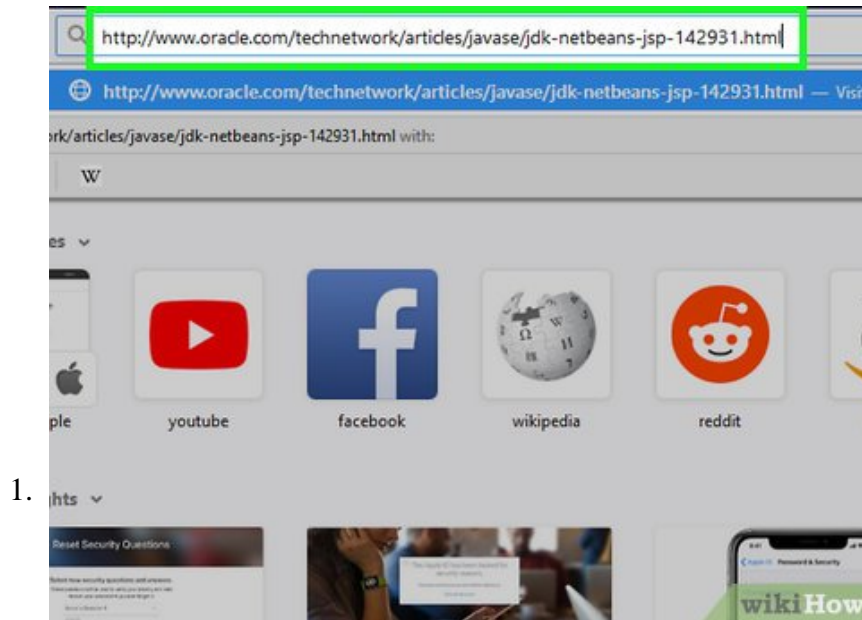


How to Create a Java Class in Netbeans to Compute the Area of a Triangle Using Heron's Formula

This guide will walk you through the steps need to compute the area of a triangle using the Heron's Formula in NetBeans. You may use any IDE you like, but this tutorial will be using NetBeans and Oracle's Java JDK 8u101. Download and...

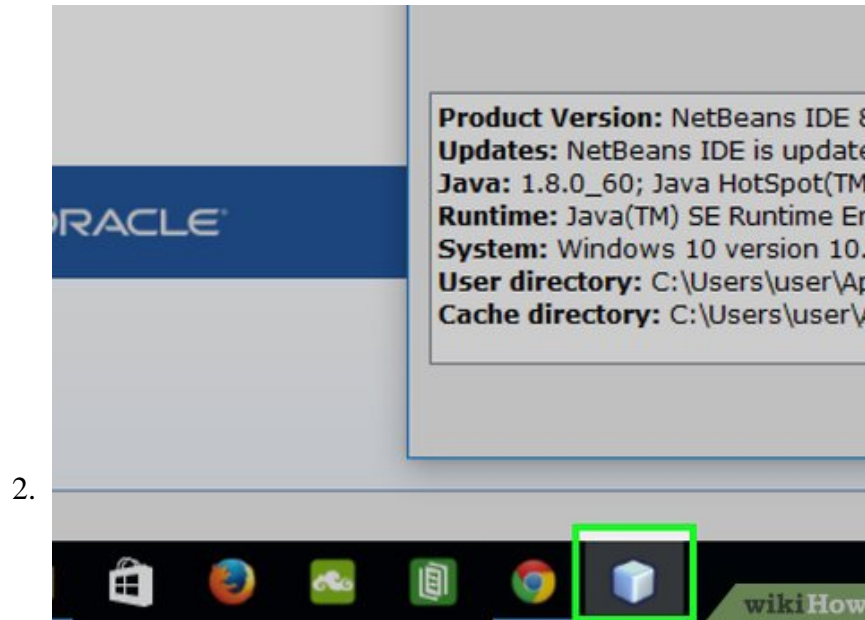
Part 1 of 2:

Preparing Your Java Project

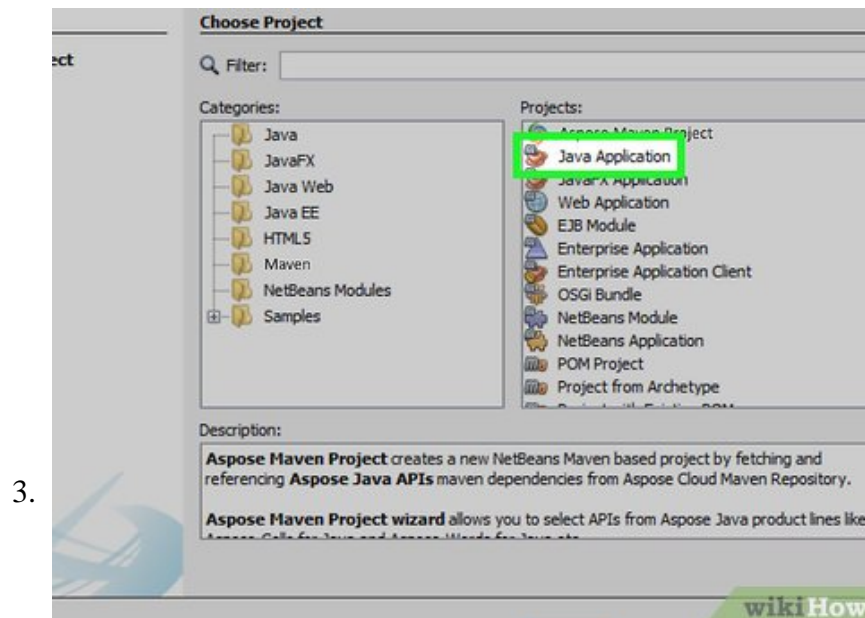


Download and install the Java JDK and NetBeans bundle here:

<http://www.oracle.com/technetwork/articles/javase/jdk-netbeans-jsp-142931.html>

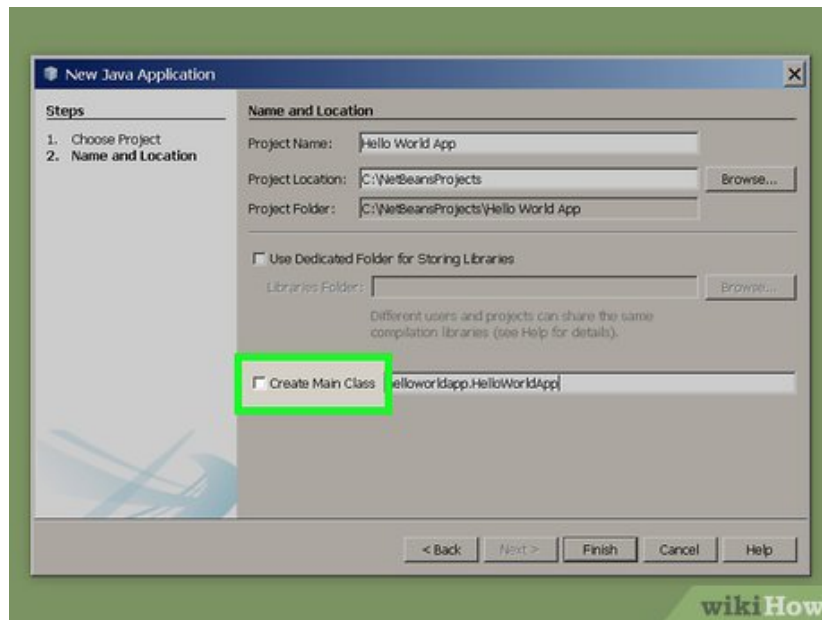


Open NetBeans on your computer and select new project.



Pick your project settings. Under categories select `Java` and under projects select `Java Application`. Then click `Next`.

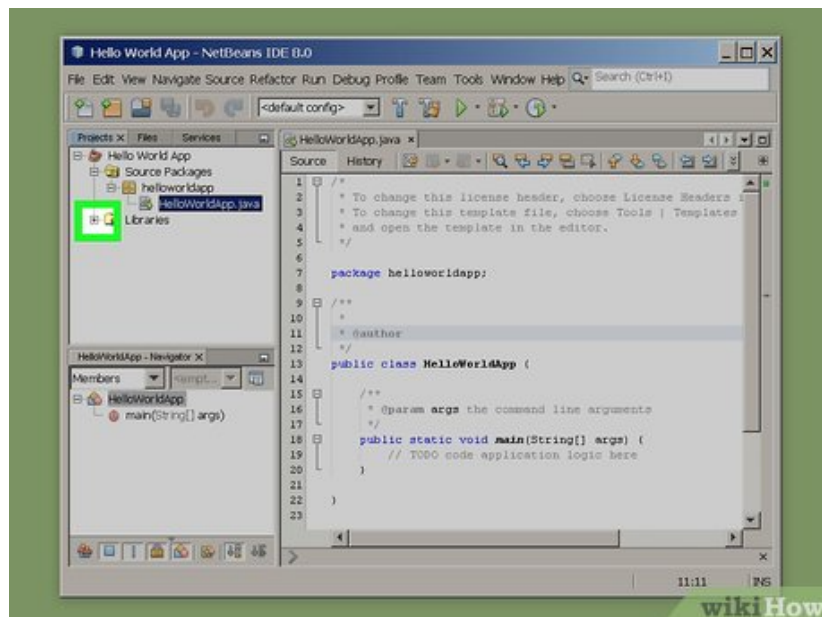
4.



Deselect `Create Main Class` and then give your project a name. The one here is named "Heron". Then click `Finish`.

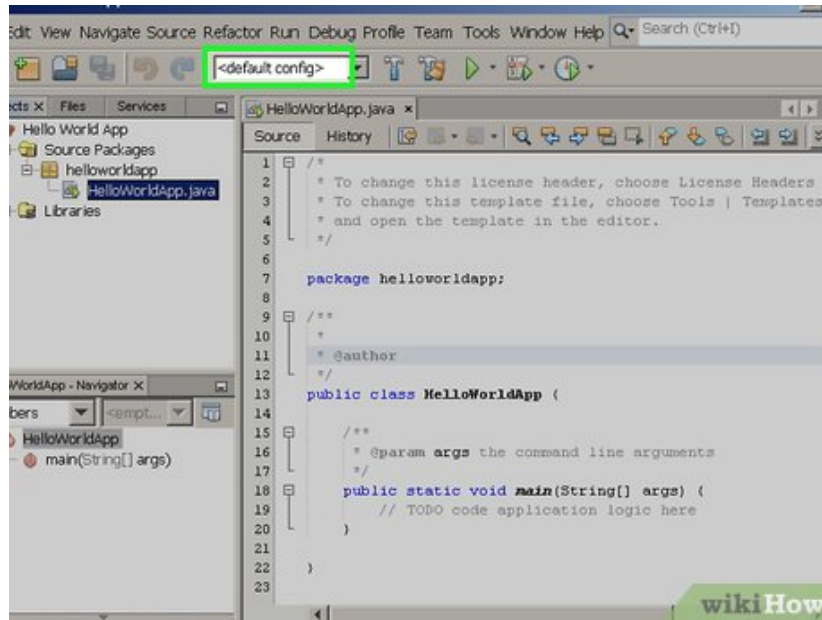
1. You can choose any file destination for this project.

5.



Open the menu to create a new Java class.

1. By default the `Projects` tab will be open. With the projects tab open, select the "+" (or drop down icon) to the right of your "Heron" project within the Projects tab.
2. You will then see two more items, the `Source Packages` and `Libraries`.
3. Click the `+` to open the Source Packages (or drop down icon).
4. Right click, select `New > Java Class`.

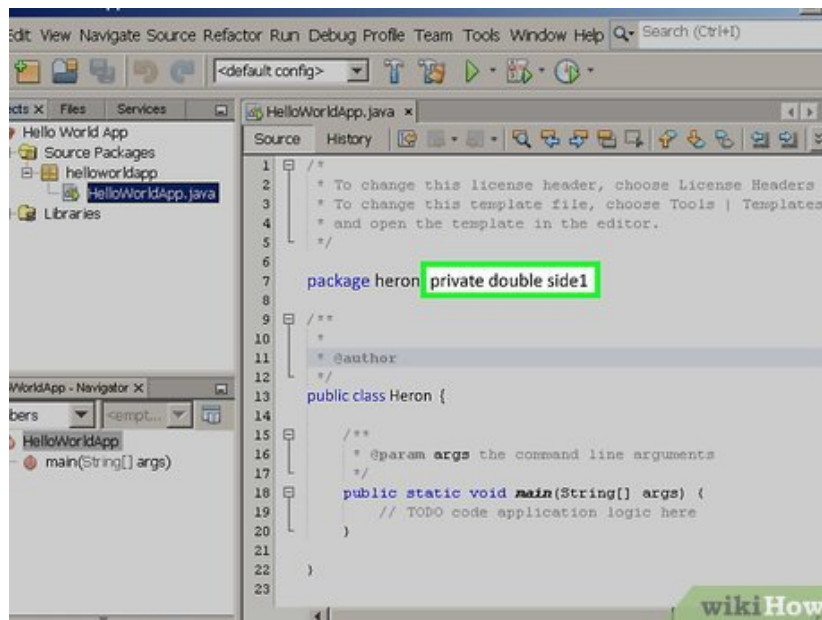


Give your class a name. Remember that it must start with an uppercase letter.

1. Click . You are now ready to write the code.

Part 2 of 2:

Programming Your Java Class



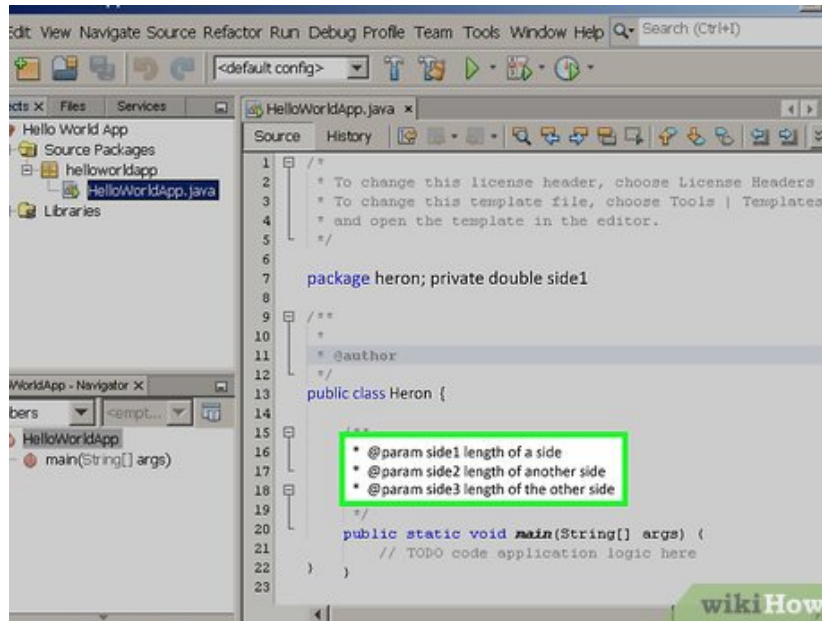
Set up 3 instance variables for this class, one for each of the three sides.

1. It is good to make them private and you will also want to give them a type of double so that you can get decimals for more accuracy.
2. Give your instance variables each their own unique name. Make it something literal, for example, mine was side1, side2, side3.

3. Insert the instance variables under the bracket after Heron.

1. `private double side1;`
2. `private double side2;`
3. `private double side3;`

2.

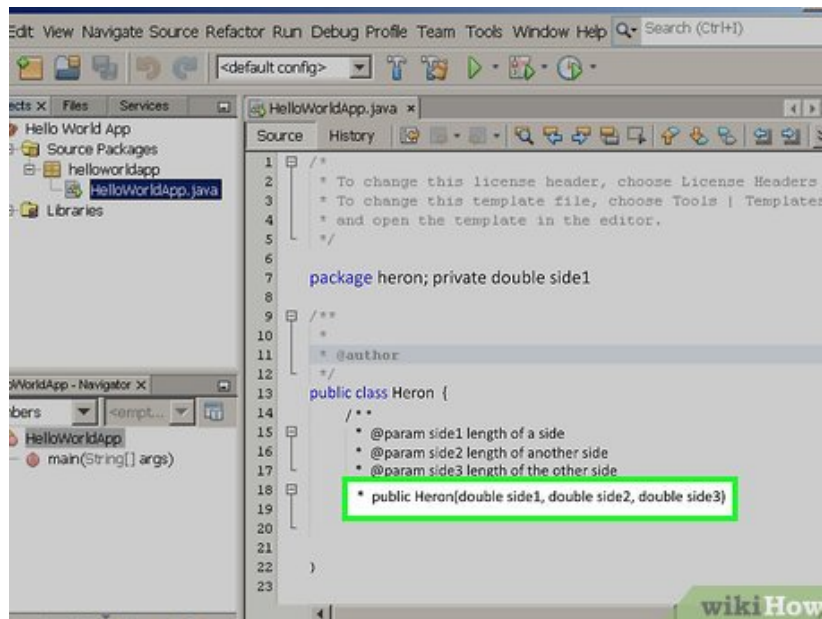


```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6
7  package heron; private double side1
8
9
10 /**
11 *
12 * @author
13 */
14 public class Heron {
15
16     /**
17     * @param side1 length of a side
18     * @param side2 length of another side
19     * @param side3 length of the other side
20     */
21     public static void main(String[] args) {
22         // TODO code application logic here
23     }
24 }
```

Create the parameters.

1. Under the instance variables, create a comment section listing the parameters.
2. To make a long comment, type `/**` and then hit `Enter`. This will create a multi-line comment.
3. Type the parameters in this space.
4. Create a triangle, given the lengths of the side:
 1. `@param side1` length of a side
 2. `@param side2` length of another side
 3. `@param side3` length of the other side

3.

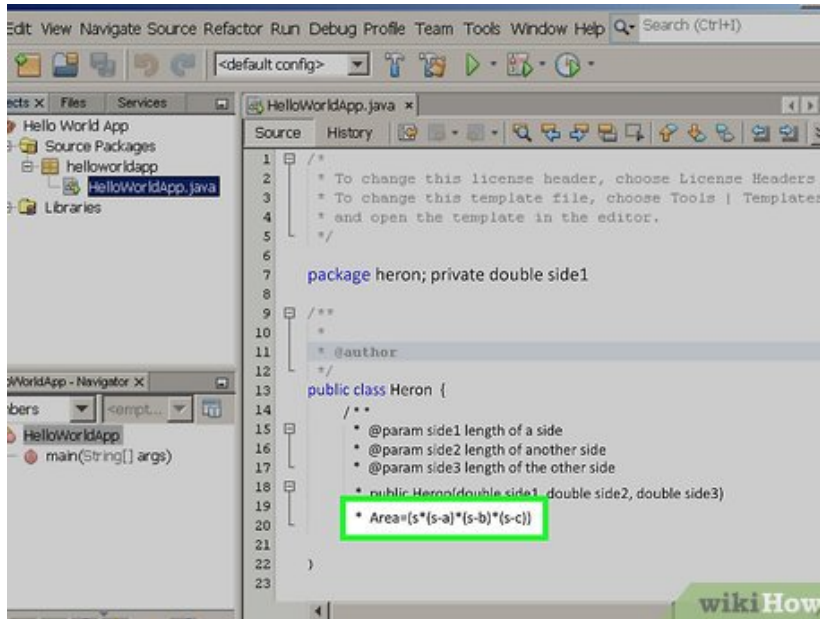


```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6
7  package heron; private double side1
8
9
10 /**
11 *
12 * @author
13 */
14 public class Heron {
15
16     /**
17     * @param side1 length of a side
18     * @param side2 length of another side
19     * @param side3 length of the other side
20     */
21     public Heron(double side1, double side2, double side3)
22
23 }
```

Create our constructor.

1. Create a public constructor so that when you create an object in your tester class, it will have the 3 parameters you established earlier.
2. Your constructors parameters should all be set to type double.
3. It should read: `public Heron(double side1, double side2, double side3) {`
4. Now create 'this' statements to link to the parameters. Type these after the bracket and then close the bracket.

4.



Create a mutator method to return the area of the triangle.

1. Create a mutator method that uses Heron's formula.
2. Set the method to public and give it a type double.
3. Give the method a name of your choosing, like `getArea`. Make sure to use camel case meaning the first word is lowercase and the second is uppercase.
4. It should read: `public double getArea() {`
5. You need a variable that calculates half of the perimeter. To do this, give it a type double and a name, like "halfPerim". Set it equal to a formula that would return half of the perimeter of a triangle. In this case, $halfPerim = (side1 + side2 + side3) / 2$
6. Now you need to create the formula that actually gives us the area and uses Heron's formula. You will also give this a type double and set area equal to the formula.
7. Heron's formula:

Picture 11 of How to Create a Java Class in Netbeans to Compute the Area of a Triangle Using Heron's Formula

where

Picture 12 of How to Create a Java Class in Netbeans to Compute the Area of a Triangle Using Heron's Formula

is equal to half the perimeter, and

Picture 13 of How to Create a Java Class in Netbeans to Compute the Area of a Triangle Using Heron's Formula

,

Picture 14 of How to Create a Java Class in Netbeans to Compute the Area of a Triangle Using Heron's Formula

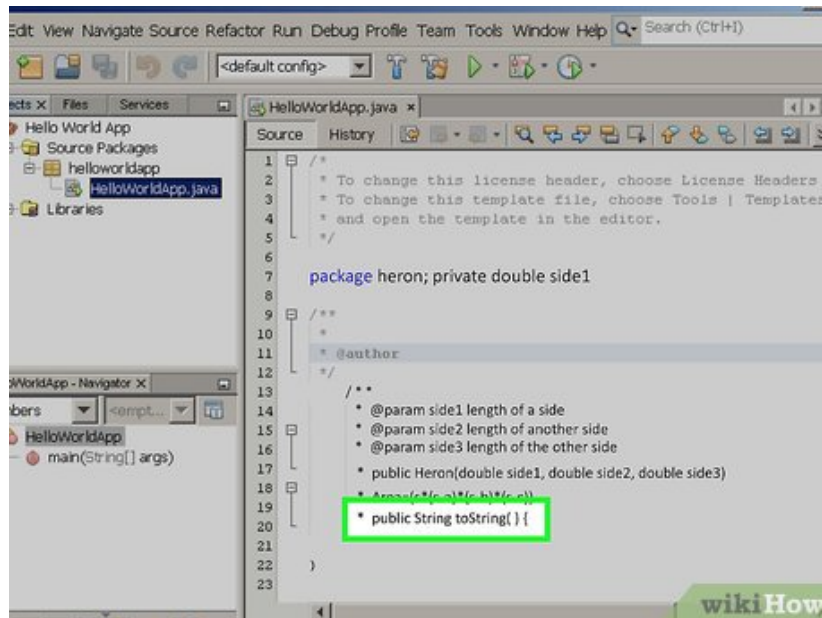
, and

Picture 15 of How to Create a Java Class in Netbeans to Compute the Area of a Triangle Using Heron's Formula

are the side lengths.

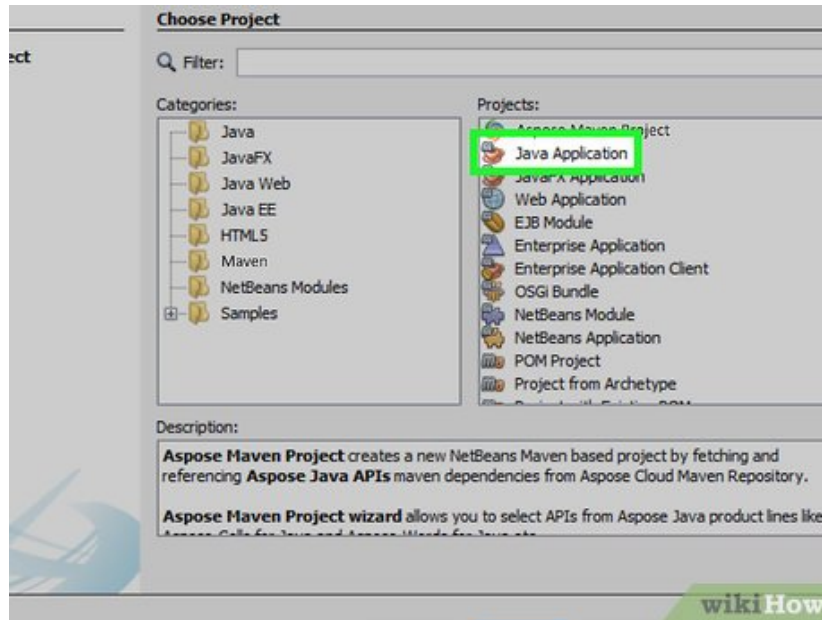
8. `double area = Math.sqrt(halfPerim * (halfPerim - side1) * (halfPerim - side2) * (halfPerim - side3));`
9. The last step is to create a return statement. Use `return area` and then close your bracket.

5.



Create a string to return each side of the triangle.

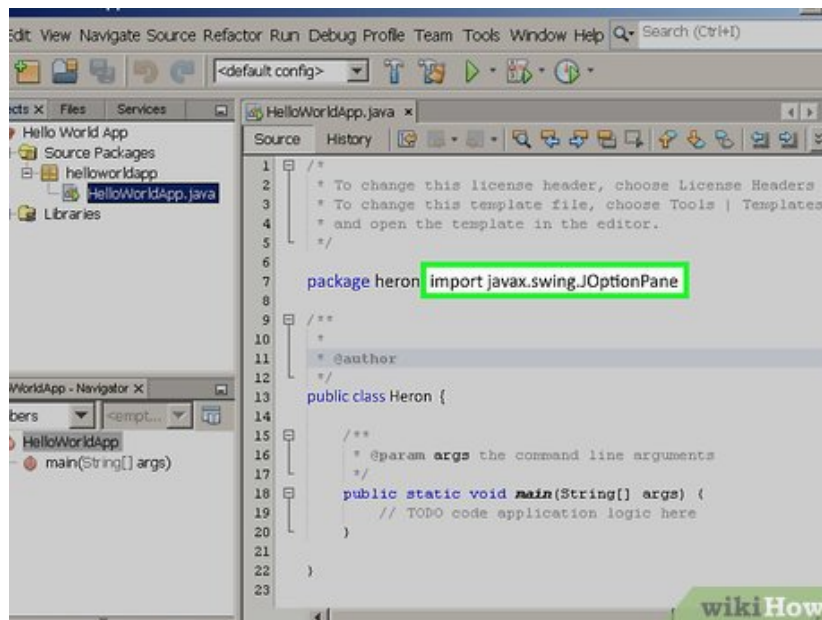
1. Create a public string to return the lengths of each side of the triangle:
 1. `public String toString() {`
2. Make sure the S in string is capitalized and then after the brackets insert your return statement:
 1. `return side1 + ", " + side2 + ", " + side3`
3. Then close your bracket and your class is now complete!



6.

Create a test class. It will be your Main Class that will run the file.

1. Create a tester class to show that your Heron class is properly functioning.
2. To create another Java Class, you can repeat the same steps in Part 1 of this tutorial; the only difference is that when in the "New Java Application" window, you will *check* `Create a Main Class` and name your new class `HeronTester`, because your files can't have the same name. This class will be saved in the same folder as your 'Heron" class.
3. Click `Finish` and now you can write the tester code.

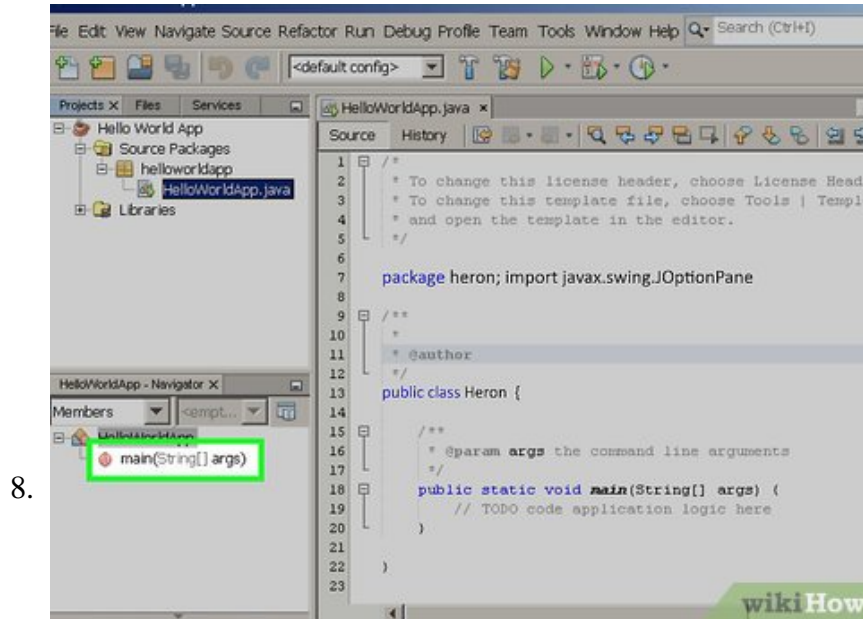


7.

Import swing classes to be able to allow the user to input data in your tester.

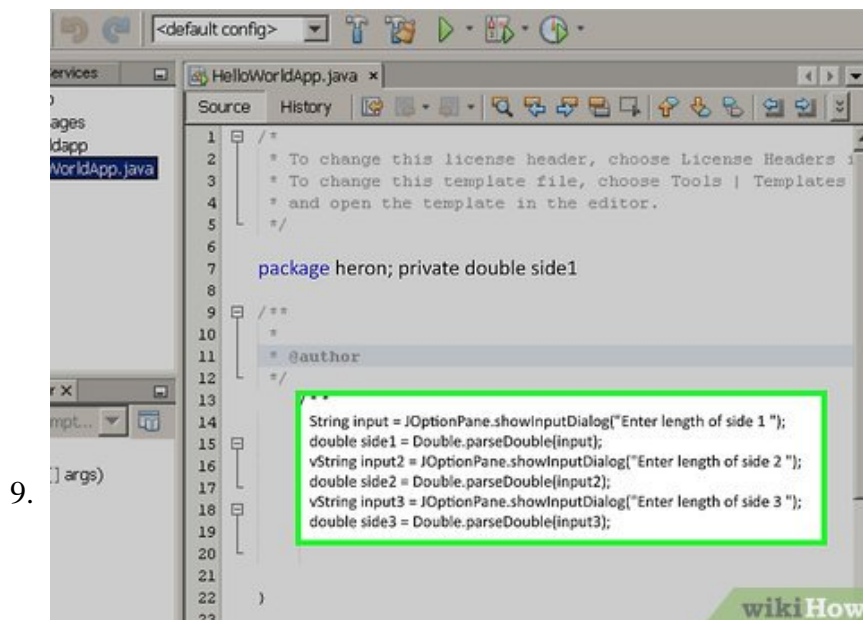
1. Before you start the code, you should import the swing class in order to be able to input data and easily change it.
2. The swing class is: `import javax.swing.JOptionPane`
3. This needs to be placed at the very top above `public class HeronTest`

4. If you are unsure where to put this line of code, you can put it on the very first line above all of your other code.



Add the code to make it your "main" class, if needed. By default, when you created a new java class and checked the box to create the main class, NetBeans will automatically generate the code necessary to consider it the "main" class. If you don't see the code, you can follow these instructions.

1. To create a main method, after the bracket type: `public static void main(String args[]) {`
2. This makes it the first method that will be read and in this case, it is the only method to be read.

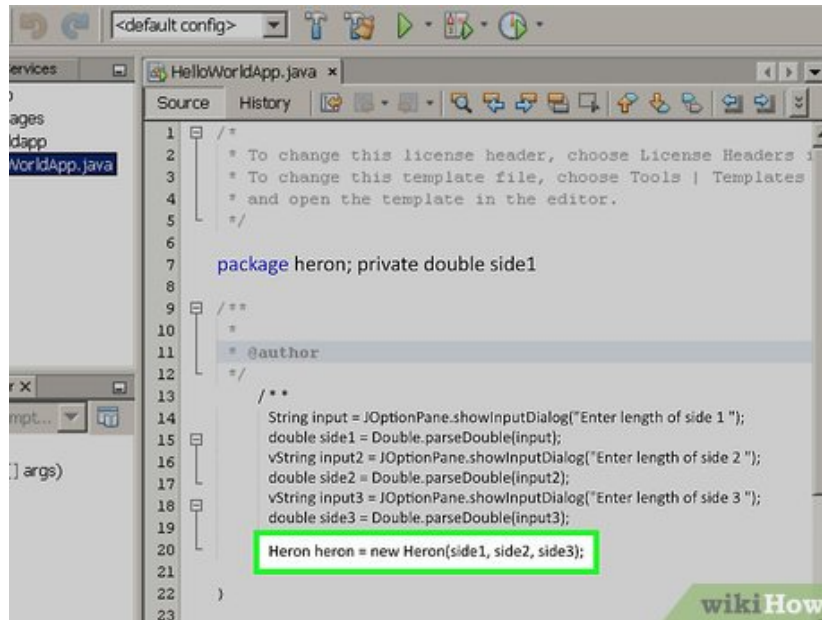


Create the statements that allow the user to input the data.

1. Insert statements to get the lengths of the sides from the user and store them in side1, side2, and side3. This is where we will use the swing class to create boxes to input information.
2. Type the following lines of code below to create input boxes when you run the program.

1. `String input = JOptionPane.showInputDialog("Enter length of side 1 ");`
2. `double side1 = Double.parseDouble(input);`
3. `vString input2 = JOptionPane.showInputDialog("Enter length of side 2 ");`
4. `double side2 = Double.parseDouble(input2);`
5. `vString input3 = JOptionPane.showInputDialog("Enter length of side 3 ");`
6. `double side3 = Double.parseDouble(input3);`

10.

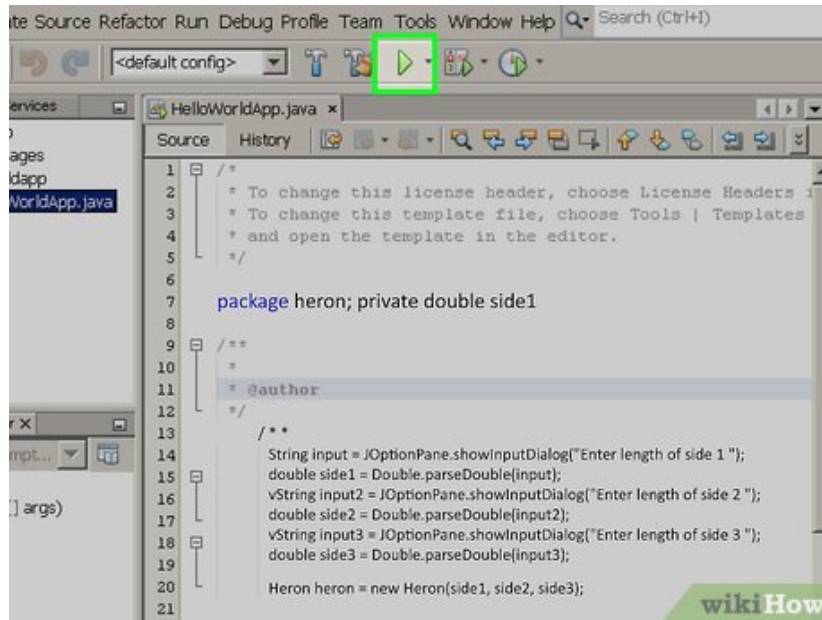


```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6
7  package heron; private double side1
8
9  /**
10 *
11 * @author
12 */
13
14 /**
15 String input = JOptionPane.showInputDialog("Enter length of side 1 ");
16 double side1 = Double.parseDouble(input);
17 vString input2 = JOptionPane.showInputDialog("Enter length of side 2 ");
18 double side2 = Double.parseDouble(input2);
19 vString input3 = JOptionPane.showInputDialog("Enter length of side 3 ");
20 double side3 = Double.parseDouble(input3);
21
22 Heron heron = new Heron(side1, side2, side3);
23
24 */
```

Write the code to create a new Heron object, and print the results in the run console.

1. To do this, Type the name of the object with a capital, the name of this version of the object (can be any name you want) to make it look like the line below:
 1. `Heron heron = new Heron(side1, side2, side3);`
2. To get the line to print we need to use a `System.out.println` statement:
 1. `System.out.println("A triangle with sides of " + heron.toString() + ".\n...has area of " + heron.getArea());`

11.



Run the project!

1. Click the green arrow at the top. (Or right click anywhere in the code and hit `Run`). A box should pop up saying 'Enter length of side 1'. Enter a number and repeat the process for sides 2 and 3.
2. You should then get an output stating 'A triangle with sides of x, x, x... ...has area of x'.

You finished reading the article "**How to Create a Java Class in Netbeans to Compute the Area of a Triangle Using Heron's Formula**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.