**1) How to read information from a file in Java?**

To read information from a file in Java, you can use classes like FileReader, BufferedReader, or Scanner. Below are common methods:

**Using BufferedReader:**

import java.io.\*;

public class FileReaderExample {

public static void main(String[] args) {

try {

// Open the file

BufferedReader reader = new BufferedReader(new FileReader("example.txt"));

String line;

// Read the file line by line

while ((line = reader.readLine()) != null) {

System.out.println(line);

}

// Close the file

reader.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**Using Scanner:**

import java.io.\*;

import java.util.Scanner;

public class FileScannerExample {

public static void main(String[] args) {

try {

// Open the file

File file = new File("example.txt");

Scanner scanner = new Scanner(file);

// Read the file word by word

while (scanner.hasNextLine()) {

System.out.println(scanner.nextLine());

}

// Close the scanner

scanner.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

}

}

}

**Key Points:**

1. **Exception Handling**: File operations often throw IOException or FileNotFoundException.
2. **File Path**: You must provide the correct file path, either relative or absolute.
3. **Closing Resources**: Always close the file reader or scanner to release system resources.

**2) How to write information to a file in Java?**

To write to a file in Java, you can use classes like FileWriter, BufferedWriter, or PrintWriter.

**Using BufferedWriter:**

public class FileWriterExample {

public static void main(String[] args) {

try {

// Open the file

BufferedWriter writer = new BufferedWriter(new FileWriter("example.txt"));

// Write to the file

writer.write("Hello, World!");

writer.newLine(); // Add a new line

writer.write("This is a file-writing example.");

// Close the writer

writer.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**Using PrintWriter:**

public class PrintWriterExample {

public static void main(String[] args) {

try {

// Open the file

PrintWriter writer = new PrintWriter("example.txt");

// Write to the file

writer.println("Hello, World!");

writer.println("This is another example.");

// Close the writer

writer.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**Key Points:**

1. **File Overwriting**: FileWriter and PrintWriter overwrite the file by default. To append to a file, use:

new FileWriter("example.txt", true);

1. **Creating Files**: If the file does not exist, it will be created automatically.
2. **Closing Resources**: Always close the writer to ensure data is flushed to the file.

**Summary**

* **Reading**: Use BufferedReader for line-by-line reading or Scanner for word/line reading.
* **Writing**: Use BufferedWriter or PrintWriter for efficient writing.