Sri SAI RAM ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY
West Tambaram, Chennai - 44

Sairam INSTITUTIONS

SAIRAM DIGITAL RESOURCES

YEAR II

SEM III

CS8392

OBJECT ORIENTED PROGRAMMING
(Common to CSE, EEE, EIE, ICE, IT)

UNIT NO 3

EXCEPTION HANDLING

3.9 READING AND WRITING FILES

COMPUTER SCIENCE & ENGINEERING

JAVA FILES

- File handling is an important part of any application.
- Java has several methods for creating, reading, updating, and deleting files.

Java File Handling

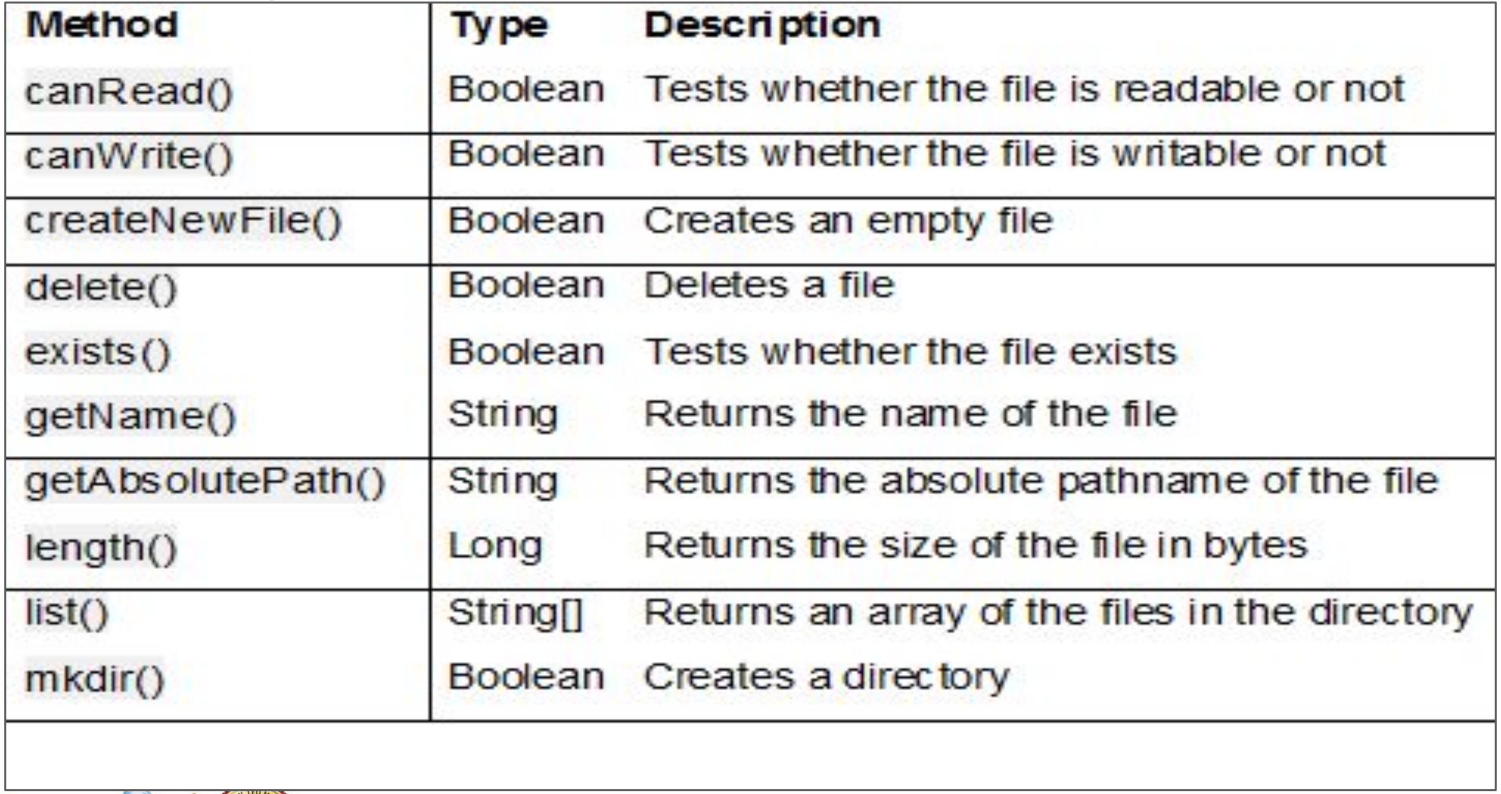
- The File class from the java.io package, is used to work with files.
- To use the File class, create an object of the class, and specify the filename or directory name:

Example

- `import java.io.File;  // Import the File class`
- `File myObj = new File("filename.txt"); // Specify the filename`

METHODS

- The FILE class has many useful methods for creating and getting information about files.
- For example



Method	Type	Description
canRead()	Boolean	Tests whether the file is readable or not
canWrite()	Boolean	Tests whether the file is writable or not
createNewFile()	Boolean	Creates an empty file
delete()	Boolean	Deletes a file
exists()	Boolean	Tests whether the file exists
getName()	String	Returns the name of the file
getAbsolutePath()	String	Returns the absolute pathname of the file
length()	Long	Returns the size of the file in bytes
list()	String[]	Returns an array of the files in the directory
mkdir()	Boolean	Creates a directory

CREATE A FILE

- createNewFile() method is used to create a file in Java.
- This method returns a boolean value: true if the file was successfully created, and false if the file already exists.
- Note that the method is enclosed in a try...catch block. This is necessary because it throws an IOException if an error occurs (if the file cannot be created for some reason)
- To create a file in a specific directory (requires permission), specify the path of the file and use double backslashes to escape the "\" character (for Windows).
- On Mac and Linux you can just write the path, like: /Users/name/filename.txt

Example

```
File myObj = new File("filename.txt");
```

EXAMPLE

```java
import java.io.File;  // Import the File class
import java.io.IOException;  // Import the IOException class to handle errors

public class CreateFile {
  public static void main(String[] args) {
    try {
      File myObj = new File("filename.txt");
      if (myObj.createNewFile()) {
        System.out.println("File created: " + myObj.getName());
      } else {
        System.out.println("File already exists.");
      }
    } catch (IOException e) {
      System.out.println("An error occurred.");
      e.printStackTrace();
    }
  }
}
```

READ A FILE

InputStream or Reader is used to read data from the file

```
// importing the FileReader class
import java.io.FileReader;
class Main {
  public static void main(String[] args) {
           char[] array = new char[100];
    try {
      // Creates a reader using the FileReader
              FileReader input = new FileReader("input.txt");
      // Reads characters
              input.read(array);
              System.out.println("Data in the file:");
              System.out.println(array);
      // Closes the reader
              input.close();
    }
    catch(Exception e) {
             e.getStackTrace();
    }
  }
}
```

WRITE A FILE

OutputStream or Writer is used to write data to the file

```
// importing the FileWriter class
import java.io.FileWriter;
 class Main {
   public static void main(String args[]) {

     String data = "This is the data in the output file";
     try {
       // Creates a Writer using FileWriter
              FileWriter output = new FileWriter("output.txt");

       // Writes string to the file
               output.write(data);
               System.out.println("Data is written to the file.");

       // Closes the writer
                output.close();
     }
     catch (Exception e) {
                e.getStackTrace();
     }
  }
   }
```

DELETE A FILE

```
import java.io.File;
class Main {
 public static void main(String[] args) {
   // creates a file object
          File file = new File("file.txt");
   // deletes the file
          boolean value = file.delete();
   if(value) {
          System.out.println("The File is deleted.");
   }
   else {
          System.out.println("The File is not deleted.");
   } }  }
```

VIDEO LINK

- https://www.youtube.com/watch?v=SslMi6ptwH8
- https://www.youtube.com/watch?v=BxCbxfpwC7Q