

Lab Assignment #3: Exploring Programmer-Defined Classes

Due Tuesday, 31 May 2016 10:05 am
(Remember, Monday is Memorial Day!)

Please help NCC conserve resources by not printing this document on campus.

Objectives

- Practice creating a Java program using Eclipse
- Practice using documentation to understand programmer-defined classes
- Practice instantiating objects and calling methods on those objects
- Become comfortable dealing with syntax errors in a program, and programs made of multiple classes

Partners

- **Daniel Adrien and Janani Thiruvengadam**
- **Robin Ambooken and Liam Turner**
- **Justin Barratt and Tyler Litz**
- **Heather Bo and Emma Zorfass**
- **Christian Burns and Matthew Sixt**
- **Jonathon Carrera and Chuck Okonkwo Aguolu**
- **Ryan Feraren and Kevin Lopez**
- **Derick Hansraj and Wahab Quazi**
- **Tom Marszal and Sean Rowland**
- **Ali Sharif and Re'and Ward**

Read This First

As I've mentioned before, reading the specification before starting can reduce the possibility of making mistakes during the assignment. This is especially true today, because this assignment is somewhat more complicated than the first two. I strongly recommend you read this whole document before starting Eclipse or making any files, because it may be difficult to correct any careless mistakes in this assignment.

JAR Files

A JAR (Java Archive) file contains executable `.class` files (byte code), and possibly other files like image files. Many user-defined classes can be saved into a JAR file, which can then be accessed by the IDE for use in your own programs.

You will find the file `cardgames.jar` on the Reference Materials portion of my web site. Download it now and save it into your workspace. In this assignment, you will learn how to have Eclipse include it in your project.

Create a New Project

Open Eclipse and choose **File | New | Java Project**. Call the project `lab3`. Make sure the directory that appears is `j:\workspace\lab3`. **Press Next, not Finish**. See Figure 1 on page 3.

The next dialog box that comes up will have multiple tabs. Click the **Libraries** tab to add the JAR file you downloaded before. Click on **Add External JARs** and open the `cardgames.jar` file from your workspace. Now click Finish. See Figure 2 on page 4.

Create a new class in this project by choosing **File | New | Class**. Make sure the *source folder* that appears is `lab3/src`. Type `lab3` in the Package text box, and call the class `Lab3App`.

Writing Java Statements

To use the code in the JAR file, you must write an **import statement**. Type:

```
import cardgames.*;
```

right below the package statement.

Make sure you have appropriate comments at the beginning of the class, similar to those of the first two lab assignments. Leave the description blank for now; you will fill it in

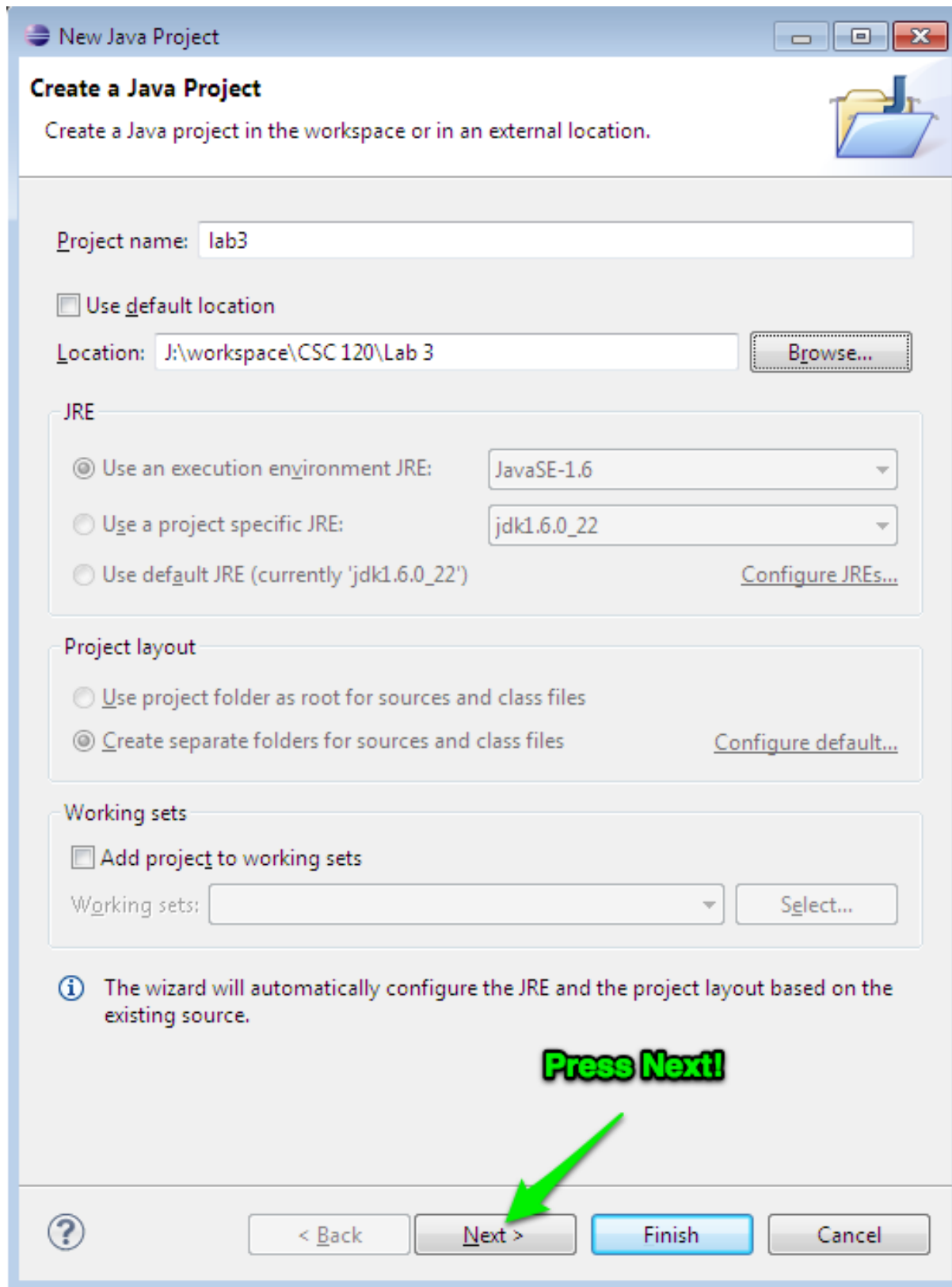


Figure 1: Press Next to Include a JAR File

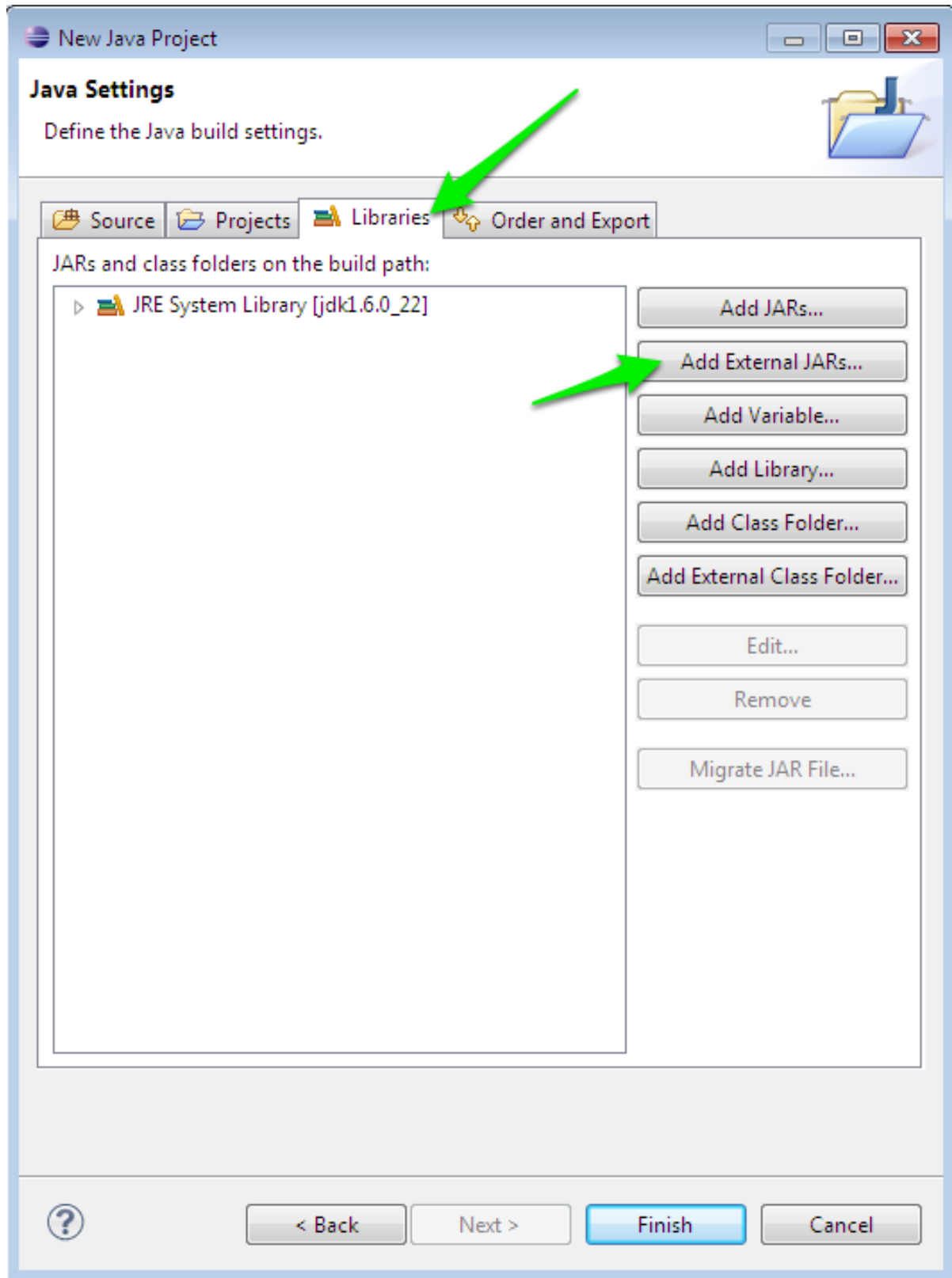


Figure 2: Add a JAR File

later, once you understand what the program does.

Write the main method header inside the class definition. Remember that it must look like this: `public static void main(String args[])`. Include an opening curly brace at the end of this line, and a closing brace on a later line. Create a JavaDoc comment above this header. Next to `@param args`, type `Arguments`.

Use the documentation for the `GUI`, `Deck`, and `Card` classes that is available on the Lab Assignment web page to complete the steps below. You may want to open a blank text file and type your answers to the numbered questions into it as you go, so you can just copy it and paste it into the website when you're done to submit your answers.

1. Place your cursor within the curly braces of the `main` method, and type:
`GUI theWindow = new Deck();`
to declare a reference to a `GUI` object named `newWindow` and instantiate it using the default constructor. A red x will appear in the left-hand margin to indicate that there is a syntax error with this statement. Hover your mouse cursor over the x to see the error message.

Question 1: What is the syntax error message?

Question 2: What do you think this error means?

2. Fix this statement so that it properly instantiates a `GUI` object using the default constructor, and stores a reference to this new object in `theWindow`.
3. Run the program.
4. Write a comment above the statement you just wrote, explaining what the statement does.
5. Close the `GUI` window.
6. Read about the `Deck` class' constructor in the documentation.
7. Declare a reference to a `Deck` object named `theDeck`, and instantiate it using the default constructor. Fix any errors before continuing.
8. Type `theWindow.shuffleDeck();`. This will cause another syntax error.

Question 3: What is the syntax error message?

Question 4: What do you think this error means?

Question 5: In which class is the `shuffleDeck` method defined? What is the header for this method?

9. Fix the call to `shuffleDeck` so that it is called on the appropriate object. Fix any errors before continuing. Write a comment above this statement explaining what it does.
 10. Read about the `dealCard` method in the `Deck` class.
-

Question 6: Are there any parameters specified for this method? If so, what is/are the data type(s) of the parameter(s)?

Question 7: What is the data type of the return value?

11. Declare a variable called `cardOne` so that the value returned from the `dealCard` method can be stored in it. Call the `dealCard` method and store its return value.
 12. Fix any errors before continuing. Write a comment above this statement explaining what it does.
 13. Read about the `toString` method in the `Card` class.
-

Question 8: Are there any parameters specified for this method? If so, what is/are the data type(s) of the parameter(s)?

Question 9: What is the data type of the return value?

14. Write a call to the `toString` method on your `Card` object. Place this call between the parentheses of a call to `System.out.println()`, so that the value returned by `toString` is displayed to the screen.
 15. Fix any errors before continuing.
-

Question 10: Run the program. What is displayed in the message pane at the bottom of the screen?

16. Write a comment above this statement explaining what it does. Close the GUI window.

17. Type `cardOne.showCard(theWindow);`. This will cause another syntax error.
-

Question 11: What is the syntax error message?

Question 12: What do you think this error means?

18. Find the `showCard` method in the documentation, and then fix the call to `showCard` so that it is called correctly, on the appropriate object. Fix any other errors before continuing, and then run the program. Write a comment about this statement. Close the GUI window.
19. Read about the `getValue` method in the `Card` class.
-

Question 13: Are there any parameters specified for this method? If so, what is/are the data type(s) of the parameter(s)?

Question 14: What is the data type of the return value?

20. Call the `getValue` method on the object referenced by `cardOne`. Think about where to store this value.
21. Display the value returned by `getValue` using `System.out.println()`. Use an appropriate label.
-

Question 15: Why do we need both a `toString` method and a `getValue` method for this class? Don't both of them tell us what value the card has?

22. Write comments about the statements you just wrote.
23. Read about the `getPointValue` method in the `Card` class.
-

Question 16: Are there any parameters specified for this method? If so, what is/are the data type(s) of the parameter(s)?

Question 17: What is the data type of the return value?

24. Call the method. Think about where to store the return value.

25. Display the return value with an appropriate label.

Question 18: Why do we need both a `getValue` method and a `getPointValue` for this class? What is the difference between the values that get returned? (You might need to run the program several times to answer this question.)

26. Update this program's description in your initial comment, to describe what this program does.

Output

Your output should consist of three lines: the value and suit of a card, the value of the card (with a label), and the point value of the card (with a label).

Submission

Web-CAT

Upload `Lab3App.java` to Web-CAT to make sure it's running as intended.

My Website

Please upload `Lab3App.java` to the website, and then enter the answers to the numbered questions in the text box.

Please note that these files must be uploaded, and these questions answered, before Tuesday's class starts. Late submissions will not be accepted.