

# Term Project

---

Contract due: Friday 10/16/2020 11:59pm

Project due: Sunday 11/15/2020 11:59pm

## Contents

<b>Project</b>	<b>2</b>
(a) Contract Agreement . . . . .	2
(b) Minimum Requirements . . . . .	2
(c) Example Project Ideas . . . . .	2
<b>Submission</b>	<b>3</b>
<b>Rubric</b>	<b>3</b>
<b>Contract</b>	<b>4</b>

## Project

You will develop a non-trivial program of your choice in Java Swing and present it by the end of this course.<sup>1</sup> The goal of this project is to demonstrate that you understand how to build a non-trivial OOP program with Java Swing GUI as a front end. A contract agreement between you and your instructor is required. The program needs to be of sufficient complexity to be approved. Ideally it will utilize many of the concepts and Java syntax that you have explored in this course.

If you are working in a team, please send your instructor an email with the names of all team members. If working by yourself, here is the URL to setup your GitHub Classroom repository:

<https://classroom.github.com/a/4Yij3Itu> <- - - - - Don't click if working in a team!

### (a) Contract Agreement

**Your instructor must sign off on your proposed project before you can begin working on it.** Please complete [the contract below](#) and send it to your instructor. Once all functionality is agreed upon, the instructor will sign and return it to you. **The sooner you do this, the more time you have to work on your project**, so don't delay! Once signed by both parties, commit and push this contract to your project repository. Note that various programs can open and edit PDF files, including: Microsoft Word, Google Drive, and many free online editors. If necessary, you can also screenshot, print, type / hand write, scan / capture the contract. Only the [contract](#) portion of this document is required.

### (b) Minimum Requirements

- Must have a Java Swing front-end GUI
- Must utilize some OOP paradigms: encapsulation, inheritance, polymorphism, abstraction
- Must contain *at least* 3 distinct classes
- Must utilize at least 1 array or ArrayList
- Must validate inputs from the user (input validation)
- Must save and/or read information from a file

### (c) Example Project Ideas

These are just examples of sufficiently complex programs. Please feel free to propose your own project.

- A persistent, editable, file-backed collection of some objects (e.g. students, clients, events, etc.)
- An encrypter and decrypter of selected files
- A program to manage a checklist of TODO items
- A very simple painting program w/ different colors and brushes
- A more complete ATM program w/ multiple accounts, login, and file-backed encrypted data
- A scientific calculator that logs history of computations to file
- A very simple video game
- A polished D&D RPG player creator / stat generator
- etc.

---

<sup>1</sup> All presentations will be held virtually online.

## Submission

You will commit and push your changes to your individual or team GitHub Classroom repository for this assignment. If you are in a group, all team members will commit and push to the same repository (and all team members are expected to be visible in the commit history). Please commit and push early and often to demonstrate your work ethic and progress. You will also commit and push your contract and any presentation slides / materials used for your presentation.

## Rubric

Task	Percentage
Using more than 10% of code from an existing Java project online	Instant 0
General attempt at completing your project	35%
Program compiles and runs w/out crashing	10%
Use of OOP paradigms	10%
Use of 3 or more distinct classes	10%
Use of at least 1 array or ArrayList	10%
Use of file input / output	10%
User input is validated, making it difficult to crash program w/ bad input	5%
Presentation professionalism and thoroughness	10%
Total	100%

