**Getting Started with the Scanner Class in Java**

The **Scanner** class is a useful utility in Java for reading input from the user, as well as from other input streams such as files.

The **Scanner** class is a tool that helps us read things from the computer. It can read numbers, words, and other kinds of information.

To use the **Scanner** class, we first have to tell the computer to get ready to read something. Then, we can ask the **Scanner** to read a number or a word for us.

For example, we can ask the **Scanner** to read a number like this:

1. import java.util.Scanner;
3. public class Main {
4. public static void main(String[] args) {
5. Scanner sc = new Scanner(System.in); // Create a Scanner object that reads from the standard input stream
6. System.out.print("Enter an integer: "); // Prompt the user to enter an integer
7. int num = sc.nextInt(); // Read the integer entered by the user
8. System.out.println("You entered: " + num); // Print the integer
9. sc.close(); // Close the Scanner object when you are done using it (why? explained soon)
10. }
11. }

To use the **Scanner** class, you first need to import it from the **java.util** package.

Then, you can create a new **Scanner** object that reads from the standard input stream by calling the **Scanner** constructor and passing it **System.in** as an argument (e.g. **Scanner sc = new Scanner(System.in);**).

You can then use the various methods of the **Scanner** class to read different types of input from the user.

For example, you can use

* **nextInt()** method to read an integer,
* **nextLine()** method to read a string, and
* **nextDouble()** method to read a floating-point number.

Reading a **string**from the user:

1. Scanner sc = new Scanner(System.in);
2. System.out.print("Enter a string: ");
3. String str = sc.nextLine();

Reading a **floating**-point number from the user:

1. Scanner sc = new Scanner(System.in);
2. System.out.print("Enter a floating-point number: ");
3. double num = sc.nextDouble();

Don't forget to close the **Scanner** object when you are done using it to free up system resources (e.g. **sc.close();**).

It is a good practice to close a **Scanner** object when you are done using it because it helps to free up system resources. When a **Scanner** object is no longer needed, it is a good idea to close it so that it can release any resources it is holding onto, such as file handles or network sockets.

Closing a **Scanner** object is especially important when you are reading input from a file, as it will release the file handle and allow other programs to access the file.

To close a **Scanner** object, you can call the **close()** method on it (e.g. **sc.close();**).