

Console Input and Output

Getting Information and Displaying Information on
the console

Back to the beginning

- Start a new project
 - Open NetBeans and select new project from the file menu and name it Project2
- Enter the following code in main method of the Project2 class:

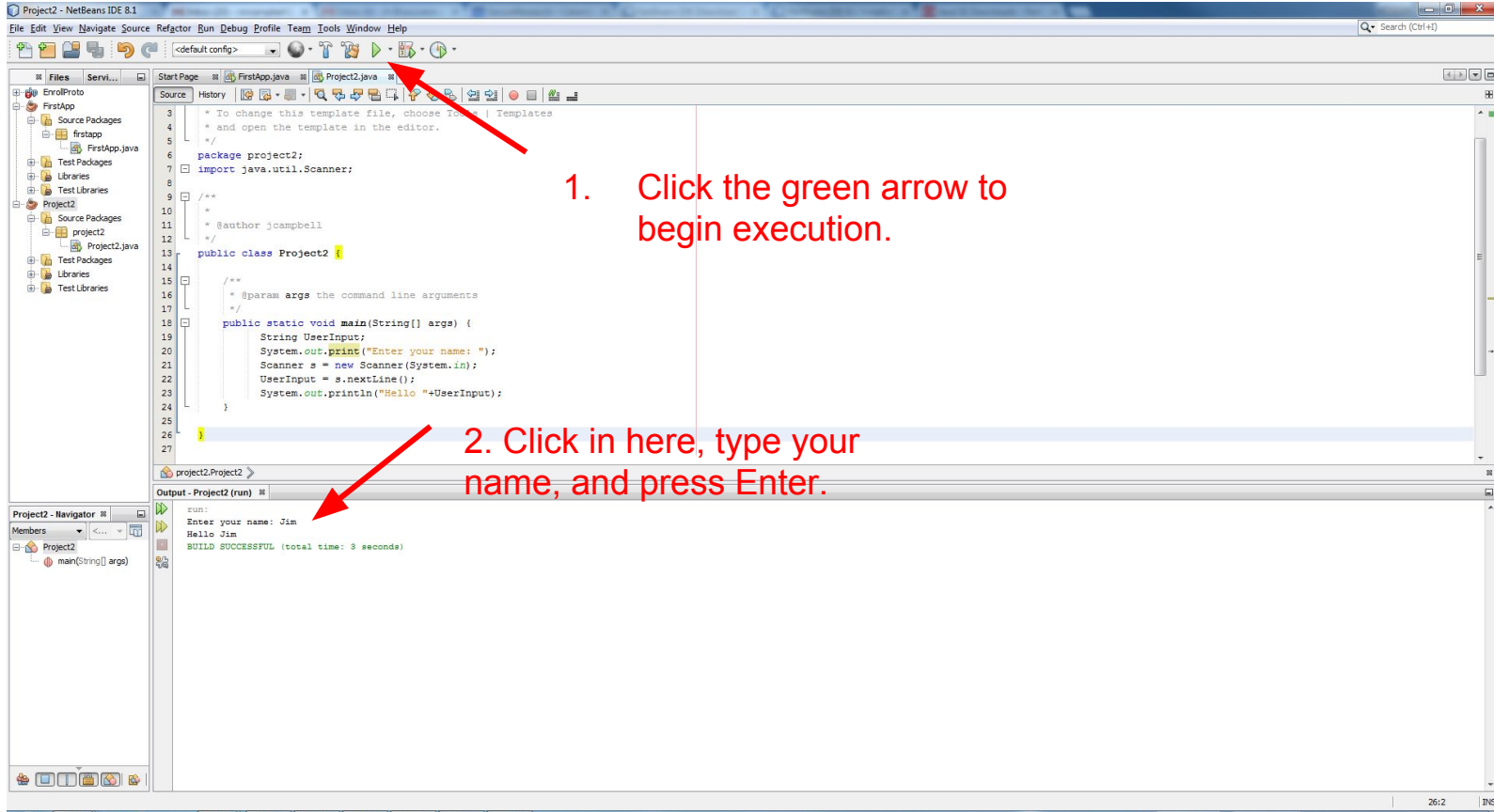
```
String userInput;  
  
System.out.print("Enter your name: ");  
  
Scanner s = new Scanner(System.in);  
  
UserInput = s.nextLine();  
  
System.out.println("Hello "+UserInput);
```

- Netbeans may add:

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

At the top of the file. If it does not please add this line. This imports the Scanner class into your project.

Execute the project



1. Click the green arrow to begin execution.

```
3  * To change this template file, choose the version for your IDE from the Templates
4  * and open the template in the editor.
5  */
6  package project2;
7  import java.util.Scanner;
8
9  /**
10 *
11 * @author jcampbell
12 */
13 public class Project2 {
14
15     /**
16     * @param args the command line arguments
17     */
18     public static void main(String[] args) {
19         String userInput;
20         System.out.println("Enter your name: ");
21         Scanner s = new Scanner(System.in);
22         userInput = s.nextLine();
23         System.out.println("Hello " + userInput);
24     }
25
26 }
27
```

2. Click in here, type your name, and press Enter.

```
run:
Enter your name: Jim
Hello Jim
BUILD SUCCESSFUL (total time: 3 seconds)
```

String

- In this example we define UserInput as a String.
- String is a Java class with several methods
 - (<https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>)
- We can use the compareTo method to evaluate against another string.
 - UserInput.compareTo("Jim");
 - If the return value is 0 (zero), the strings are the same.
 - compareTo is case sensitive - compareToIgnoreCase is another method that is not case sensitive.
- There are other methods that are part of the string class. See the documentation referenced above for details.

Make a decision!

- Conditionals evaluate variables and take action based on the result.
 - == equals ,< less than,> greater than
- Let's evaluate our String to see if we have a great name.

```
int x;  
x=UserInput.compareTo("Jim");  
if(x==0){  
    System.out.println("Great name!");  
}else{  
    System.out.println("I guess that's on OK name!");  
}
```

Try the code

Add the code from the previous slide to your project

What happens when you execute it?

The if conditional

```
if (x > y){
```

```
    /*if x is greater than y do this */
```

```
}else{
```

```
    /*if x is not greater than y do this*/
```

```
}
```

Operators

- < - Less than
 - > - Greater Than
 - == - Equal to
 - != - not Equal to
 - || - or
 - && - and
 - ! - not
- 1<2
 - 2>1
 - 1==1
 - 1!=2
 - 1||0 (true) 1||1 (true) 0||0 (false)
 - 1&&1 (true) 1&&0 (false) 0&&0 (false)
 - if(x<y && y>z)...

Different ways of doing things...

In our example we used:

```
int x;
x=UserInput.compareTo("Jim");
if(x==0){
    System.out.println("Great Name");
}else{
    System.out.println("I guess that's on OK
name!");
}
```

Different ways of doing things...

We could have done

```
if (UserInput.compareTo("Jim")==0) {  
    System.out.println("Great Name");  
}else{  
    System.out.println("I guess that's on OK name!");  
}
```

Project 2: Source Code

<http://km2arc.kellenberg.org/code/java/project-2/>