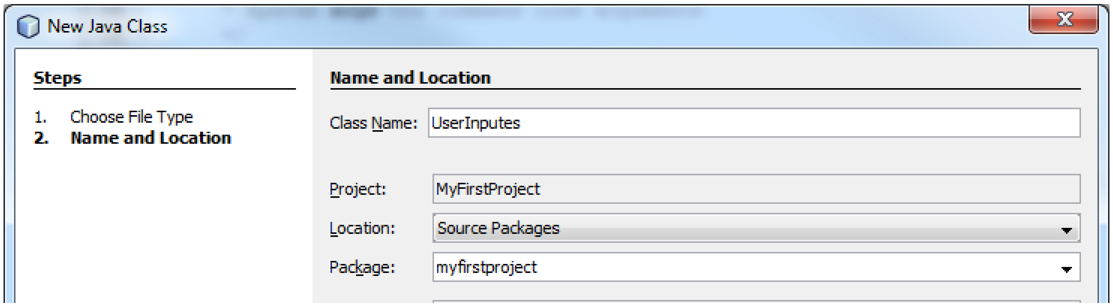
Working with user inputs in Java

You will need to create a new class in **myfirstproject** package. So, right click the package myfirstproject, select **New File** from the menu and then select Java under Categories and **Java Class** under file types.

Let us call the new class **UserInputs** so, from the **New Java Class** window that will appear, enter **UserInputs** as the Class Name. Click **Finish**.



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

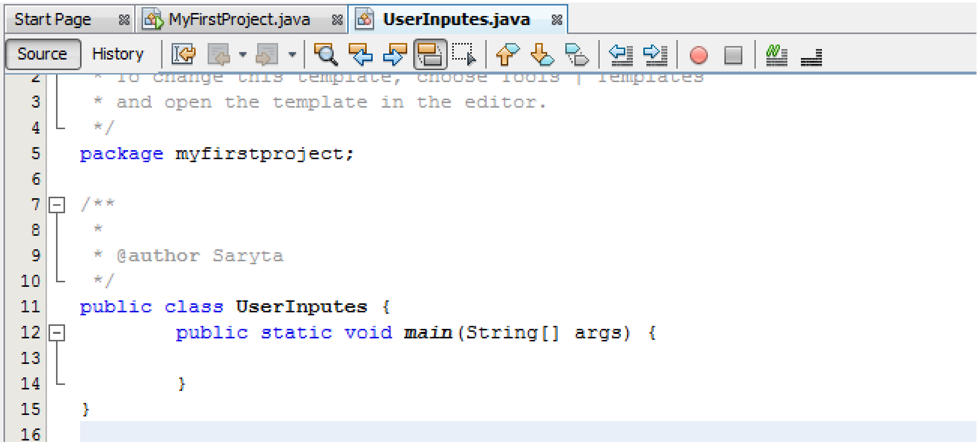
Class Name: UserInputes

Project: MyFirstProject

Location: Source Packages

Package: myfirstproject

Include the "**main**" method in your class. Your code editor window should look like shown below:



```
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package myfirstproject;

/**
 *
 * @author Saryta
 */
public class UserInputes {
        public static void main(String[] args) {

        }
}
```

As I mentioned earlier, Java contains a lot of functions or methods that are really useful to a programmer. These methods are written to accomplish a specific task that you may need to execute in your applications.

Related methods are grouped together and stored as a class. Related classes are stored in a package in Java library. When we want to use a certain method in Java, we reference its package and then the class into which it belongs.

One of the useful classes that handle inputs from a user in Java is the **Scanner** class. The **Scanner** class is located in the **util** (utility) package of the Java library. To use the **Scanner** class, we use the keyword **import** to reference it in our code, see below.

import java.util.Scanner;

The above statement tells Java that you want to use the Scanner class located in the **util** package of the **java** library. To use the class, we create new object of the class. This object work as the link between our code and the class and so we can be able to access the functions of the class from our code. We create the new object by using the following code:

Scanner user_input = new Scanner(System.in);

Notice that instead of creating an **int** variable or a **String** variable, we are creating a **Scanner** variable. We have called our Scanner variable **user_input**. After an equals sign, we have the keyword **new**. The keyword **new** is used to create new objects from a class. Here, we are creating an object from the **Scanner** class. In between the brackets we are specifying that the object is for a system input and we show this by including **System.in** inside the brackets.

To get the inputs from the user, we are going to use our new object, **user_input**, to call one of the many input methods available in the **Scanner** class. We'll use the **next** method. This method gets the next string of text that a user types on the keyboard. It is one of the many methods in Scanner class. As you type **next** in the code editor you'll see a list of other available methods.

```
Scanner user_input = new Scanner( System.in );
String user_name; user_name = user_input.next( );
```

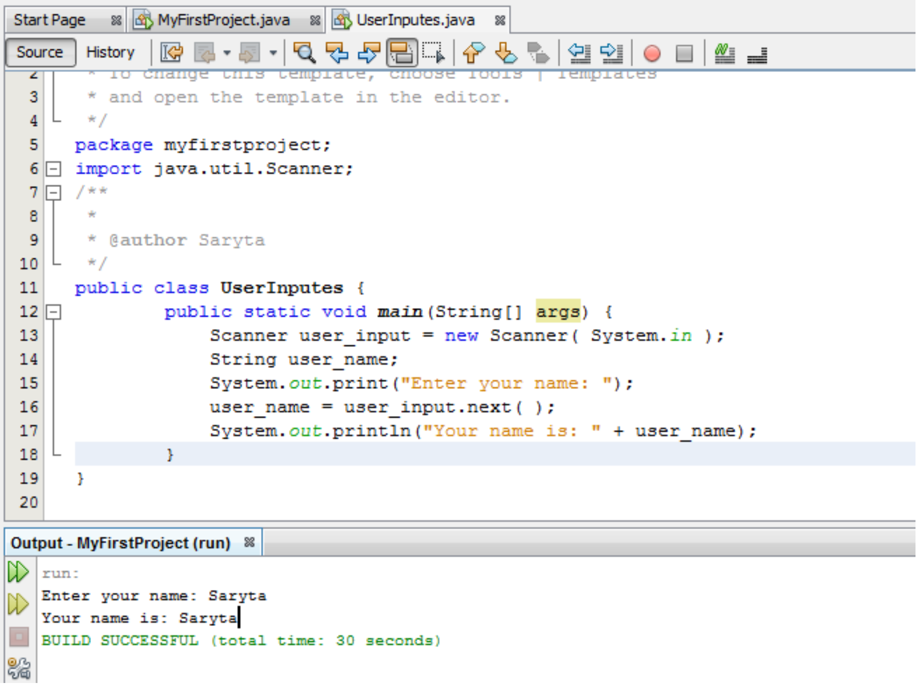
Before we take the user name as input, we'll add another statement to ask the user to enter the input. We have also created a **String** variable called **user_name** to store that input.

```
Scanner user_input = new Scanner( System.in );
String user_name;
System.out.print("Enter your name: ");
user_name = user_input.next( );
```

Notice that we have used **print** instead of **println** like we did earlier. The difference between the two is that **println** makes the cursor to move to a new line after the output, but with **print** the cursor stays on the same line. Let us now add our popular method to output the user input on the output window.

```
Scanner user_input = new Scanner( System.in );
String user_name;
System.out.print("Enter your name: ");
user_name = user_input.next( );
System.out.println("Your name is: " + user_name);
```

You code editor window should now look like shown below:



```
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package myfirstproject;
import java.util.Scanner;
/**
 *
 * @author Saryta
 */
public class UserInputes {
        public static void main(String[] args) {
            Scanner user_input = new Scanner( System.in );
            String user_name;
            System.out.print("Enter your name: ");
            user_name = user_input.next( );
            System.out.println("Your name is: " + user_name);
        }
}
```

```
run:
Enter your name: Saryta
Your name is: Saryta
BUILD SUCCESSFUL (total time: 30 seconds)
```

To run your program, go to the project window, right click UserInput and select **Run File**

When you run the program, Java will pause and ask you to enter your name, enter a name of your choice and press enter.

Exercise: Modify your code so that you can input surname and first name.