

# Seneca College

Applied Arts & Technology  
SCHOOL OF COMPUTER STUDIES

**JAC444****Submission date:****Feb 01, 2023**

## Workshop 1

**Description:**

The first Assignment lets you practice basic java coding techniques, creating classes, methods and using arrays.

**Task 1:**

The explosive growth of Internet communications and data storage on Internet-connected computers has greatly increased privacy concerns. The field of cryptography is concerned with coding data to make it difficult (and hopefully—with the most advanced schemes—impossible) for unauthorized users to read.

In this exercise you'll investigate a simple scheme for encrypting and decrypting data. A company that wants to send data over the Internet has asked you to write a program that will encrypt it so that it may be transmitted more securely.

All the data is transmitted as four-digit integers. Your application should read a four-digit integer entered by the user and encrypt it as follows:

- Replace each digit with the result of adding 7 to the digit and getting the remainder after dividing the new value by 10.
- Then swap the first digit with the third and swap the second digit with the fourth.
- Then print the encrypted integer.

Write a separate application that inputs an encrypted four-digit integer and decrypts it (by reversing the encryption scheme) to form the original number.

[*Optional reading project:* Research “public key cryptography” in general and the PGP (Pretty Good Privacy) specific public key scheme.

You may also want to investigate the RSA scheme, which is widely used in industrial-strength applications.]

**Task 2:**

This task is required to create Craps, which is a popular dice game played in casinos. You are supposed to write a program to play a variation of the game, as follows:

- Roll two dice. (Each roll should produce two random numbers between 1 to 6)
- Each die has six faces representing values 1, 2, ..., and 6, respectively.
- Check the sum of the two dice. If the sum is 2, 3, or 12 (your program should display *craps*), you lose the game.

- If the sum of the two dice is 7 or 11 (your program should display *naturals*), you win the game.
- If the sum of two dice is any value (i.e., 4, 5, 6, 8, 9, or 10), your program should establish a point in the game (meaning store that sum). Continue to roll the dice until the sum is either a 7 or the same point value which was established. If rolled sum is 7, you lose the game. Otherwise, if the rolled sum is equal to established point you win.

Your program acts as a single player.

You rolled 5 + 6 = 11  
Congratulations, You win

You rolled 1 + 2 = 3  
Craps, Better Luck Next Time, You lose

You rolled 4 + 4 = 8  
Point is (established) set to 8  
You rolled 5 + 1 = 6  
You rolled 1 + 1 = 2  
You rolled 6 + 2 = 8  
Congratulations, You Win

You rolled 5 + 1 = 6  
Point is (established) set to 6  
You rolled 2 + 5 = 7  
Craps, Better Luck Next Time, You Lose

### Task 3: (Modification of Craps)

Modify the craps program from above to allow **wagering**.

- Initialize variable bankBalance to 1000 dollars.
- Prompt the player to enter a wager.
- Check that wager is less than or equal to bankBalance, and if it's not, have the user reenter wager until a valid wager is entered.
- Then, run one game of craps.
- If the player wins, increase bankBalance by wager and display the new bank-Balance.
- If the player loses, decrease bankBalance by wager, display the new bank-Balance, check whether bankBalance has become zero and, if so, display the message  
"Sorry. You busted!"
- As the game progresses, display various messages to create some "chatter," such as  
"Oh, you're going for broke, huh?" or  
"Aw c'mon, take a chance!" or  
"You're up big. Now's the time to cash in your chips!"
- Implement the "chatter" as a separate method that randomly chooses the string to display.

*Continue to the next page...*

## Workshop Header

/\*\*\*\*\*

**Workshop #**

**Course:**<subject type> - Semester

**Last Name:**<student last name>

**First Name:**<student first name>

**ID:**<student ID>

**Section:**<section name>

*This assignment represents my own work in accordance with Seneca Academic Policy.*

*Signature*

**Date:**<submission date>

\*\*\*\*\*/

## Code Submission Criteria:

Please note that you should have:

- Appropriate indentation.
- Proper file structure
- Follow java naming convention
- Document all the classes properly using JavaDoc
- JavaDoc should be generated properly in the project
- Do Not have any debug/ useless code and/ or files in the assignment

## Deliverables and Important Notes:

**All these deliverables are supposed to be uploaded on the blackboard once done.**

- You are supposed to create **video with voice** of your running solution. **(50%)**
  - Screen Video captured file should state your last name and id, like Ali\_123456.mp4 (or whatever the extension of the file is)
- OR**
- Show your work during the lab
- A text file which will reflect on learning of your concepts in this workshop. **(20%)**
  - Should state your Full name and Id on the top of the file and save the file with your last name and id, like Ali\_123456.txt
- JavaDocs must be used for proper documentation of each task. **(15%)**

- Submission of working code. (15%)
  - Make sure you follow the “**Code Submission Criteria**” mentioned above.
  - You should zip your whole working project to a file named after your Last Name followed by the first 3 digits of your student ID. For example, **Ali123.zip**.
- Your marks will be deducted according to what is missing from the above-mentioned submission details.
- Late submissions would result in additional 10% penalties for each day or part of it.
- Remember that you are encouraged to talk to each other, to the instructor, or to anyone else about any of the assignments, but the final solution may not be copied from any source.