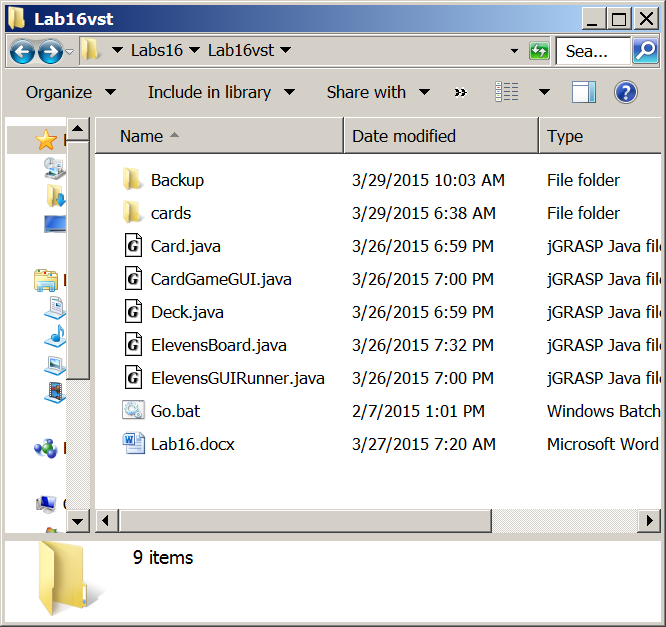
|  |  |
| --- | --- |
| **AP Computer Science** | **Lab 16 Assignment** |
| **The "AP Elevens" Program** | **100 Point Versions** |
| **Assignment Purpose:**  The purpose of this lab is to make the Elevens Solitaire game functional, such that it allows playing the game with the proper game rules. | |

The student starting files for Lab16 allows you to play the solitaire game of Elevens. The folder, shown below, lists all the necessary files for this lab assignment. They are the same files that you used for Chapter16, Experiment3. In that experiment you noticed that the program compiled, and it also executed. Actually, that is a rather nice version since a player will always wins. There are no checks made to see if a player’s move is legal and any number of cards can be removed at will.

For this lab assignment you will need to use the **Card.java**, **CardGameGUI.java**, **Deck.java** and **ElevensGUIRunner.java** files exactly as provided. Make no changes in those files. Completion of this lab assignment revolves around changing of the **ElevensBoard.java** file.



Start by loading the runner program in your IDE, compile, and execute. Anytime you debug, alter, or enhance a program, you must understand what you are working with. The program should compile and execute.

Load the **ElevensBoard.java** in your edit window. ***Remember that this is the only file where you make any changes.*** This is realistic in the real programming world. Programs are very large and are developed by a large team of programmers. No one person writes the entire program. You will be working, creating and testing one segment of the program. Such is this exercise.

Imagine that you have a functional game, but at this stage the game is flawed and allows improper moves to be made.You have identified that the game needs the proper implementation of four methods:

i**sLegal**, **anotherPlayIsPossible**, **containsPairSum11,** and **containsJQK**

These four methods are already located in the **ElevensBoard** class as *stubs*, which is a method with a heading and an empty body. Actually, the methods are not completely empty. Each one includes the **return true;** statement. The four stubs are shown below, minus the comments describing the purpose of each method. The complete comments are included with the Program16 provided Java code files. Read those comments carefully, to determine the intended purpose of the methods. Sample executions are not shown with this assignment description. An earlier experiment showed precisely what such a execution looks like. Once these four methods are properly written the game should play exactly as the version that you played for Experiment2.

|  |
| --- |
| **public boolean isLegal(List<Integer> selectedCards)**  {  /\* \*\*\* TO BE IMPLEMENTED IN LAB16 \*\*\* \*/  return true;  }    **public boolean anotherPlayIsPossible()**   {  /\* \*\*\* TO BE IMPLEMENTED IN LAB16 \*\*\* \*/  return true;  }    private boolean containsPairSum11(List<Integer> selectedCards)   {  /\* \*\*\* TO BE IMPLEMENTED IN LAB16 \*\*\* \*/  return true;  }   **private boolean containsJQK(List<Integer> selectedCards)**   {  /\* \*\*\* TO BE IMPLEMENTED IN LAB16 \*\*\* \*/  return true;  } |

|  |
| --- |
| REALITY CHECK  This is meant to be a realistic lab assignment. Right now the majority of students will have some blank stares wondering how this is expected to be done. You may not have a clue how to start. Welcome to the real world.  Imagine that you have been hired to replace a person who left. The previous person did not finish the **ElevenBoard** class. That is now your job.  You know how the **Card** class and the **Deck** class work. Completion of the required four methods starts by understanding what is already functional in the **ElevensBoard** class. It has methods that have not been described. This was intentional.  You need to analyze what already exists and find the tools (methods) that you will need to use to write these new methods. |