Here is a second example using the Scanner class to get user input, building on the concepts of classes and objects. We'll create a Rectangle class and then use user input to define the dimensions of a rectangle object.

**The Rectangle Class**

First, create a Rectangle class. This class will represent a rectangle with properties for width and height, and a method to calculate its area.

Java

// Rectangle.java

public class Rectangle {

// Properties

double width;

double height;

// Method to calculate the area

public double getArea() {

return width \* height;

}

}

This class is a simple blueprint. It doesn't have any predefined values for width and height. We'll use user input to set these values for an object.

**Getting Input for an Object**

Next, create a new class, maybe called ShapeCalculator, with a main method. We'll use the Scanner class to get input from the user and create a Rectangle object based on that input.

Java

// ShapeCalculator.java

import java.util.Scanner;

public class ShapeCalculator {

public static void main(String[] args) {

// Create a Scanner object to read user input

Scanner scanner = new Scanner(System.in);

// Prompt the user for the rectangle's width

System.out.print("Enter the width of the rectangle: ");

double widthInput = scanner.nextDouble();

// Prompt the user for the rectangle's height

System.out.print("Enter the height of the rectangle: ");

double heightInput = scanner.nextDouble();

// Create a new Rectangle object

Rectangle myRectangle = new Rectangle();

// Set the object's properties using the user input

myRectangle.width = widthInput;

myRectangle.height = heightInput;

// Calculate and display the area

double area = myRectangle.getArea();

System.out.println("The area of the rectangle is: " + area);

// Close the scanner

scanner.close();

}

}

**How it works:**

1. **import java.util.Scanner;**: This line imports the necessary Scanner class.
2. **Scanner scanner = new Scanner(System.in);**: This creates a new **Scanner object**. The System.in argument tells it to read from the standard input stream, which is your keyboard.
3. **double widthInput = scanner.nextDouble();**: This line waits for the user to type a number and press Enter. The typed number is then stored in the widthInput variable. The same process happens for heightInput.
4. **Rectangle myRectangle = new Rectangle();**: An instance of the Rectangle class is created.
5. **myRectangle.width = widthInput;**: The values from the user input (widthInput and heightInput) are assigned to the myRectangle object's properties.
6. **myRectangle.getArea();**: The object's method is called to perform the calculation based on its newly assigned properties.

This example demonstrates how an object's properties can be **dynamically set at runtime** by a user, making the program interactive.