**Accepting Input from a User**

One of the strengths of Java is the huge libraries of code available to you. This is code that has been written to do specific jobs. All you need to do is to reference which library you want to use, and then call a method into action. One really useful class that handles input from a user is called the **Scanner** class. The Scanner class can be found in the **java.util** library. To use the Scanner class, you need to reference it in your code. This is done with the keyword **import**.

import **java.util.Scanner;**

The import statement needs to go just above the Class statement:

|  |
| --- |
| **import java.util.Scanner;**  **public class StringVariables {**  **}** |

This tells java that you want to use a particular class in a particular library - the Scanner class, which is located in the java.util library.

The next thing you need to do is to create an object from the Scanner class. (A class is just a bunch of code. It doesn't do anything until you create a new object from it.)

To create a new Scanner object the code is this:

|  |
| --- |
| **Scanner user\_input = new Scanner(**System.in**);** |

So instead of setting up an **int** variable or a **String** variable, we're setting up a **Scanner** variable. We've called ours **user\_input**. After an equals sign, we have the keyword **new**. This is used to create new objects from a class. The object we're creating is from the Scanner class. In between round brackets we have to tell java that this will be System Input (System.in).

To get the user input, you can call into action one of the many methods available to your new Scanner object. One of these methods is called **next**. This gets the next string of text that a user types on the keyboard:

|  |
| --- |
| **String first\_name; first\_name = user\_input.next( );** |

So after our user\_input object we type a dot. You'll then see a popup list of available methods. Double click **next** and then type a semicolon to end the line. We can also print some text to prompt the user:

|  |
| --- |
| **String first\_name; System.out.print("Enter your first name: "); first\_name = user\_input.next( );** |

Notice that we've used **print** rather than **println** like last time. The difference between the two is that println will move the cursor to a new line after the output, but print stays on the same line.

We'll add a prompt for a family name, as well:

|  |
| --- |
| **String family\_name; System.out.print("Enter your family name: "); family\_name = user\_input.next( );** |

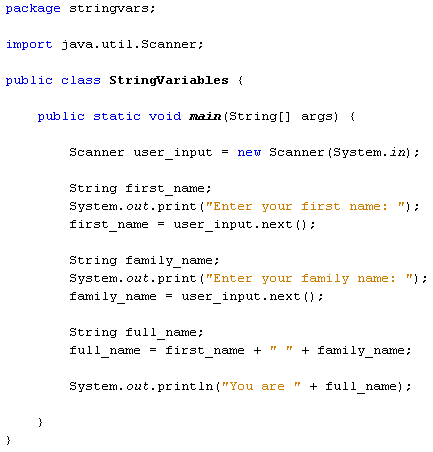
This is the same code, except that java will now store whatever the user types into our family\_name variable instead of our first\_name variable.

To print out the input, we can add the following:

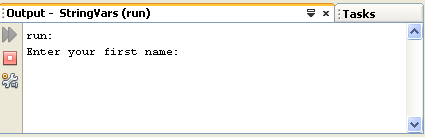
|  |
| --- |
| **String full\_name; full\_name = first\_name + " " + family\_name;**  **System.out.println("You are " + full\_name);** |

We've set up another String variable, **full\_name**. We're storing whatever is in the two variables **first\_name** and **family\_name**. In between the two, we've added a space. The final line prints it all out in the Output window.

So adapt your code so that it matches what is shown below:



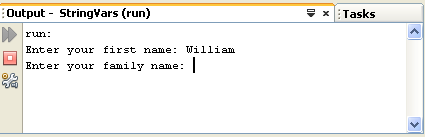
Run your program until your Output window displays the following:



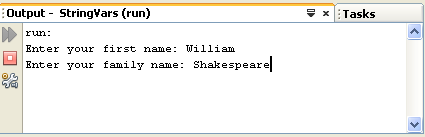
Java is now pausing until you enter something on your keyboard. It won't progress until you hit the enter key. So left click after "Enter your first name:" and you'll see your cursor flashing away. Type a first name, and then hit the enter key on your keyboard.

After you hit the enter key, java will take whatever was typed and store it in the variable name to the left of the equals sign. For us, this was the variable called first\_name.

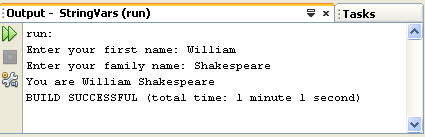
The program then moves on to the next line of code:



Type a family name, and hit the enter key again:



The user input has now finished, and the rest of the program executes. This is the output of the two names. The final result should like this:



So we used the Scanner class to get input from a user. Whatever was typed was stored in variables. The result was then printed to the Output window.

**Exercises:**

1. Change the program so that it also adds the users’ age. Use the following steps:

* Create an int variable called “Age”.
* Ask for the users age
* Use the code age = keyboard.nextInt(); to get the int value.
* Output the users Name and Age on the same line

### **Name, Age, and Salary**

1. Create a new program and ask the user for their name. Then display their name to prove that you can recall it. Ask them for their age. Then display that. Finally, ask them for how much they earn and display that. You should use the most appropriate data type for each variable.

### **More User Input of Data**

1. Create a new program and ask the user for several pieces of information, and display them on the screen afterward as a summary.

* First name
* Last name
* Date of birth
* Current Qualification
* Student id number

You must use the most appropriate type for each variable and not just Strings for everything.

### **Age in Five Years**

1. Create a new program. Ask the user for their name. Then display their name to prove that you can recall it. Ask them for their age. Then display what their age would be five years from now. Then display what their age would be five years ago.