# Contents java.io.File class

[Contents java.io.File class 1](#_Toc403409031)

[Uses of Java.io.FileClass: 1](#_Toc403409032)

[Constructors of java.io.File class: 2](#_Toc403409033)

[Creating Files/Folders on HD 2](#_Toc403409034)

[Creating File 2](#_Toc403409035)

[Example Program to Create a File 3](#_Toc403409036)

[Creating an empty directory 4](#_Toc403409037)

[Example Program to Create Directory 4](#_Toc403409038)

[Deleting an empty directory and any File 5](#_Toc403409039)

[How to delete any file or empty directory? 5](#_Toc403409040)

[How to get size of the file? 6](#_Toc403409041)

[File or Folder existence Validation 6](#_Toc403409042)

[Write a program to check whether a file or folder is existing or not? 7](#_Toc403409043)

[Files and File Dates 8](#_Toc403409044)

[About java.util.Date() class 8](#_Toc403409045)

[Write a program to print pc date on monitor 8](#_Toc403409046)

[Write a program to print pc date continuously 8](#_Toc403409047)

[How to get last modified date of a file? 9](#_Toc403409048)

[Program to get Last Modified Date File / Folder 9](#_Toc403409049)

[Renaming File or Folder: 10](#_Toc403409050)

[Program to Rename file / Folder 10](#_Toc403409051)

[Listing Folder Contents: 11](#_Toc403409052)

[How to get contents of the folder? 11](#_Toc403409053)

[Write program to get the contents of the folder: 11](#_Toc403409054)

[Write a program to find size of a directory? 12](#_Toc403409055)

# Uses of Java.io.FileClass:

By using java.io.File class we can do the following operations:

1. We can create an empty file.
2. We can create an empty folder.
3. We can delete any file.
4. We can delete empty folder.
5. We can rename an existing file.
6. We can rename an existing directory.
7. We can check whether a file/folder is existing or not.
8. We can list the files in folder.
9. We can get the size of the file.
10. We can create temp files.
11. We can get file creation date, last modified date etc

# Constructors of java.io.File class:

* File(URI uri);
* File(String pathName);
* File(String parent, String child);
* File(File parent, String child);

File class constructors are used to create File class objects in RAM

File class constructor are not used to create files/folders on the HD.

# Creating Files/Folders on HD

## Creating File

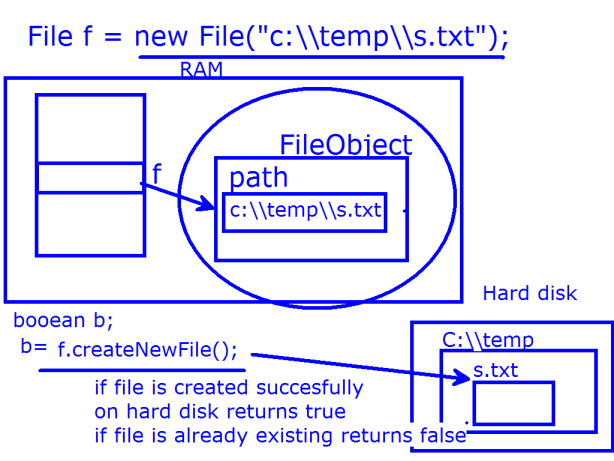
By using the createNewFile() method of java.io.File class we can create an empty File on HD.

Prototype:

*boolean createNewFile()throws IOException*

If a file does not exist on HD with the name given in constructor then this method will create empty new file.

* If file does not exist on HD then this method creates new file and returns true.
* If file already exists on HD then this method returns false.
* If file creation is failed then this method throws IOException.



### Example Program to Create a File

CreateNewFileDemo.java

import java.io.File;

import java.io.IOException;

import java.util.Scanner;

public class CreateNewFileDemo{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("Enter file name including path to create:");

String fn = sc.nextLine();

//file object is created in RAM

File file = new File(fn);

try{

//this method call creates file

boolean b = file.createNewFile();

if(b==true)

System.out.println("file is created successfuly");

else

System.out.println("file is already existing");

}catch(IOException ioe){

System.out.println("Error!File creation not possible bcoz :"+ioe.getMessage());

}

}

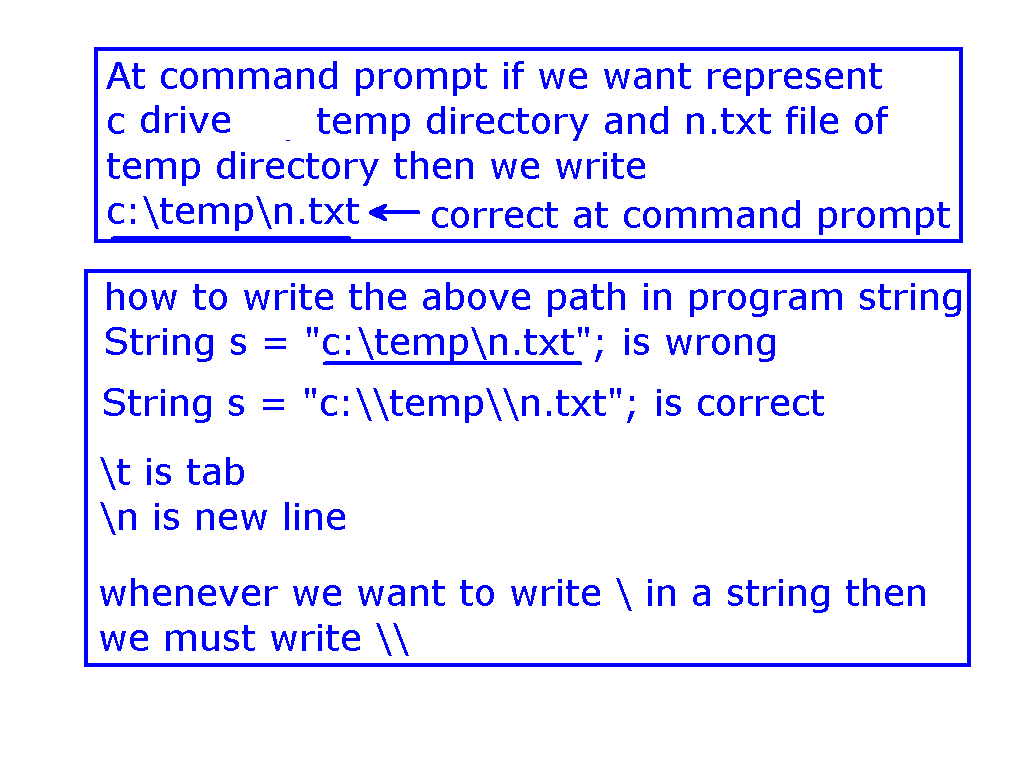
}

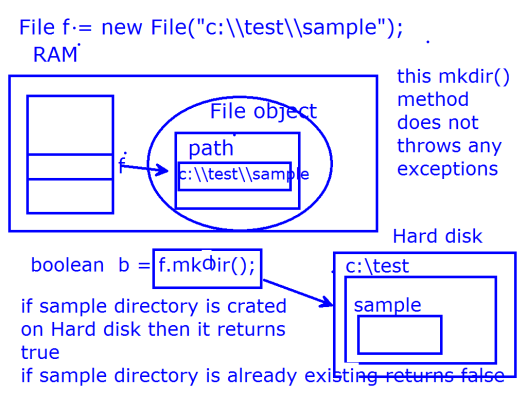
## Creating an empty directory

To create a directory we use the following method of *java.io.File* class

*boolean mkdir();*

* If directory is not existing then this method creates the directory and returns true.
* If directory is already existing then this method returns false.





### Example Program to Create Directory

CreateDirDemo.java

import java.io.File;

import java.util.Scanner;

public class CreateDirDemo{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("Enter dir name including path to create:");

String fn = sc.nextLine();

//file object is created in RAM

File file = new File(fn);

//file object is created on HD

boolean b = file.mkdir();

if(b==true)

System.out.println("dir is created successfuly");

else

System.out.println("dir is already existing");

}

}

# Deleting an empty directory and any File

## How to delete any file or empty directory?

ans: By using delete() method of java.io.File class

boolean delete();

This method returns true if the file or folder is delete successfully.

This method return false if the folder is not empty.

**DeleteDemo.java**

import java.io.File;

import java.util.Scanner;

public class DeleteDemo{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("Enter file/dir name including path to delete:");

String fn = sc.nextLine(); //file object is created in RAM

File file = new File(fn);

boolean b = file.delete();

if(b==true)

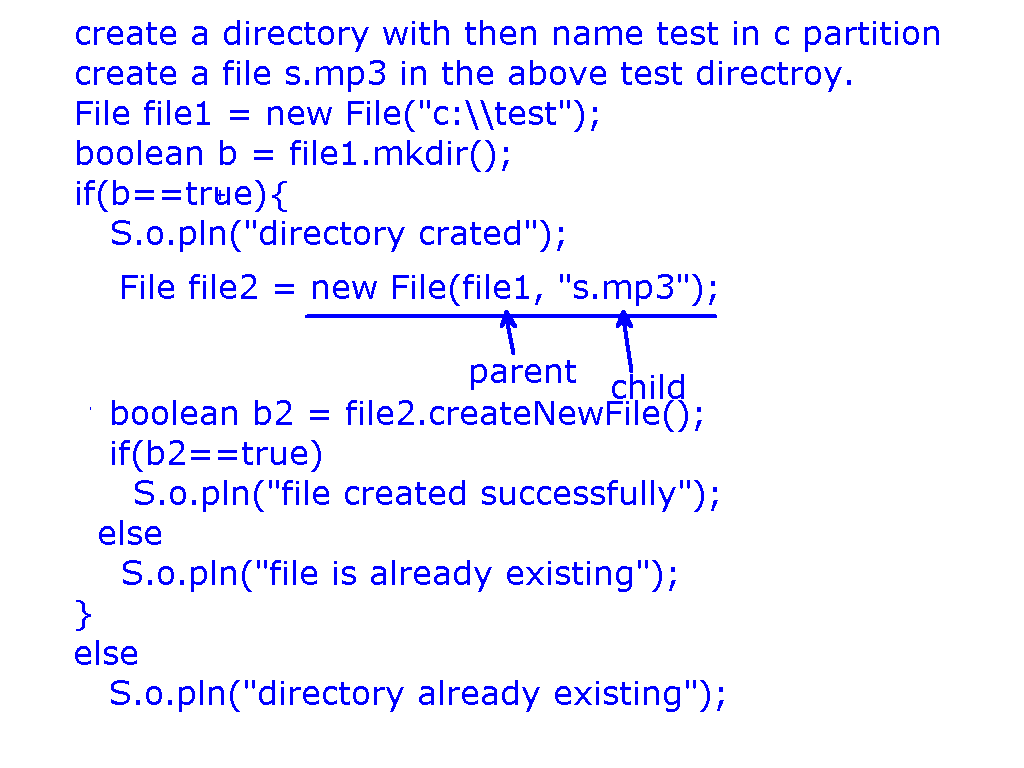
System.out.println("file/dir is deleted successfuly");

else

System.out.println("dir is not empty");

}

}



## How to get size of the file?

ans: Using long length() method of java.io.File class this method returns the size in bytes

**FileSizeDemo.java**

import java.util.Scanner;

import java.io.File;

public class FileSizeDemo{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("enter the name of the file including path:");

String fileName = sc.nextLine();

File file = new File(fileName);

long bytes = file.length();

double KB = bytes/1024.0;

double MB = KB/1024.0;

System.out.println("file size in bytes="+bytes);

System.out.println("file size in kilo bytes="+KB);

System.out.println("file size in mega bytes="+MB);

}

}

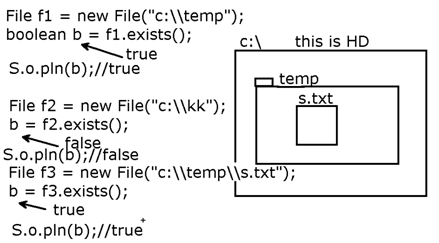
# File or Folder existence Validation

How can we find whether a file or folder is existing or not?

ans: boolean exists();

If the file or folder is existing on HD then this method returns true.

If the file or folder with the given name does not exist on HD then this method return false.



How to check that the file object is pointing a file on HD?

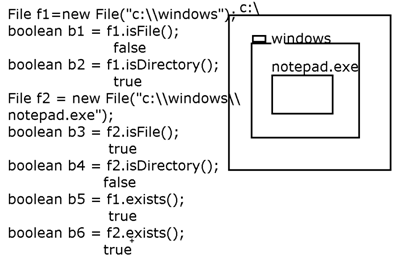
ans: *boolean isFile();*

This method returns true if the file object is pointing a file on HD.

How to check that whether the file object is pointing a folder or not?

ans: *boolean isDirectory();*

This method return true if the file object is pointing a directory on HD



## Write a program to check whether a file or folder is existing or not?

import java.io.File;

import java.util.Scanner;

public class CheckDemo{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.printf("Enter file/folder path including file/folder name:");

String fpath=sc.nextLine();

File f = new File(fpath);

if(f.exists()){

if(f.isFile())

System.out.println("it is existing file");

if(f.isDirectory())

System.out.println("it is existing directory");

}

else{

System.out.println("no such file or directory");

}

}

}

# Files and File Dates

### About java.util.Date() class

### Write a program to print pc date on monitor

import java.util.Date;

public class DateDemo1{

public static void main(String[] args){

Date d=new Date();//gets current pc date

System.out.println(d);//prints current pc date

}

}

### Write a program to print pc date continuously

import java.util.Date;

public class DateDemo2{

public static void main(String[] args){

Date pd = new Date();

System.out.printf("\r"+pd);

boolean b = true;

while(b){

Date nd = new Date();

if(nd.compareTo(pd)>0){

System.out.printf("\r%s",nd);

pd = nd;

}

}

}

}

Giving time in milliseconds to Date class constructor to create Date object?

If 0 milliseconds is given then date is 1970 jan 1st 5:30 IST

Observe the output of the following example:

import java.util.Date;

public class DateDemo3{

public static void main(String[] args){

Date d1 = new Date(0L);

System.out.println(d1);

Date d2 = new Date(1000L);

System.out.println(d2);

Date d3 = new Date(2000L);

System.out.println(d3);

File f = new File("c:\\gabbarsingh");

boolean b = f.mkdir();

long time = f.lastModified();//milliseconds from 1970 jan 1st 5:30 am to today

Date ld = new Date(time);

System.out.println(ld);

}

}

### How to get last modified date of a file?

ans: *long lastModified();*

We can give this return value to a date object and we can get date to be displayed.

### Program to get Last Modified Date File / Folder

import java.io.File;

import java.util.Scanner;

public class LastModifiedDemo{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.printf("Enter file/folder path including file/folder name:");

String fpath=sc.nextLine();

File f = new File(fpath);

if(f.exists()){

if(f.isFile())

System.out.println("it is existing file with last modified date:"+new java.util.Date(f.lastModified()));

if(f.isDirectory())

System.out.println("it is existing folder with last modfied date:"+new java.util.Date(f.lastModified()));

}

else{

System.out.println("no such file or directory");

}

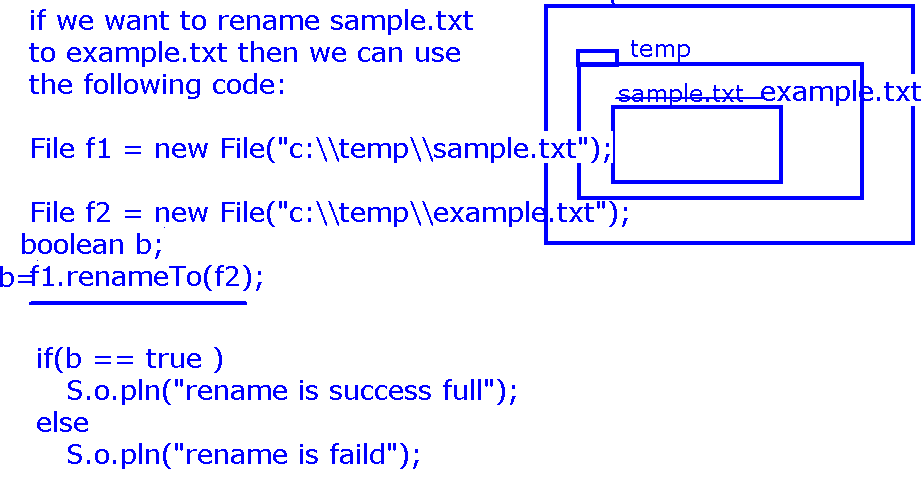
}

}

# Renaming File or Folder:

How to rename an existing file or folder?

ans: We use the following method: *boolean renameTo(File f);*



### Program to Rename file / Folder

import java.io.File;

import java.util.Scanner;

class RenameDemo{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.printf("Enter source file or folder path:");

String sfn = sc.nextLine();

System.out.printf("Enter destination file or folder path:");

String dfn = sc.nextLine();

File sf = new File(sfn);

File df = new File(dfn);

boolean b = sf.exists();

if(b==true){

boolean status = sf.renameTo(df);

if(status==true)

System.out.println("rename successfull");

else

System.out.printf("rename faild");

}

else

System.out.printf("no such file or directory");

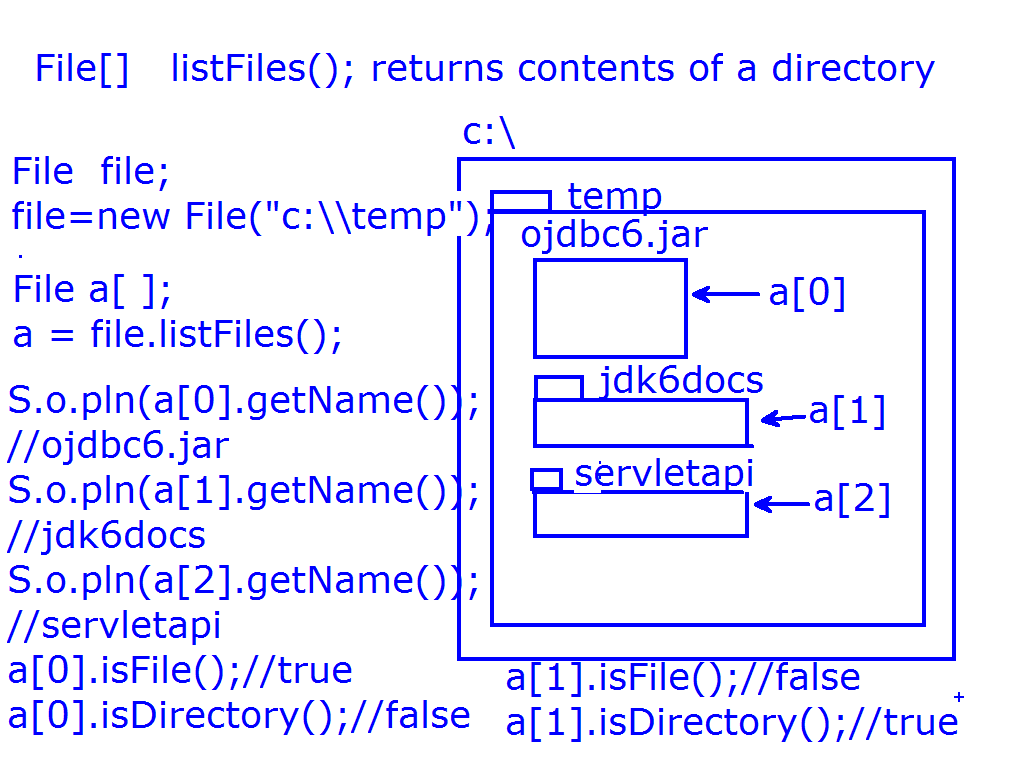
}

}

# Listing Folder Contents:

## How to get contents of the folder?

*File[] listFiles();* this method returns list of files stored in a directory



Program to get the Contents of the Folder:

### Write program to get the contents of the folder:

import java.util.Scanner;

import java.io.File;

public class ListFilesDemo{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("Enter folder path to show the contents:");

String fn = sc.nextLine();

File f = new File( fn );

File a[] = f.listFiles();

int fc=0, dc=0;

for(int i=0;i<a.length;i++){

String n = a[i].getName();

if(a[i].isFile()){

System.out.println(n+" is file");

fc++;

}

if(a[i].isDirectory()){

System.out.println(n+" is directory");

dc++;

}

}//end of for loop

System.out.println("number of files="+fc);

System.out.println("number of dirs="+dc);

}//end of main method

}//end of class

### Write a program to find size of a directory?

import java.util.Scanner;

import java.io.File;

public class DirSizeDemo{

static long size(File dir){

long length=0;

File a[] = dir.listFiles();

for(int i=0;i<a.length;i++){

System.out.printf("\radding size of:%s",a[i].getName());

if(a[i].isFile())

length = length + a[i].length();

if(a[i].isDirectory())

length=length+size(a[i]);

}

return length;

}

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("Enter directory path to find the size:");

String fn = sc.nextLine();

File f = new File(fn);

if(f.exists()){

if(f.isFile())

System.out.println("it is existing file size="+f.length()+" bytes");

if(f.isDirectory()){

long sinb = size(f);

double kb = sinb/(1024\*1.0);

double mb = kb/(1024\*1.0);

double gb = mb/(1024\*1.0);

System.out.println("size of directory in GB="+gb);

}

} else

System.out.println("no such file or directory");

}

}