CSE 11 **START EARLY!**
Spring 2015
Programming Assignment #1 **100 Pts**

Due: 10 April 2015 at 11:59pm (2359)
Covers Chapters: ZY 1-3

This is a combination of a programming assignment and ungraded exercises

Exercises are optional, not graded.

Exercise #1: Make certain you can log into the campus lab computers in the CSE building. All of your programming should be done in Linux. ALL PROGRAMS MUST BE TURNED IN FROM LAB MACHINES.

Exercise #2: If you have a Windows or Mac laptop and would like to have Linux environment for development, please go to the class website and look at the page called "Run a Development VM on your Laptop"

Exercise #3: If you do not know either `emacs` or `vim` , then run through the tutorial called "`vimtutor`" in the lab

Exercise #4:

Create a subdirectory in your home area called PR1

```
$ cd $HOME
$ mkdir PR1
$ cd $HOME
```

Download `HelloWorld.java` and `Twist.java` from the class web site and place them into your PR1 subdirectory.

Following the directions in the comments of each program, compile and run these programs

PROGRAMMING Assignment (100 pts) : Rock, Paper, Scissors

You are to create a program called `RPS.java.` You will use random number generator from `java.util.Random` to generate each player's choice. The program first prompts for integer input to be used the numeric seed for the random number generator. Then the program uses the generator to have each of two players choose Rock, Paper, or Scissors. It prints out what each player chose and then prints out who won the game. If player 1 wins, it prints out "Player 1 Wins", if player 2 wins, it prints out "Player 2 Wins". If neither player wins it prints out "Nobody Wins". The program exits after playing a single game.

The rules of Rock, Paper, Scissors: Rock beats Scissors, Scissors beat Paper, and Paper beats Rock.

Here are a few runs of the program. **Boldface** indicates input by the user. `[PR1]$` is the command-line prompt.

```
[PR1]$ java RPS
Enter Seed : 15
Player 1 : rock
Player 2 : scissors
Player 1 Wins
[PR1]$

[PR1]$ java RPS
Enter Seed : 25
Player 1 : scissors
Player 2 : rock
Player 2 Wins
[PR1]$

[PR1]$ java RPS
Enter Seed : 8
Player 1 : paper
Player 2 : paper
Nobody Wins
[PR1]$
```

Key requirements of your program

1) You must initialize your random number generator with the number input by the user (the seed). You should only create one instance of a Random number generator.

2) Print out each player's choice just as above. Use your seeded Random instance to create each player's choice. Include the ":". Put a space on either side of the ":". This will make our autograder more reliable. Do NOT put any extra output. No extra lines. The game is played once.

3) Print the following statement after the players have chosen rock,paper, or scissors
   a) "Nobody Wins", if both players chose the same thing
   b) "Player 1 Wins", if player 1 is the winner
   c) "Player 2 Wins", if player 2 is the winner
4) The class must be called RPS and must be in the file RPS.java
5) Do not add any extra lines of output. Your output should look just like the examples above.
6) You must use constants in your program (variables declared as `final`). Don't use so-called "magic numbers", instead define a constant and use that in your program

Formatting/Commenting Requirements of your Program

1) Your code must be properly indented. You may uses spaces or tabs. It is recommended in this class that you use TABS. (10 Points)
2) You must put in comments of your program at the top of the file (10 Points)

```
Name: <Your Full Name>
Email: <Your UCSD Email>
ID:  <Your Student ID, e.g. A123456789>
```

Hints:

1) A `return` statement can be used to stop execution of your main method with executing any

further statements

2) Build your program in stages. You don't have to write the entire program before you begin to debug. Write part of the program, test it, then add another part, test it, and so on.
3) Think how you can "convert" a randomly generated integer to Rock, Paper, or Scissors.

Turning in your Program

YOU MUST BE ON THE LAB MACHINES FOR THIS TO WORK. PLEASE VERIFY WELL BEFORE THE DEADLINE THAT YOU CAN TURNIN FILES

You will be using the "bundlePR1" program that will turn in the following file

`RPS.java`

No other files will be turned in and they **must be named exactly as above.** BundlePR1 uses the department's standard turnin program underneath.

To turn-in your program, you must be in the directory that has your source code and then you execute the following

```
$ /home/linux/ieng6/cs11s/public/bin/bundlePR1
```

Sample output of a successful turnin

```
$ /home/linux/ieng6/cs11s/public/bin/bundlePR1
Good; all required files are present:

RPS.java

Do you want to go ahead and turnin these files? [y/n]y
OK.  Proceeding.

Unpacking files ... tar xfBp /tmp/cs11w.tar --dir=/tmp/tmpg28LKG
Copying support files
Checking Compilation of RPS.java ...
Compilation of RPS.java OK
...SUCCESS

Performing turnin of approx. 3929 bytes (+/- 10%)
Copying to /home/linux/ieng6/cs11s/turnin.dest/cs11s.PR1
.
Done.
Total bytes written: 3584
Please check to be sure that's reasonable.
Turnin successful.
```

Sample output of an unsuccessful turnin (Compiler error)

```
$ /home/linux/ieng6/cs11s/public/bin/bundlePR1
Good; all required files are present:

RPS.java
```

```
Do you want to go ahead and turnin these files? [y/n]y
OK.  Proceeding.

Unpacking files ... tar xfBp /tmp/cs11w.tar --dir=/tmp/tmpsKoKce
Copying support files
Checking Compilation of RPS.java ...
Compilation of RPS.java failed
Failed Compilation Test
=== Errors seen while Compiling ===
RPS.java:9: error: ';' expected
import java.util.Random
                       ^
1 error
=== End of Errors seen while Compiling ===
COMPILER ERRORS, You CANNOT turn in
```

You can turn in your program multiple times. The turnin program will ask you if you want to overwrite a previously-turned in project. **ONLY THE LAST TURNIN IS USED!**

Suggestion: if you have classes that compile and do some or most of what is specified, turn them in early. When you complete all the other aspects of the assignment, you can turn in newer (better) versions.

Don't forget to turn in your best version of the assignment.

Frequently asked questions

1) **Does my program have to achieve the identical results of the sample output when the seeds are the same?** *No. You may be using your Random number generator slightly differently than the code used in the example output. Your code should give the same output when using the same seed.*
2) **How important are the colons and fields in the comments?** *Critical. You will lose points if you don't have a : separating your Name from your full name, player's choices. etc.*
3) **What if my program doesn't compile, can I get partial credit?** *No.*
4) **I program in an IDE like Dr. Java, do I have to use the command line**? *Yes.*
5) **Do I need to check for bad input (like typing in a string of letters instead of an integer)?** *No.*
6) **How will this be graded?** *20 points of your 100 points are identified above. We will check the form of your program and look for correct functionality. For example it should generate random choices. We'll check that for randomness. If we run your program 100 times, about 1/3 of the time, rock should be selected, about 1/3 of the time paper should be chosen and about 1/3 of the time scissors should be chosen. We'll also check that you get the right answer for who wins each game*
7) **Can I get partial credit?** *If your program compiles, yes. If your program does not compile, you cannot turn it in.*

START EARLY!   ASK QUESTIONS!