Create a Project in jGRASP 1.8.8_19

Here is some help on how to create a project with jGRASP 1.8.8_19
First of all create a folder where you'll save all the class projects. Name the folder CS1400
Then open jGRASP.
If you have a different version than 1.8.8_19 the instructions might need to be adapted.

- On the top menu click **Project**. A menu opens. Click **New**. The *jGRASP: Create Project* wizard opens.
- Use the drop down list next to *Look In* together with the field below to navigate to the CS1400 folder
- In the textbox under Project Name (on the right) enter **Patterns**.
  Make sure the *Create New Directory* checkbox is checked
- Click the **Next** button.
  This leads you to the next page, the *Confirm Project Creation* page.
- Ensure that the checkbox next to *Add Files to Project Now* is checked
- The second checkbox next to *Open UML Window* remains unchecked.
- Click the *Create* button
  The *Add Source Files To Patterns* window opens.
  At this point you have created a project. You see the newly created project with a purple icon in the *Open Projects* window to your left.
- Caveat: you need to navigate to your project again before you can add the files.
  Once you are in the folder named Patterns and you see the project name Patterns.gpj with a purple icon you are at the right place.
- In the textbox next to *File Name* enter **Patterns.java**.
  (feel free to override what was there. It is no longer needed)
- Click the **Add** button. A status message at the bottom of the window will let you know that Patterns.java got added.
  You will also see the new file together with a green icon in the Open Projects window to your left.
- Click the **Done** button.
- Double click Patterns.java in the Open Projects window. A new window appears. jGRASP lets you know that Patterns.java does not exist yet and it asks you whether you would like to create it.
- click the **Create** button.

Now you have an open Java source code file.
First we need to create a class. Remember that the name of a public class needs to match the file name (including upper and lower case).

```
public class Patterns
{
}
```

Here at SLCC we have the coding convention that the opening and closing braces are in separate lines.
Whenever you type an opening brace add the corresponding closing brace in the line below. Then add a new line between the braces and press tab once to indent the code between the braces.

Next we need to create the main method as the entry point of our program.

Inside the body of the class create the following method:

```
public static void main(String[] args)
{
}
```

Make sure that the main method is indented and that the braces are aligned. The indentation should be exactly one tab.

Just for testing purposes we will display a simple pattern of one star surrounded by 2 dashes on each side.
Between the braces of the main method add
`System.out.println("--*--");`
Your code should look now like this:

```
public class Patterns
{
      public static void main(String[] argv)
      {
            System.out.println("--*--");
       }
 }
```

Compile the code by clicking the compile icon (green cross). A message will appear in the Compile Messages window. If the operation completed without errors you are ready to execute your program.
Click the run icon (red running guy).

```
--*--
```

will be displayed in the Run I/O window.

Add the following line below the first print statement:
`System.out.println("-*-*-");`
Compile and run. How does the output change?

Create a pattern that looks like a diamond of stars:

```
--*--
-*-*-
*---*
-*-*-
--*--
```

How would your program change to produce a diamond of circles? Which parts would remain the same? Which parts would change?

Run the program. Adjust the windows so that you can see both the code and the output.
Then let me know that you are finished so I can come and check you off.