JH4 Homework (100 points)

Create an Eclipse project named JH4_xxxxx where xxxxx is your NetID. For example: JH4_chasselb. Create the programs described below in this project. Fill in the required fields in the JH4_worksheet.txt file. Export your project file and upload it to blackboard as your work for this assignment. The worksheet file will be pasted into the blackboard comments section.

As you write your code make sure you adhere to the standards described in the rubric.

Get started early, and don't skip necessary video/lectures. If you find yourself stuck doing this homework assignment, first consider doing the practice work that would help prepare you for it.

Java practice files can be found on Blackboard.

Make sure you do your Class Participation (10 points)

---

roman_calculator problem (35 points):

Check the instructions online on Blackboard

Enter the output into the appropriate section of the JH3 worksheet.

JING of a Debug session with your RomanCalculator program (20 points).

I want to make sure you have tried out the Eclipse debugger by now. To show that you have used the debugger, I want you to make a JING video (see the link in Learning Units on how to setup and use JING). Your JING video should demonstrate the following with your RomanCalculator program. :

1. That you have set the Step Filters as shown in the Eclipse debugger video.
2. Demonstrate setting a break point and using "Step Over".
3. Demonstrate looking at variables.
4. Stepping into a method.
5. Use the "Step Return" feature
6. Setting another break point and letting Eclipse go to the next break point

Completing this exercise should give a significant Java debugging advantage!!!!!!!!!!

Hangman: object_oriented problem (25 points):

Check the instructions online on Blackboard

Enter the output into the appropriate section of the JH4 worksheet.

Hangman: static_approach problem (10 points):

The modern approach to creating programs is "object oriented" and we took this approach in the previous problem. However, there is an older approach that you can take that uses static methods and data. This approach is a bit old fashion and it is the approach programmers had to take in the "Fortran days".

There are good reasons to use static methods and variables, but normally you will want to take the object oriented approach. The Math class has some good examples of using static: Math.PI, Math.sqrt(), etc.

To emphasize these concepts we created a cute dramatization in: Optional Learning Enrichment: static versus object-oriented approach

Often beginning programmers need to be broken of bad habits ... for example putting all of your code in main. Another bad habit is over using static methods. In this problem, we will have you rework your previous object oriented Hangman solution into a static approach. Normally we want you to use the object oriented approach, but it is good for you to see that Hangman could have been written with the "old fashioned" static approach. To do this, follow these instructions:

- Create a new Hangman in a different package named static_approach
- Add the keyword static to the beginning of the methods: isIn, printCurrStatus, getNextWord, playGame
- Add the keyword static to any instance variables that you have such as: keyboard, rand, etc.
- The structure of your code should look something like:

  ```
  package static_approach;

  ......
  ```

```
public class Hangman
{
    static Scanner keyboard = new Scanner(System.in);
    static Random rand = new Random();

    static boolean isIn(char c, String str)
    {
       .....
    }
    static boolean printCurrStatus(String strToGuess, StringuserInputs)

       .....
    }

    static String getNextWordToGuess()
    {
        ....
    }

    static void playGame()
    {
 .....
    }

    public static void main(String[] args)
    {

        String response="";
        do
        {
            playGame();
            System.out.print("Do you want to play static approach
Hangman again? (y or n): ");
            response = keyboard.next();
        } while (response.charAt(0) == 'y');

        System.out.println("Bye");
    }
}
```

Run your new program and enter the output into the appropriate section of the JH4 worksheet.