What is file handling, read file, write data into a file, Java Input and Output Streams, Scanner class, BufferedReader class.

-K.L.MADHAVI

File Handling

File handling in Java is defined as reading and writing data to a file.The File Class is inside the java.io package that allows us to handle and work with different formats of files. Thus, if we want to use a file class, we need to create an object of that particular class and should specify the filename or directory name.

Why File Handling is Required?

- File Handling is an integral part of any programming language as file handling enables us to store the output of any particular program in a file and allows us to perform certain operations on it.

- In simple words, file handling means reading and writing data to a file.

Example program to create a file

```
// Importing File Class
import java.io.File;

class FileCreate {
	public static void main(String[] args)
	{

		// File name specified
		File obj = new File("myfile.txt");
		System.out.println("File Created!");
	}
}
```

Streams in Java

- In java, a sequence of data is known as a Stream.
- This concept is used to perform I/O operations on a file.
- There are 2 types of streams:

    1.input stream

    2.output stream

1.Input Stream

The Java InputStream class is the superclass of all input streams. The input stream is used to read data from numerous input devices like the keyboard, network, etc. InputStream is an abstract class, and because of this, it is not useful by itself. However, its subclasses are used to read data.

There are several subclasses of the InputStream class, which are as follows:

1.AudioInputStream

2.ByteArrayInputStream

3.FileInputStream

4.FilterInputStream

5.StringBufferInputStream

Creating an InputStream

```
// Creating an InputStream
InputStream obj = new FileInputStream(); //Here, an input stream is created using FileInputStream
```

2.Output Stream

The output stream is used to write data to numerous output devices like the monitor, file, etc. OutputStream is an abstract superclass that represents an output stream. OutputStream is an abstract class and because of this, it is not useful by itself. However, its subclasses are used to write data.

There are several subclasses of the OutputStream class which are as follows:

1. ByteArrayOutputStream
2. FileOutputStream
3. StringBufferOutputStream
4. DataOutputStream
5. PrintStream

Creating an OutputStream

```
// Creating an OutputStream
OutputStream obj = new FileOutputStream(); //Here, an output stream is created using FileOutputStream
```

File operations in Java

The following are the several operations that can be performed on a file in Java :

- **Create a File:** It is used to create a new file.
- **Read from a File:** We will use the Scanner class in order to read contents from a file.
- **Write to a File:** We use the FileWriter class along with its write() method in order to write some text to the file.
- **Delete a File:** We use the delete() method in order to delete a file.

Reading a file

Using scanner class:

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class ScannerExample {
    public static void main(String[] args) {
        try (Scanner scanner = new Scanner(new File("example.txt"))) {
            while (scanner.hasNextLine()) {
                String line = scanner.nextLine();
                System.out.println(line);
            }
        } catch (FileNotFoundException e) {
                System.out.println(e);
      }
    }
}
```

Writing into a file

Using printwriter class:

```
import java.io.PrintWriter;
import java.io.IOException;

public class PrintWriterExample {
    public static void main(String[] args) {
        try (PrintWriter pw = new PrintWriter("example.txt")) {
            pw.println("Hello, PrintWriter!");
            } catch (IOException e) {
                    System.out.println(e);
         }
    }
}
```

Scanner Class

In Java, Scanner is a class in java.util package used to read input or data from the keyboard.

Methods available in scanner class are:

1.next()-used to read String.

2.nextBoolean()-used to read boolean value.

3.nextByte()-used to read byte value.

4.nextShort()-used to read short value.

5.nextInt()-used to read int value.

6.nextLong()-used to read long value.

7.nextFloat()-used to read float value.

8.nextDouble()-used to read double value.

Continue…

Creation of Scanner class Object:

```
Scanner Sc=new Scanner();
```

We need to pass “System.in” as a parameter to read input from keyboard as shown:

```
Scanner Sc=new Scanner(System.in);
```

Example Program

```
import java.util.Scanner;
public class Main{
	public static void main(String[] args) {
		Scanner Sc=new Scanner(System.in);
		    String name=Sc.next();
		    int a=Sc.nextInt();
		    float b=Sc.nextFloat();
		    double c=Sc.nextDouble();
		    System.out.println(name);
		    System.out.println(a);
		    System.out.println(b);
		    System.out.println(c);
	 }
}
```

BufferedReader Class

In Java, BufferedReader is a class in java.io package used to read input or data from the keyboard and files.Reads characters from character input stream.

→InputStreamReader-read bytes and decoded to character set.

→FileReader-used to read data from files.

if file doesn’t exist then it will raise an io exception.

Methods:

1.read():reads single character.

2.readLine():reads multiple characters.

Example program

```
import java.io.*;
public class Main{
	public static void main(String[] args) {
		InputStreamReader ir=new InputStreamReader(System.in)
		BufferedReader br=new BufferedReader(ir);
		    int a=Integer.parseInt(br.readLine());
		    float b=Float.parseFloat(br.readLine());
		    String str=br.readLine();
		    System.out.println(a);
		    System.out.println(b);
		    System.out.println(str);
	 }
}
```