# Assignment 9: Do-It-Yourself Rock-Paper-Scissors

## Objectives

* Create a program that will play your own version of Rock-Paper-Scissors
* Create a menu application that loops until the user quits.
* Keep track of your wins, the computer wins, and the ties.
* Create methods for the different aspects of the software.

## Motivation

In the game Rock-Paper-Scissors, two people (in secret) select on of the three items (Rock, Paper, or Scissors). They compare their selection. Rock beats Scissors, Paper beats Rock, Scissors beats Paper. If they select the same item, it's a tie.

For this assignment, I'd like for you to make your own version of the game with your own set of three items. They could be "dogs, cats, gerbils" or "canoeing, rafting, kayaking" or "planes, trains, and automobiles". For whatever three items you pick, you have to select one item to beat another item for all three items (much like Rock-Paper-Scissors). It doesn't have to make sense. Why does paper beat rock, anyway? Don't answer that.

You'll need to present a menu to allow the user to select their item and you'll need to have the computer to randomly select an item. (This will require the **Random** class.)

After each item is selected, determine who won (or if there is a tie because you both picked the same item). You'll need variables to keep track of the results. The results should be displayed after every round. (See my example.)

Also after every round, you should ask the user if they would like to continue.

When you are done, you should refactor your code using NetBean's "Introduce Method" feature to break the game into private helper methods. The final source code should be a readable document.

This is a variation on problem 17 on page 317 in the textbook. Like the textbook, I use the **Random** class to select a number from 1 to 3 for the computer's selection.

In fact, I am going to provide my entire main method. Your code should be as readable as mine.

public static void main(String[] args) {  
 String response = "yes";  
  
 Scanner keyboard = new Scanner(System.in);  
 Random rng = new Random();  
  
 greetTheUser();  
 displayTheRules();  
  
 while ("yes".equals(response.toLowerCase())) {  
 int cpuChoice = getCPUChoice(rng);  
   
 diplayTheItemChoices();  
 int userChoice = getUserChoice(keyboard);  
  
 displayTheComputerChoice(cpuChoice);  
 displayWhoWon(userChoice, cpuChoice);  
 response = askUserIfTheyWouldLikeToPlayAgain(keyboard);  
 }  
   
 displayClosing();  
}

## Instructions

Name your project FirstnameLastnameAssignment9

Have your program do the following.

1. Greet the user: "Welcome to Dr. Church's Dog-Cat-Gerbil Game."
2. Display the rules for your version of the game.
3. Loop while the user still wishes to play:
   1. Randomly select the CPU's choice
   2. Display the item choices for the user.
   3. Get the user's choice from the keyboard.
   4. Display who won.
   5. Ask the user if they would like to play again.
4. Display a closing message. "Thank you for playing!"
5. Once you are done with this, make sure it works by playing your game.
6. Refactor the essential parts of the code. All of these methods will be kept as **private**.
   1. The greeting.
   2. The rules.
   3. The CPU choice
   4. The prompt message for the user's selection.
   5. The keyboard input of the user's selection.
   6. The display the computer's choice.
   7. The display for who won.
   8. The display prompt to ask the user if they would like to play again.
   9. The closing message.

## Documentation

Your source code must include the following documentation:

* Your name
* The class (CS 2070) and the section number (on ground is 08, online is W1).
* The date on which you turned in the assignment.
* A short description of the software. Usually a sentence or two is sufficient.

## Example Run

Welcome to the game of Dog-Cat-Gerbil.  
Gerbil beats Cat.  
Cat beats Dog.  
Dog beats Gerbil.  
Let's play! Enter a guess:   
1: Dog  
2: Cat  
3: Gerbil  
1  
The computer picked cat.  
You won!  
Should we play again? [Yes]   
yes  
Let's play! Enter a guess:   
1: Dog  
2: Cat  
3: Gerbil  
2  
The computer picked gerbil.  
Computer won.  
Should we play again? [Yes]   
yes  
Let's play! Enter a guess:   
1: Dog  
2: Cat  
3: Gerbil  
3  
The computer picked cat.  
Computer won.  
Should we play again? [Yes]   
no  
Thanks for playing!

## Turning it in.

To turn in your application, find the folder containing your entire project (not the folder with the "java" file), zip it up, and turn it in.