

# JAVA - FILEWRITER CLASS

This class inherits from the `OutputStreamWriter` class. The class is used for writing streams of characters.

This class has several constructors to create required objects.

Following syntax creates a `FileWriter` object given a `File` object.

```
FileWriter(File file)
```

Following syntax creates a `FileWriter` object given a `File` object.

```
FileWriter(File file, boolean append)
```

Following syntax creates a `FileWriter` object associated with a file descriptor.

```
FileWriter(FileDescriptor fd)
```

Following syntax creates a `FileWriter` object given a file name.

```
FileWriter(String fileName)
```

Following syntax creates a `FileWriter` object given a file name with a boolean indicating whether or not to append the data written.

```
FileWriter(String fileName, boolean append)
```

Once you have `FileWriter` object in hand, then there is a list of helper methods, which can be used to manipulate the files.

SN	Methods with Description
1	<b>public void write(int c) throws IOException</b> Writes a single character.
2	<b>public void write(char [] c, int offset, int len)</b> Writes a portion of an array of characters starting from offset and with a length of len.
3	<b>public void write(String s, int offset, int len)</b> Write a portion of a String starting from offset and with a length of len.

## Example:

Following is the example to demonstrate class:

```
import java.io.*;

public class FileRead{

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

        File file = new File("Hello1.txt");
```

```
// creates the file
file.createNewFile();
// creates a FileWriter Object
FileWriter writer = new FileWriter(file);
// Writes the content to the file
writer.write("This\n is\n an\n example\n");
writer.flush();
writer.close();

//Creates a FileReader Object
FileReader fr = new FileReader(file);
char [] a = new char[50];
fr.read(a); // reads the content to the array
for(char c : a)
    System.out.print(c); //prints the characters one by one
fr.close();
}
}
```

This would produce the following result:

```
This
is
an
example
```