Java Lecture-4

Input/Output in Java

- The **io** package supports Java basic I/O (input/output) system
- Java programs perform I/O through streams
- A *stream* is an abstraction that either produces or consumes information.
- Java implements streams within class hierarchies defined in the **java.io** package.

- All Java programs automatically import the **java.lang** package.

io

- This package defines a class called **System**, which encapsulates several aspects of the run-time environment.

- **System** also contains three predefined stream variables: **in**, **out**, and **err**.

- These fields are declared as **public**, **static**, and **final** within **System**.

- This means that they can be used by any other part of your program and without reference to a specific **System** object.

- **System.out** refers to the standard output stream. By default, this is the console.

- **System.in** refers to standard input, which is the keyboard by default.

- **System.err** refers to the standard error stream, which also is the console by default.

- **System.in** is an object of type **InputStream**;

- **System.out** and **System.err** are objects of type **PrintStream**.

- To obtain a character-based stream that is attached to the console, wrap **System.in** in a **BufferedReader** object.

- **BufferedReader** supports a buffered input stream. A commonly used constructor is shown here:

  BufferedReader(Reader *inputReader*)

- Here, *inputReader* is the stream that is linked to the instance of **BufferedReader** that is being created.

- **Reader** is an abstract class.

- One of its concrete subclasses is **InputStreamReader**, which converts bytes to characters.

- To obtain an **InputStreamReader** object that is linked to **System.in**, use the following constructor:

  InputStreamReader(InputStream *inputStream*)

- Because **System.in** refers to an object of type **InputStream**, it can be used for *inputStream*.

- InputStreamReader() is a function that converts the input stream of bytes into a stream of characters so that it can be read as BufferedReader expects a stream of characters.

- Putting it all together, the following line of code creates a **BufferedReader** that is connected to the keyboard:

  BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

- After this statement executes, **br** is a character-based stream that is linked to the console through **System.in**.

```
// Java Program for taking user input using BufferedReader Class

import java.io.*;

class InputIntStr {

	// Main Method
	public static void main(String[] args)
		throws IOException
	{
		// Creating BufferedReader Object
		// InputStreamReader converts bytes to
		// stream of character
		BufferedReader br = new BufferedReader(
			new InputStreamReader(System.in));

		System.out.println("Enter String");
```

```
            String str = br.readLine();

            System.out.println("Enter Integer");

            int it = Integer.parseInt(br.readLine());

            System.out.println("Enter float");

            float ft = Float.parseFloat(br.readLine());

            //The parseInt() function is used to convert a String value to an integer value.

            System.out.println("Entered String : " + str);

            System.out.println("Entered Integer : " + it);

            System.out.println("Entered Float : " + ft);
        }
}
```

Using Scanner Class for Taking Input in Java

- It is an advanced version of BufferedReader which was added in later versions of Java.

- The scanner can read formatted input. It has different functions for different types of data types.

  - The scanner is much easier to read as we don’t have to write throws as there is no exception thrown by it.
  - It contains predefined functions to read an Integer, Character, and other data types.

Syntax of Scanner class

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

Importing Scanner Class

To use the Scanner we need to import the Scanner class from the util package as

```
import java.util.Scanner;
```

Inbuilt Scanner functions are as follows:

- **Integer:** nextInt()
- **Float:** nextFloat()
- **String :** next() and nextLine()

```
// Java Program to show how to take input from user using Scanner Class

import java.util.*;
class InputScan {

    public static void main(String[] args)
    {
        // Scanner definition
        Scanner scn = new Scanner(System.in);

        // input is a string ( one word )
        // read by next() function
        String str1 = scn.next();

        // print String
        System.out.println("Entered String str1 : " + str1);

        // input is a String ( complete Sentence )
        // read by nextLine()function
        String str2 = scn.nextLine();
```

```
// print string
            System.out.println("Entered String str2 : " + str2);

            // input is an Integer
            // read by nextInt() function
            int x = scn.nextInt();

            // print integer
            System.out.println("Entered Integer : " + x);

            // input is a floatingValue
            // read by nextFloat() function
            float f = scn.nextFloat();

            // print floating value
            System.out.println("Entered FloatValue : " + f);
      }
}
```

```
// Use a BufferedReader to read characters from the console.
      import java.io.*;

     class BRRead {
       public static void main(String args[]) throws IOException
       {
        char c;
         BufferedReader br = new
               BufferedReader(new InputStreamReader(System.in));
         System.out.println("Enter characters, 'q' to quit.");

         do {
          c = (char) br.read();
          System.out.println(c);
         } while(c != 'q');

} }
```