

If it’s not available then we are just creating Java File object represent the name “abc.txt”.

File f = new File(“abc.txt”);

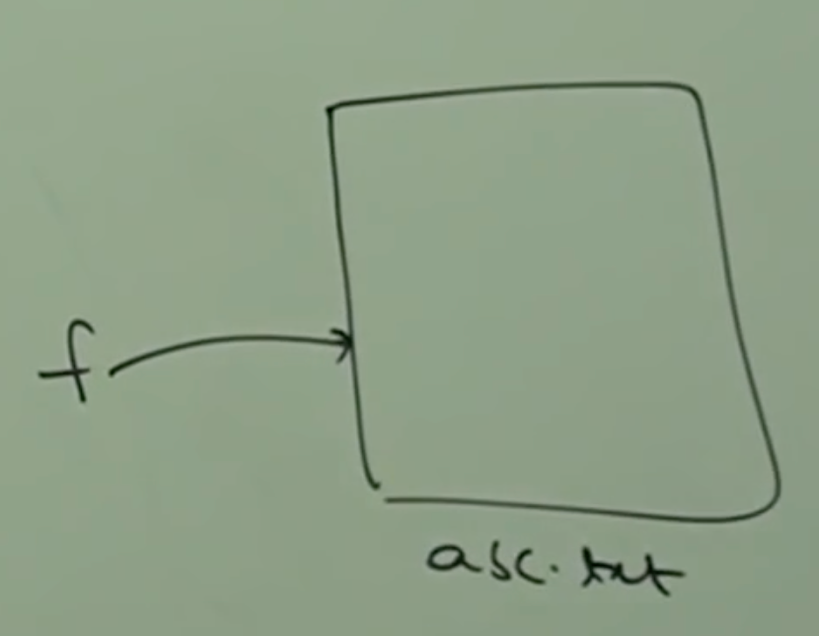
System.out.println(f.exists()); // false

f.createNewFile(“abc.txt”);

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

1st Run Output: false, true

2nd Run Output: true, true



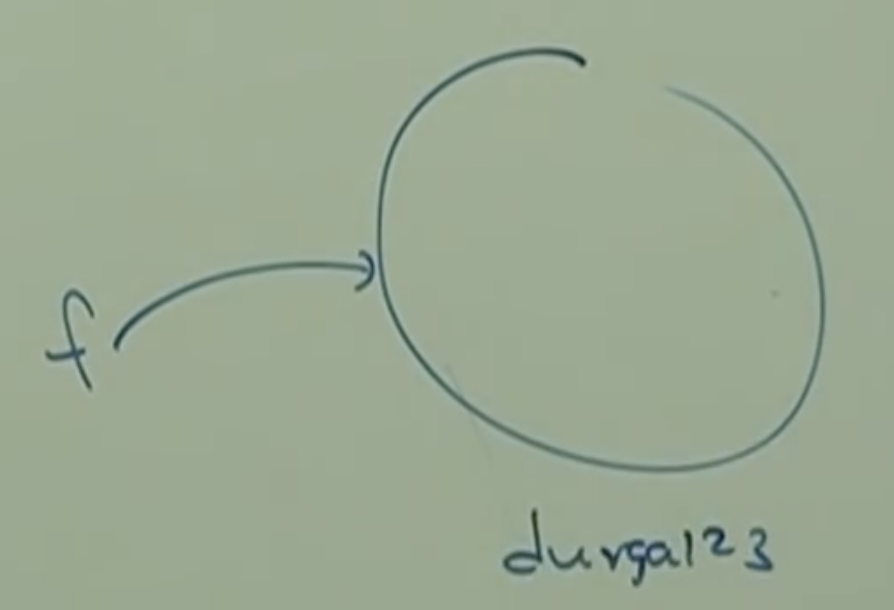
We can use Java file object to represent directory also.

File f = new File(“durga123”);

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

f.mkdir();

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



Note:

In Unix everything is treated as a “File”.

Java file IO concept is implemented based on Unix operating system. Hence, Java File object can be used to represent both files and directories.

* **File Class Constructors:**

File f = new File(String name);

Creates a Java File object to represent name of the file or directory in current working directory.

File f = new File(String subDirName, String name);

Creates a Java File object to represent name of the file or directory present in specified sub-directory.

File f = new File(File subdir, String name);

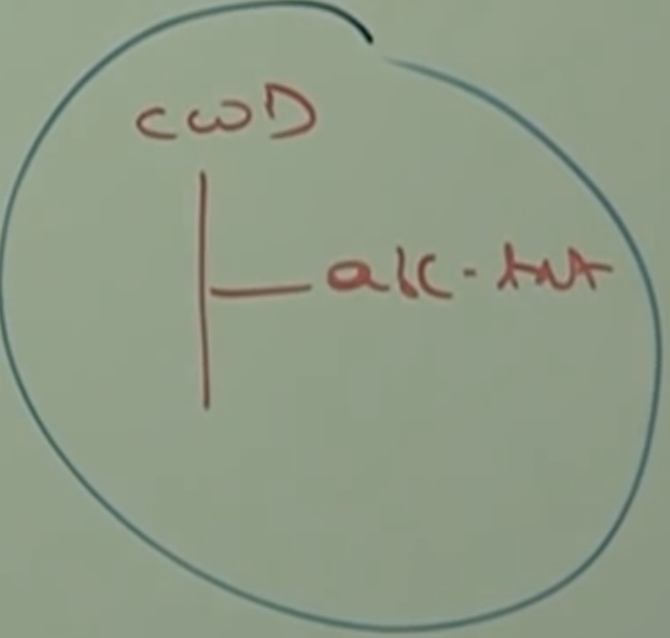
Same as above constructor.

**Example\_01:**

Write code to create a file named with “abc.txt” in current working directory.

File f = new File(“abc.txt”);

f.createNewFile();



**Example\_02:**

Write a code to create a file (folder) named with “durga123” in the current working directory and create a file named with “demo.txt” in that directory.

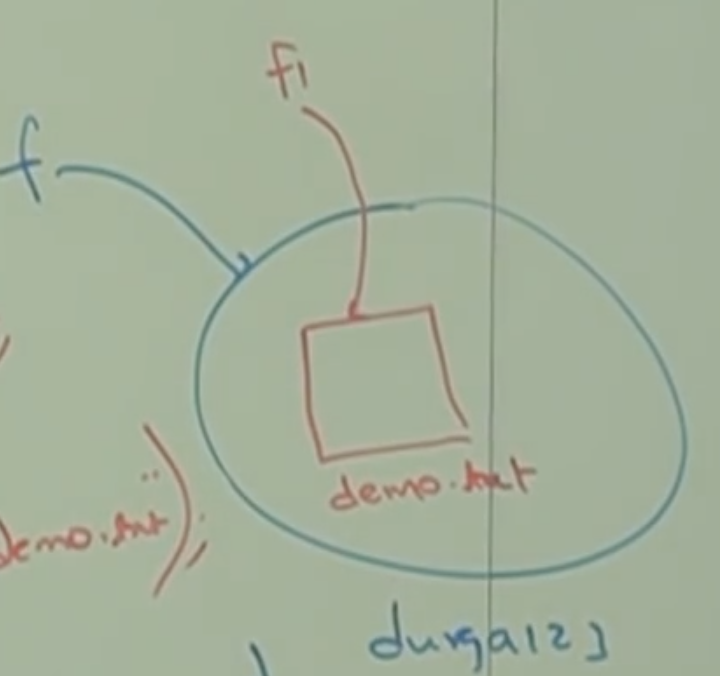
File f = new File(“durga123”);

f.mkdir();

File f2 = new File(“durga123”, “demo.txt”);

File f2 = new File(f, “demo.txt”);

f2.createNewFile();

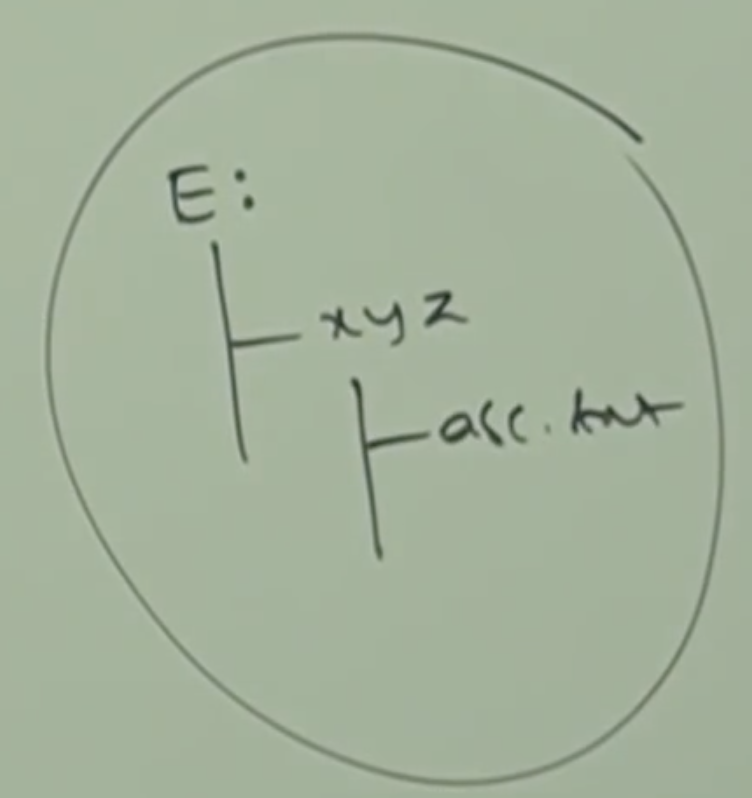


**Example\_03:**

Write code to create a file named with “abc.txt” in “E:\xyz” folder.

File f = new File(“E:\\xyz”, “abc.txt”);

f.createNewFile();



Assume that “E:\\xyz” folder already available in our system.

* **Important Methods In File Class:**

boolean exists();

It returns true, if the specified file or directory is available, else it will return false.

boolean createNewFile();

First this method will check whether this specified file is already or not, if it is already available then this method returns false without creating any physical file.

If the file is not already available then this method will create a new file and returns true.

boolean mkdir();

boolean isFile();

Returns true, if the specified file object pointing to physical file.

boolean isDirectory();

String[] list();

This method returns the names of all files and sub-directories present in specified directory.

long length:

Returns number of characters present in specified file.

boolean delete();

To delete specified file or directory.

Example\_01:

Write a program to display the names of all files and directories present in C:\\durga\_classes

class Test{

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

int count = 0;

File f = new File(“E:\\durga\_classes”);

String[] s = f.list();

for(String name: s){

count++;

System.out.println(name);

}

System.out.println(“Total Number of directories and files:”+count);

}

}

Program to display only the files:

class Test{

public static void main(String[] args){

File f = new File(“E:\\durga\_classes”);

String[] s = f.list();

for(String name: s){

File f1 = new File(f, name);

if(f1.isFile()){

count++;

System.out.println(s);

}

}

System.out.println(“Total Number of files:”+count);

}

}

Program to display only the directories:

class Test{

public static void main(String[] args){

File f = new File(“E:\\durga\_classes”);

String[] s = f.list();

for(String name: s){

File f1 = new File(f, name);

if(f1.isDirectory()){

count++;

System.out.println(s);

}

}

System.out.println(“Total Number of Directories is:”+count);

}

}

