

Java for Non Majors

CGS 3416: Fall 2019

Department of Computer Science, Florida State University

Homework 4: 100 points

Due date: 11:59 PM 10/18/2019

1 Objective

This assignment will consist of writing two small programs that involve practice writing and calling methods, using java built-in libraries.

2 Requirements

- The required tasks must be performed with the methods specified (not just with a single main() routine)
 - Note that each program requires the writing of a method, and a main routine to test that method.
 - Each method should do exactly the task specified in item 1 of each program
 - Item 2 of each program specifies what to do in main() – this will always involve CALLING the method, sometimes more than once
 - Note that there is NO keyboard-input/screen-output specified in the methods themselves (i.e. "return" does not mean "print") – this means you should NOT have any print statements or Scanner usage inside these named methods. Any printing and/or keyboard input is done by main() in each exercise
- When you write source code, it should be readable and well-documented. You should have tested your program thoroughly before you turn it in. Each compiler error will result in a 10 point penalty.
- Please add your name and FSUID as comments on top of your programs.

3 Task

Write the following programs, each in a separate file. Filenames should be:

- Dice.java
- Primes.java

4 Program 1

Filename: Dice.java

1. Write a method called rollDice that returns the result from rolling two standard six-sided dice
 - The method will take no parameters, but it will return the resulting total of the dice roll as an integer.
 - Remember that the rolling of a die involves randomness, so you'll need to use the random number generation library. There are 2 ways to generate a random number as we discussed in class, you need to use each for one die (using both of random generation methods for rolling two dice, if you only use one of them, 20 points will be deducted).
 - Note that rolling two 6-sided dice is NOT equivalent to picking a random number from 2 through 12. This is because when you roll 2 dice, there is a much larger change of hitting some totals than others. You need to roll two dice and add them together.

2. To test this method, write a `main()` routine that does the following:

- Ask the user to enter how many times they want to roll the dice, and let them enter a value (for this writeup, I'll call it `N`)
- Call the `rollDice` method this many times (i.e. `N` times), and count how many times a total of 2 (known as "Snake Eyes") appears, as well as how many times a total of 7 appears.
- For each of these totals (2 and 7), print out how many times that total appeared, as well as what percentage this is of the total number of rolls. Percentages should be printed to 2 decimal places.

Sample Run 1

How many times would you like to roll the two dice? 10000

Snake eyes (double 1s) appeared
268 times
2.68 % of the time

A roll of 7 appeared
1675 times
16.75 % of the time

Sample Run 2

How many times would you like to roll the two dice? 500000

Snake eyes (double 1s) appeared
13706 times
2.74 % of the time

A roll of 7 appeared
83300 times
16.66 % of the time

Sample Run 3

How many times would you like to roll the two dice? 500000

Snake eyes (double 1s) appeared
13921 times
2.78 % of the time

A roll of 7 appeared
83789 times
16.76 % of the time

5 Program 2

Filename: `Primes.java`

1. Recall that an integer is a prime number if it is divisible only by 1 and itself. Write a method called `isPrime` that takes in one integer parameter `X` and determines whether or not `X` is prime. The method should return a boolean value:
 - return `true` if `X` is a prime number
 - return `false` if `X` is not a prime number.

2. To test this method, write a `main()` routine that asks the user to input a positive integer `N`. Using your `isPrime()` method, find and print all the prime numbers less than or equal to `N`, where the output has 9 numbers per line (you can use tab characters to separate numbers on a line).

Sample Run

```
Please input a positive number: 500
The prime numbers less than or equal to 500 are:
 2 3 5 7 11 13 17 19 23
29 31 37 41 43 47 53 59 61
67 71 73 79 83 89 97 101 103
107 109 113 127 131 137 139 149 151
157 163 167 173 179 181 191 193 197
199 211 223 227 229 233 239 241 251
257 263 269 271 277 281 283 293 307
311 313 317 331 337 347 349 353 359
367 373 379 383 389 397 401 409 419
421 431 433 439 443 449 457 461 463
467 479 487 491 499
```