Practices - Section 3

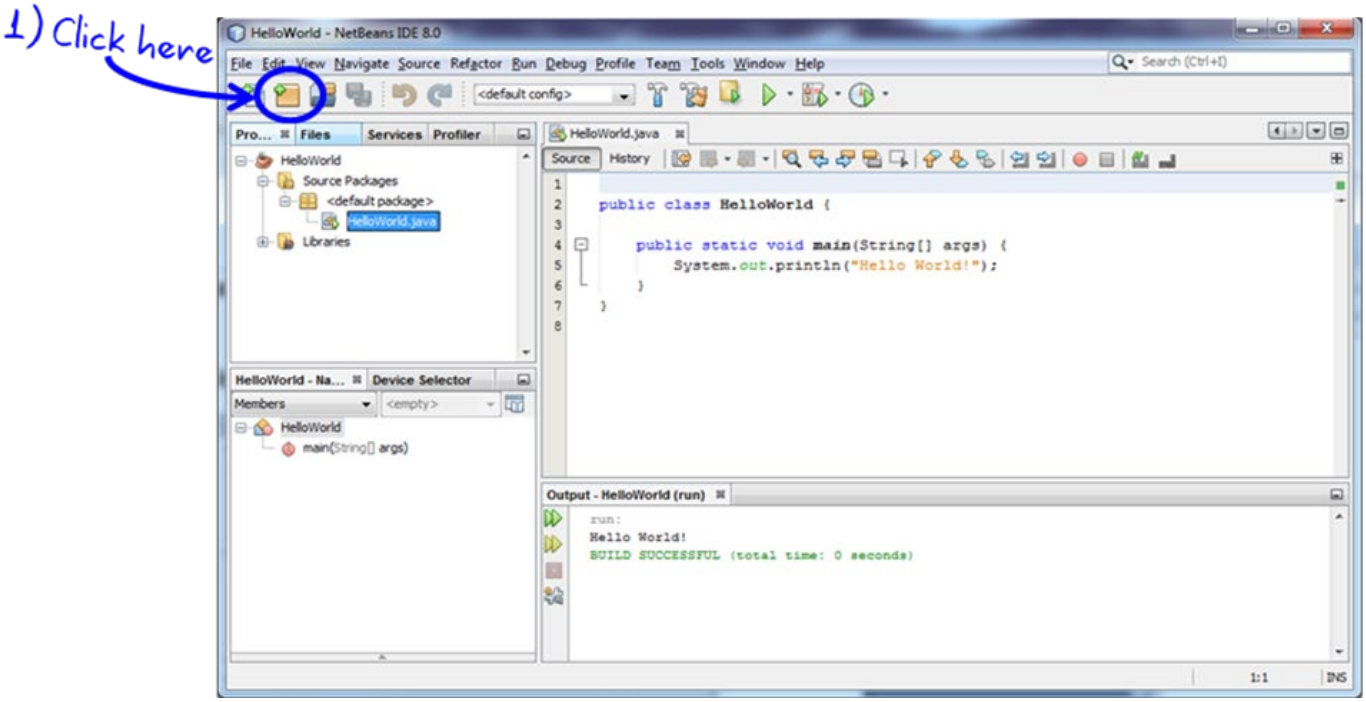
Part 1: Creating a New NetBeans Project.

Overview

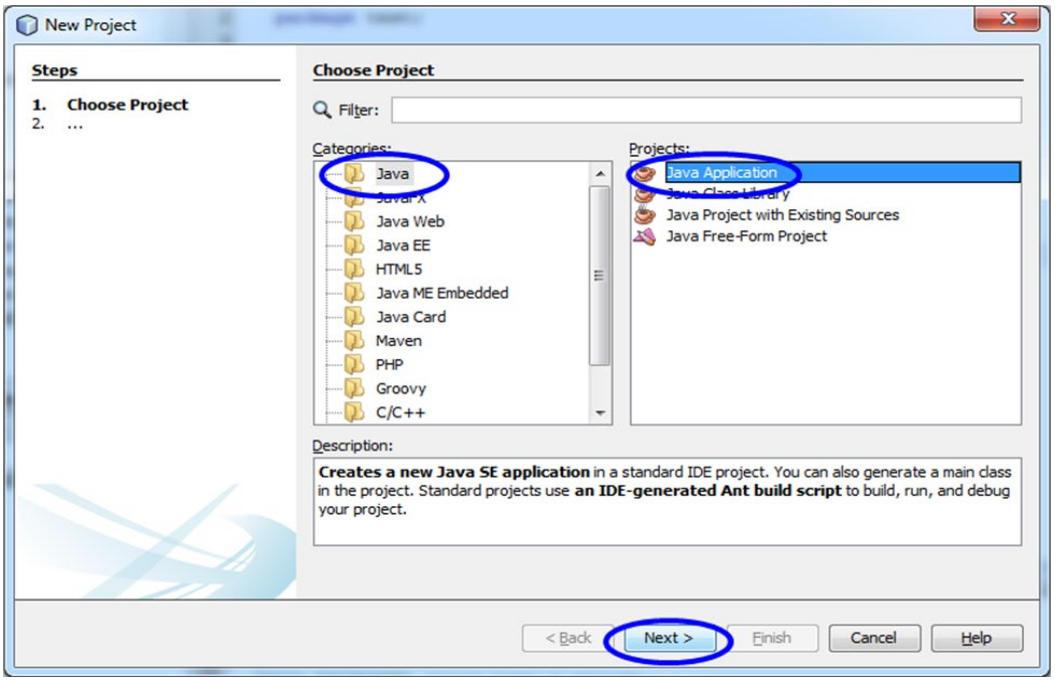
Prior to this practice, you've been importing projects into NetBeans. Have you ever wondered how NetBeans projects were created in the first place? In this practice, you'll create your first NetBeans project. This will be the NetBeans project you'll use to recreate the Java Libs program. You can apply these same steps in future exercises to create your NetBeans projects.

Tasks

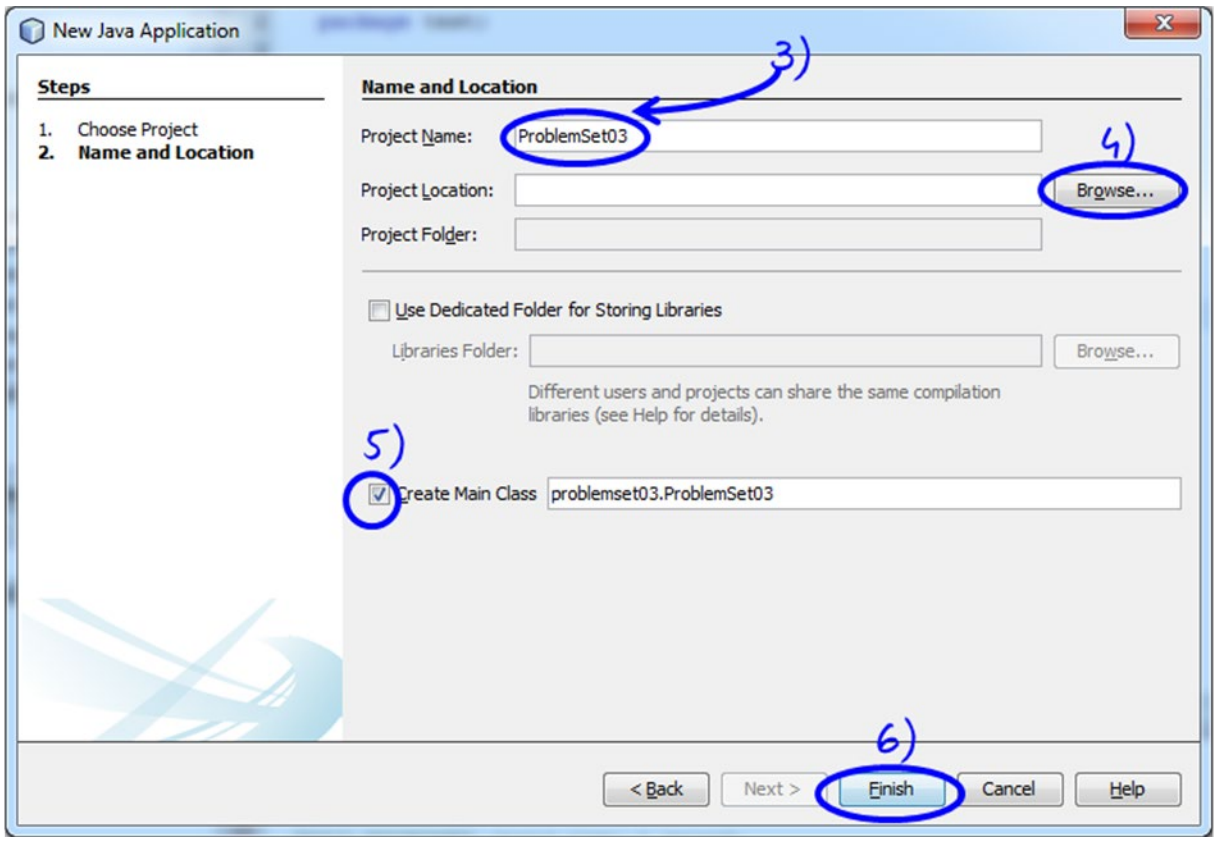
1. Click the **New Project** button in NetBeans: 



2. As your project type, choose “Java Application” and click “Next”:



3. Name your project.
4. Click **Browse** and choose where you’d like the project to be stored on your computer.
5. Be sure to check the box for automatically creating a main method. You want this for ProblemSet03, but not necessarily for every future exercise.
6. Click **Finish**. Your project should open automatically and be ready for editing



Part 1: Creating a JavaLibs Game.

Overview

Section 3 has shown you everything you'll need to recreate a JavaLibs program, similar to the one you played at the beginning of the section. It may take a bit of thinking, but it's a challenge you're definitely ready for. Your program requirements are described below. Good luck, and have fun!

Tasks

Your goal is to create a program similar to JavaLibs. Write a story where certain parts of the resulting story text are modified by the user's input. Prompt the user for various inputs.

You may accept user any number of ways, including a `JOptionPane`, or `Scanner` input from the console. However, choose only one method. Don't use multiple methods of accepting input. Similarly, if you use `JOptionPane` to get input, use `JOptionPane` to show the resulting story.

When you output your story, make sure your all your text is visible at the same time. It's not ok for text to be too long for your computer screen or output window. Your story will need to be spread across several lines instead of being printed in one giant line of output. This helps keep your output clean and your program more user friendly.

It's ok for your program to crash if the user inputs inappropriate data. In other words, It's ok if your program crashes because you've expected the user to input a number, when they've instead input a `String`. We'll cover exception handling later in the course.

Your program must also do the following:

- Accept at least 1 input, to be parsed as a `String`
- Accept at least 1 input, to be parsed as an `int`
- Accept at least 1 input, to be parsed as a `double`
- Use at least 1 input in a question for the user
- Do math with at least 1 `int` input
- Do math with at least 1 `double` input
- Accept at least 10 total inputs

It's ok for this problem set to write your entire program within the main method.