



Assignment 2

Due date: Feb. 11, 11:55PM EST.

You may discuss any of the assignments with your classmates and tutors (or anyone else) but **all work for all assignments must be entirely your own**. Any sharing or copying of assignments will be considered cheating.

You should not use any features of Java that have not been covered in class. If you have doubt if you are allowed to use certain structures, just ask your instructor.

Problem 1 (30 points): Two Dice Simulation

Rewrite the `Dice.java` program from one of the workseets so that it simulates rolling of two fair four-sided dice. The possible outcomes are the sums ranging from 2 to 8 (so you should have seven lines of output instead of four that were in `Dice.java`). Notice that you cannot simulate it by just generating a random number from 2 to 8, you need to simulate two different dice and add the values (make sure you understand why). Your output should be formatted in the same way as the output in `Dice.java` program, i.e. there should be always two numbers printed after the decimal point and the decimal points should be aligned. See the sample output below.

This program does not take any input from the user.

Call your file: `TwoDice.java`.

Output:

```
2 was rolled 6.16% of times
3 was rolled 12.48% of times
4 was rolled 18.89% of times
5 was rolled 24.99% of times
6 was rolled 18.73% of times
7 was rolled 12.44% of times
8 was rolled 6.31% of times
```

Problem 2 (30 points): Characters

Write a program that prompts the user for a non-negative integer and then displays its hexadecimal equivalent and the Unicode character that corresponds to that integer. Your program should verify that the value enter by the user is non-negative. The program should also display some additional interesting information about the character. Use the `Character` class to decide what information you want to display (it is up to you what you chose to use).

See sample runs below for the output formatting.

Use <https://docs.oracle.com/javase/8/docs/api/java/lang/Character.html> to learn more about all the "tools" available in the `Character` class.

Use <https://docs.oracle.com/javase/8/docs/api/java/util/Formatter.html> to learn about many different format specifiers that can be used with `printf()` method. You need this to figure out how to print a hexadecimal number. HINT: you do not need to do any math to convert from decimal to hexadecimal; just use appropriate format specifier for `printf()`.

Call your file: `CharacterCheck.java`.

```
Enter an integer: 15
15 (0xF) is Unicode-16 for character ''
it is an ISO control character
```

```
Enter an integer: 51
51 (0x33) is Unicode-16 for character '3'
it is a digit
```

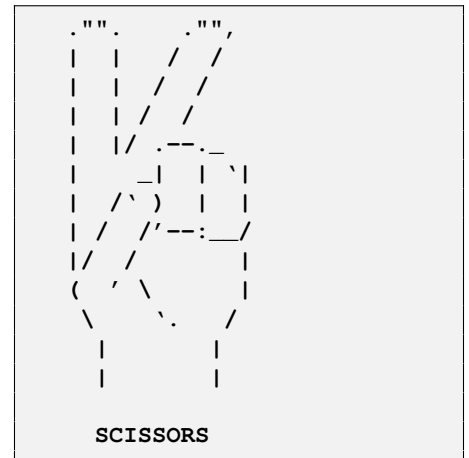
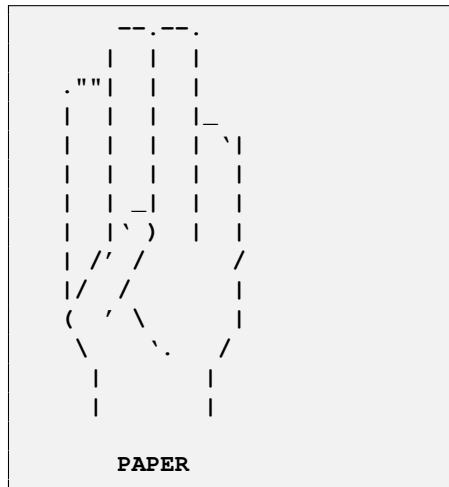
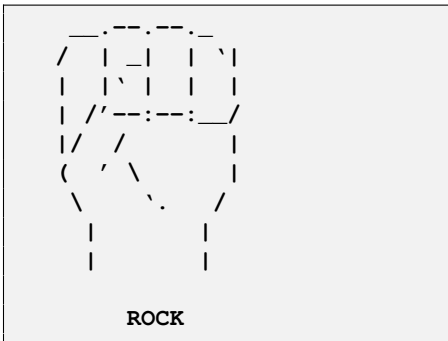


```
Enter an integer: 69
69 (0x45) is Unicode-16 for character 'E'
it is a letter
it is a upper case letter
```

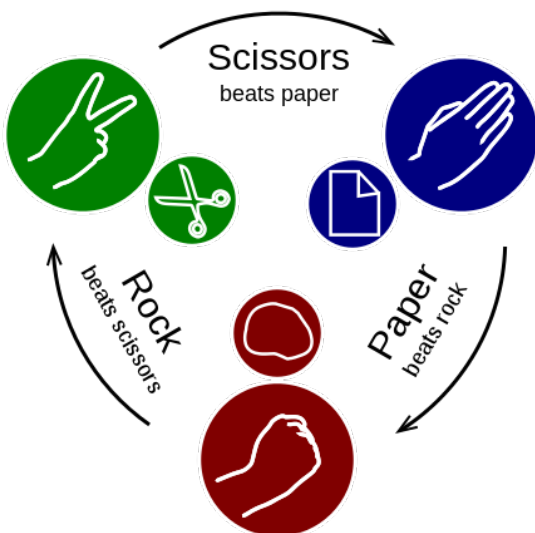
```
Enter an integer: 946
946 (0x3B2) is Unicode-16 for character 'β'
it is a letter
it is a lower case letter
```

```
Enter an integer: 19989
19989 (0x4E15) is Unicode-16 for character '丕'
it is a letter
it is an ideograph
```

Problem 3 (40 points): Rock Paper Scissors



In this problem you will simulate the hand game called Rock-Paper-Scissors. This game is usually played between two people. Your version is played between the computer and a human player. The human and the computer make a choice between one of the three options: rock, paper or scissors. The following graphics shows the interaction between three game elements:



"Rock-paper-scissors" by Enzoklop - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons - <https://commons.wikimedia.org/wiki/File:Rock-paper-scissors.svg>. Accessed June 2015.



The program should prompt the user for her/his choice, then randomly generate one of the three options, determine the winner of the game and display the result.

Here is a detailed outline of the program:

- Display the welcome banner.

```
=====
Welcome to the Rock-Paper-Scissors game!
=====
```

- Allow the player to select their element: rock, paper or scissors. The selection should be done by the first letter of the element (both uppercase and lowercase letters should be accepted).

```
Select your element:
R/r - rock
P/p - paper
S/s - scissors
```

- If the player did not enter a valid selection, **the game ends immediately**. Display the message `ERROR: you did not select a valid element`.
- Tell the player what their selection was based on the letter they entered: `You chose rock`, `You chose paper`, `You chose scissors`.
- Randomly generate the computer's element (this means generating one of three possible values) and display it on the screen: `Computer chose rock`, `Computer chose paper` or `Computer chose scissors`.
- Determine who the winner is assuming the following rules:
 - rock beats scissors
 - paper beats rock
 - scissors beats paper

The displayed message should contain the winner and the rule:

```
Computer won! (scissors beats paper)
```

or

```
You won! (rock beats scissors)
```

(Make sure you replace the rule with the one appropriate for the given game.)

or

```
It's a tie.
```

- The program terminates at the end of the game.

Call your file: `RockPaperScissors.java`.



Here are some sample runs of the program:

Assume user enters 's' for scissors, the computer generates rock.

```
=====
Welcome to the Rock-Paper-Scissors game!
=====

Select your element:
R/r - rock
P/p - paper
S/s - scissors

s

You chose scissors.
Computer chose rock.

Computer wins! (rock beats scissors)
```

Assume user enters 'P' for paper, the computer generates rock.

```
=====
Welcome to the Rock-Paper-Scissors game!
=====

Select your element:
R/r - rock
P/p - paper
S/s - scissors

P

You chose paper.
Computer chose rock.

You win! (paper beats rock)
```

Grading

Does the program compile? If not, you will lose all the points for that problem.

Is the program properly documented? (worth approximately 20% of each problem)

Proper documentation at this point in the course includes:

- preamble with the name of the author, date of creation and brief description of the program (the description should specify what the program does, not that it is a solution to problem 1 of homework 1);
- appropriately chosen variable names, i.e., descriptive names (a good name for the variable that stores the bonus amount in the last problem is `bonus`, not `x`);
- comments inside the code describing steps needed to be taken to accomplish the goal of the program;
- appropriate formatting, indentation and use of white space to make the code readable.

Remember that the code is read by humans and it should be easy to read for people who were not involved in its development.

Is the program well developed and correct? (worth approximately 80% of each problem) Make sure you create variables of appropriate types, use control statements (conditionals and loops) that are appropriate for the task, accomplish your task in a well designed and simple way (not a convoluted algorithm that happens to produce the correct output for some unknown reason). You should also design a friendly and informative user interface.



Make sure that your program produces valid results that follow the specification of the problem every time it is run. At this point you can assume a "well behaved user" who enters the type of data that you request. If the program is not completely correct, you get credit proportional to how well it is developed and how close you got it to the completely correct code.

What and how to submit?

You should submit three source code files combined into a single **zip** file to NYU Classes. Do not submit all the files that Eclipse creates, just the source code files that have `.java` extensions. Name your classes as specified in the problems.

If you wish to use your (one and only) freebie for this project (one week extension, no questions asked), then complete the form at <http://goo.gl/forms/fpUJrF64b5> **before the due date for the assignment**. All freebies are due seven days after the original due date and should be submitted to NYU Classes.

Questions

Post any questions you have regarding this assignment to Piazza under the "homeworks" topic.