# Documentation for Assignment: Basic Programming Concepts

This assignment contains four programs:

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

All the programs are in IntelliJ project AssessmentBasicProgrammingConcepts zip file under package com.contiero.\*

Rock, Paper, Scissors

**Implementation details**

*Start*

The game starts by asking the user how many rounds she wants to play. An integer variable named ***rounds***is used to store this and an ***if***check ensures *range* is within the range [1,10] before going ahead. This can be done right at the beginning of the code. If the check passes, the game is played *round* times. This whole game needs to be wrapped by a loop because the user is asked if he or she wants to play again, repeating the process of the Rounds.

*The Rounds*

The game is played ***rounds***times so a **for** loop with a variable ***i*** is used around a single round, counting how many times a single game was played. The loop will stop when ***i*** reaches ***rounds***. A few more **int** variables are required to store the winners: ***user, comp, tie***, representing how many times the user won or the computer won and if there was a tie between them. Once the rounds are over the winner is declared. we calculate the winner by comparing the ***user, comp* variables.**

*The Game*

A single round of the game starts by asking the user for his choice: 1-rock, 2-paper, 3-scissors and stores it in the variable ***userChoice.*** Then, a random generator produces the choice for the computer and stores it in ***compChoice.*** The next step is to compare the various combinations for the game: ties, computer wins, user wins. For each win, there will be an increment variable ***comp++, user++, tie++*** respectively and this will be presented to the user later on.

**Flowchart**

****

Dog Genetics

This program produces a fake report about the DNA background of a dog.

The user enters the dog’s name and a friendly message is displayed followed by the breakdown of the five main breeds from which this dog has genes from. The breakdown is presented in terms of percentage of the breeds.

The implementation follows a linear procedural path, without if statements, a straightforward calculation.

It starts by prompting the user to enter the dog’s name and uses a Scanner instance to collect the name. A random number generator is used to produce up to 10000 genes belonging to each of the breeds. The total number of genes represent 100% genome for this dog. The percentage contribution for each breed is then calculated, for example: 100\*stBernard / totalGenes, which gives the integer percentage for St. Bernard.

The report is presented to the user with percentages of St. Bernard, Chihuahua, Dramatic RedNosed Asian Pug, Common Cur and King Doberman.

Healthy Hearts

The program calculates the healthy heart rate for the user given its age.

It starts by asking the user for their age using a Scanner and computes maximum heart rate as 220 minus the given age.

Finally, it presents the healthy heart rate zone for aerobic exercises as 50% to 85% of the maximum. The calculation uses doubles for the percentages and cast it back to integer for presentation.

Summative Sums

The program implements a static method to calculate sums of an array of integers. It takes an array of **ints** and loops over each element adding each one to an accumulator, returning it to the user.

The function is used in **main** method with the provided arrays as an example.