Assessment: Basic Programming Concepts

Re-submit Assignment

Due May 14, 2018 by 9am **Points** 100 **Submitting** a website url
Available after May 10, 2018 at 5pm

Overview

This purpose of this assessment is to demonstrate your proficiency in basic Java syntax involving console input and output, variables, flow of control statements, and expressions. These core concepts are the building blocks for more complicated code that is to come.

Exercises

There are four exercises in this assessment:

1. Rock, Paper, Scissors
2. Dog Genetics
3. Healthy Hearts
4. Summative Sums

> Rock, Paper, Scissors is your priority. Focus on that first.

Rock, Paper, Scissors

In this lab, you will write a program that plays the game Rock, Paper, Scissors.

Your Task List:

1. Create a flowchart.
2. Show your flowchart to your instructor.
3. After receiving approval for your flowchart, begin writing code.

Rules

The rules of the game are as follows:

1. Each player chooses Rock, Paper, or Scissors.
2. If both players choose the same thing, the round is a tie.
3. Otherwise:
    a. Paper wraps Rock to win
    b. Scissors cut Paper to win
    c. Rock breaks Scissors to win

Requirements

This program will be a Java Console Application called RockPaperScissors.

- The program first asks the user how many rounds he/she wants to play.

  - Maximum number of rounds = 10, minimum number of rounds = 1. If the user asks for something outside this range, the program prints an error message and quits.
  - If the number of rounds is in range, the program plays that number of rounds. Each round is played according to the requirements below.
- For each round of Rock, Paper, Scissors, the program does the following:
  - The computer asks the user for his/her choice (Rock, Paper, or Scissors).
    - Hint: 1 = Rock, 2 = Paper, 3 = Scissors
  - After the computer asks for the user's input, the computer randomly chooses Rock, Paper, or Scissors and displays the result of the round (tie, user win, or computer win).
    - Hint: use the Random class.
- The program must keep track of how many rounds are ties, user wins, or computer wins.
  - Hint: Create three variables to keep track of these items and update them correctly after each round.
- The program must print out the number of ties, user wins, and computer wins and declare the overall winner based on who won more rounds.
- After all rounds have been played and the winner declared, the program must ask the user if he/she wants to play again.
  - If the user says No, the program prints out a message saying, "Thanks for playing!" and then exits.
  - If the user says Yes, the program starts over, asking the user how many rounds he/she would like to play.

Dog Genetics

Ever heard of those places that you can mail in some of your dogs hair, and they'll send back a report after doing a genetic analysis on it to tell you what kind of dogs are in your most believe pet's ancestry?

Well, we don't know how to do that. But we DO know how to generate random numbers.
And half the time, that's good enough. Especially for the Internet.

What You Should Do

Filename: DogGenetics.java

- Write a program that asks the user for the name of their dog, and then generates a fake DNA background report on the pet dog.
- It should assign a random percentage to 5 dog breeds (that should add up to 100%!)

What You Should See

```
--- exec-maven-plugin:1.2.1:exec (default-cli) @ Unit1 ---
What is your dog's name? Sir Fluffy McFlufferkins Esquire
Well then, I have this highly reliable report on Sir Fluffy McFlufferkins Esquire's pres
tigious background right here.

Sir Fluffy McFlufferkins Esquire is:

61% St. Bernard
2% Chihuahua
29% Dramatic RedNosed Asian Pug
1% Common Cur
7% King Doberman
```

```
Wow, that's QUITE the dog!
```

Healthy Hearts

What You Should Do

Filename: HealthyHearts.java

Make a simple calculator that asks the user for their age. Then calculate the healthy heart rate range for that age, and display it.

- Their maximum heart rate should be 220 - their age.
- The target heart rate zone is the 50 - 85% of the maximum.

What You Should See

```
--- exec-maven-plugin:1.2.1:exec (default-cli) @ Unit1 ---
What is your age? 50
Your maximum heart rate should be 170 beats per minute
Your target HR Zone is 85 - 145 beats per minute
```

Summative Sums

What You Should Do

Filename: SummativeSums.java

Write a static method that takes in an array of ints, adds them together, and returns the result. Call your new method inside a main method and print out the result for the following three example arrays:

```
{ 1, 90, -33, -55, 67, -16, 28, -55, 15 }
{ 999, -60, -77, 14, 160, 301 }
{ 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130,
140, 150, 160, 170, 180, 190, 200, -99 }
```

What You Should See

```
--- exec-maven-plugin:1.2.1:exec (default-cli) @ Unit1 ---
#1 Array Sum: 42
#2 Array Sum: 1337
#3 Array Sum: 2001
```

Submitting Your Assessment

Your instructor will provide instructions on how to submit your work to a GitHub classroom.

After submitting your work, schedule a time with a staff member to review your code. If you are attending the Guild in person, your code will be reviewed during the weekly code review.

Be prepared to answer questions about your code and thought processes when you complete the code review.

Basic Programming Concepts 2.4

Criteria	Ratings			Pts
Flowchart The flowchart for Rock, Paper, Scissors reasonably reflects the flow required for the activity.	**15.0 pts** **Meets Expectations**	**8.0 pts** **Needs Improvment**	**0.0 pts** **No Submission**	15.0 pts
Application Name Application is named correctly.	**5.0 pts** **Meets Expectations**	**3.0 pts** **Needs Improvment**	**0.0 pts** **No Submission**	5.0 pts
Variables Apprentice names and uses variables correctly.	**10.0 pts** **Meets Expectations**	**6.0 pts** **Needs Improvment**	**0.0 pts** **No Submission**	10.0 pts
Loops Apprentice uses loops correctly.	**10.0 pts** **Meets Expectations**	**6.0 pts** **Needs Improvment**	**0.0 pts** **No Submission**	10.0 pts
Conditionals Apprentice uses conditionals correctly.	**10.0 pts** **Meets Expectations**	**6.0 pts** **Needs Improvment**	**0.0 pts** **No Submission**	10.0 pts

Criteria	Ratings			Pts
Methods Apprentice uses main methods correctly.	**10.0 pts** **Meets Expectations**	**6.0 pts** **Needs Improvment**	**0.0 pts** **No Submission**	10.0 pts
Arrays Apprentice uses arrays correctly.	**10.0 pts** **Meets Expectations**	**6.0 pts** **Needs Improvment**	**0.0 pts** **No Submission**	10.0 pts
Operators Apprentice uses arithmetic operators correctly.	**10.0 pts** **Meets Expectations**	**6.0 pts** **Needs Improvment**	**0.0 pts** **No Submission**	10.0 pts
Debugging Apprentice can explain the mechanics of debugging.	**10.0 pts** **Meets Expectations**	**6.0 pts** **Needs Improvment**	**0.0 pts** **No Submission**	10.0 pts

Criteria	Ratings			Pts
Code Style Code is written with appropriate indents, naming conventions, and comments so that other developers can read the code easily	**10.0 pts** **Meets Expectations**	**6.0 pts** **Needs Improvment**	**0.0 pts** **No Submission**	10.0 pts
Total Points: 100.0				