

CMPS 3390 Project 1

Spring 2026

BEFORE YOU BEGIN: LOGGING YOUR WORK

For each project you will be expected to log ALL of your work including research, setting up your environment, developing your app, debugging, etc. While there are some tools available to log your work, the simplest way to get started is to just create a table or spreadsheet. For the research portion of your log you should include AT LEAST three columns: Resource, Takeaway, and Time. Under **resource** you will provide a link to the site you visited, under **takeaway** you will have a short description of what value the resource provided, and under **time** you will simply log how long you spent reading/watching the resource.

Here's an example:

RESOURCE	TAKEAWAY	TIME
https://chatgpt.com/c/bded728a-7be2-44f4-8d35-dd4870a049ff	Steps for setting up a computer to develop an Android app	0:20
https://www.youtube.com/watch?v=XLt_moCoauw	Simple Kotlin App Tutorial	0:15
https://kotlinlang.org/docs/comparison-to-java.html	ARTICLE: Kotlin comparison to Java	0:30
https://www.youtube.com/playlist?list=PLQkwcJG4YTCQTEk4J	PLAYLIST: Git for Android Developers	1:00
https://developer.android.com/training/dependency-injection	Dependency injection in android	0:30

Try to read/watch/log as many resources as possible over the next week to develop a CLEAR understanding of the frameworks and available tools for your project. I know you will be tempted to jump to the development phase of your project, but it is important that you first understand the full toolchain, the pros & cons, the main hurdles & challenges, etc. Once you do, you will be able to develop your apps quickly and efficiently.

MEASURE TWICE, CUT ONCE!

Part 1 - Research

For Project 1 you will be building AT LEAST one GUI based app using the framework of your choice. Due to the fast pace at which the "application ecosystem" changes and evolves, it is important that you do your research first. Before you dive in and start coding: decide what operating system, platform, and framework will allow you to complete your project without expending all of your resources. Out in the real world there are only two resources available to you (**TIME & MONEY**) and it would be management's job to determine how to balance the two. However, in your case, the only resource you have available for this project is **TIME**.

With this idea of **TIME** as a resource, there are a few questions you should consider:

- What operating systems do I already have available to me? Can I use a VM?
- Are there any frameworks that use languages I am already familiar with?
- Is cross compatibility really that important in the early stages, or can I focus on one now, and expand later?
- How many FREE and TRUSTWORTHY resources are available for this platform/framework?
- Do I know anyone that has real-world experience using this platform/framework?
- How many IDE's are available for this language/framework?
- What are the pros & cons of each IDE? For example: integrated version control, GUI tools, extensions, etc

Part 2 - Setup & Prototype (Hello World!)

Now that you have thoroughly researched your framework & toolchain, it's time to get ready to start making applications! This step will be different for everyone, but the main goal here is to get your environment set up so you can begin developing & testing. Notice that I still have not given you any detailed specs about the app that you will be building, as your goal for part two is simply to configure your system and become familiar with your tools.

If you want you can get a VERY SIMPLE single view GUI running with a "Hello World" label.

There are a few important things you should consider when completing this part

Most of these things you should probably already be familiar with from doing your research:

- **Are you targeting a specific operating system/environment? Will you need a VM?** If so, that will probably be your first step. Otherwise you can probably install your ide, runtime environments, etc on your own machine.

- **Does your IDE provide its own runtime environment or will you have to install it separately?**

For example:

- If you are going to use reactJS in VSCode, you will need to install node/npm on your machine.
 - If you are going to develop a Java Swing app in IntelliJ you do not HAVE to install the JDK on your machine separately, however if you want to run your app outside of IntelliJ you will at least need to make sure you bundle the JRE with your application.
 - If you are developing an android app you should be able to use the VM tools built into Android Studio, however some people choose to set up a separate Android VM for testing, or tether their phone with a USB cable.
- **Does your IDE support git/version control out of the box, or will you need to initialize a repo yourself in the project directory?** Either way you will probably need to install git on your machine. (FOR EXAMPLE, Godot's built-in GIT tools are notoriously bad, so most people just do "git init" from their project folder and manage it themselves outside of Godot.)
 - **Are there any special configuration steps you must do on your machine, like configuring special permissions or setting certain environment variables?** These days most tools (like git) will add themselves to the system path automatically, but if not you may need to do it yourself.
 - **What are the different project types available within your IDE and which one are you targeting?**

DOCUMENT EVERYTHING!

Just like with **Part 1**, it is important that you document EVERY STEP of your setup process so you can repeat it again later, or so you can show a teammate how to set up their system in the future.

All of the following should be documented:

- List of VMs, applications, runtimes, scripts, etc **with download links**
- An ordered list of install/config steps (for example, you must install node before you can use NPM)
- Any potential blockers or challenges you came across and how to fix them

You can create a separate spreadsheet for this, or you can add a new tab to your existing workbook.

But it should be separate from your research log.

Part 3 - Develop Your Apps

Ok, you've done the research, you've set up your environment, and now it's time to put your knowledge to work! It is important when you are learning a new programming language or framework that you start small. So we will be building a few simple frontend-only apps with a couple simple views with a variety of UI elements.

App 1

For your first app, keep it simple. Set up a new project called "Hello" and complete the following:

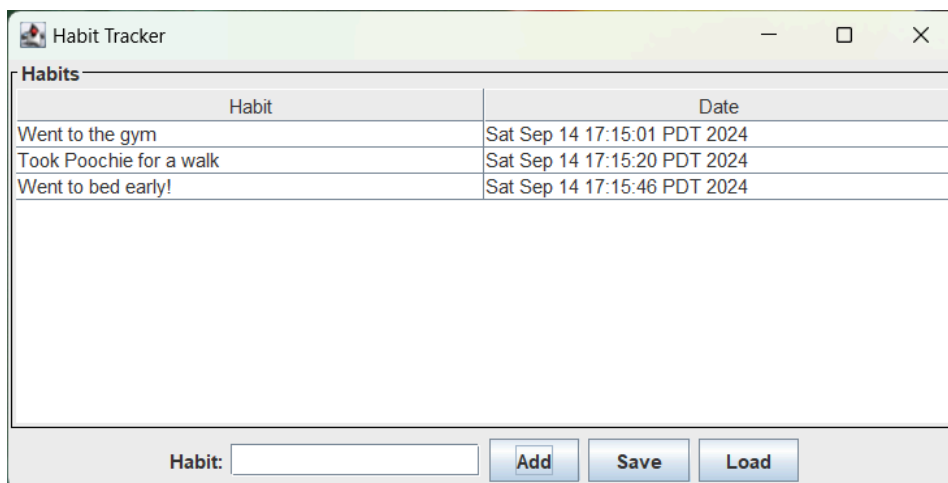
1. Create your first view.
2. The view will contain one button with the text "Hello".
3. When the "Hello" button is clicked it should launch a new view/model/alert.
4. The new view/model will have a single text label that says **"Hello World"** and an **OK** button.
5. When the OK button is clicked the application should go back to the first view.

App 2

Great, now that you're becoming more familiar with your environment, choose one of the following options to build:

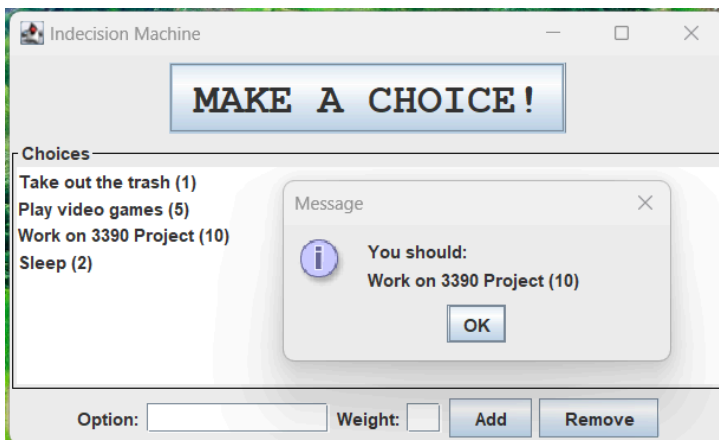
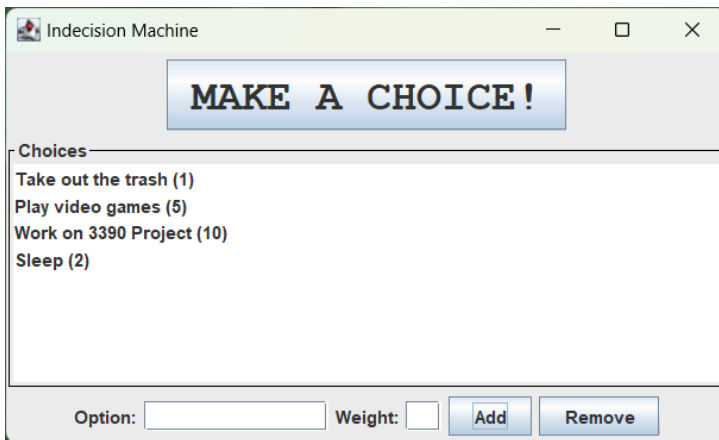
Option 1 - Habit Tracker

1. Create a view/panel with the follow:
 - a. A Table/Datagrid with a label "Habit List" and two columns: "Habit" and "Date".
 - b. A label with the text "Habit:"
 - c. A text input field
 - d. A button with the text "Add"
2. When the user enters a habit in the input field and hits the add button it should trigger a listener to:
 - a. Detect the current date/time
 - b. Add the habit along with the date/time value to the Table/Datagrid
 - c. Clear the text input field
3. **Stretch Goal:** Once you have complete steps 1 and 2, see if you can find a way to have the data persist. This will require you to maintain the data in a manner that will allow you to serialize it. Once it is serialized, you should be able to write it to a local file or save it in local storage to be retrieved the next time you open the app. How the data is saved/loaded is up to you. You can have "Save" and "Load" buttons, or EVEN BETTER: you could have the data automatically saved whenever the list is modified, and loaded when the program is first launched.



Option 2 - Indecision Machine

1. Create a view/panel with the following:
 - a. A scrollable/selectable text list with a label "Choices"
 - b. A label with the text "Option:"
 - c. A text input field
 - d. A button with the text "Add"
 - e. A button with the text "Remove"
 - f. A button with the text "Decide"
2. When the user clicks the "Add" button it should trigger a listener to:
 - a. Check if the input field has a value, then add the option to the "Choices" list.
 - b. Clear the value from the input field
3. When the user clicks the "Remove" button it should trigger a listener to:
 - a. Check if an item from the choices list is selected
 - b. Remove the Item from the list
4. When the user clicks the "Decide" button it should trigger a listener to:
 - a. Randomly select an item from the "Choices" list
 - b. Generate a new view/model/alert that displays the choice to the user
5. **Stretch Goal:** Once you have steps 1-4 completed, see if you can implement a "weight" feature will increase the odds of a choice being selected based on how much weight the user assigns to it. There are various ways to do this, but it will most likely require that you have a data model/class that is separate from your view.



Part 4 - Project Submission

To receive credit for this project you will need to submit your logs and source code to me, then present your working application in person. Your final Project 1 grade will be based on your logs/documentation, your source code, and your ability to present a working app.

Submitting Your Work

To receive full credit for your submission you will be required to send me three things:

1. Your research log
2. Your setup log/documentation
3. The source code for your application

For item 3 (source code) I am only requiring you to submit the code that you wrote for your app. Please **do not** include any workspace configs, output folders/executables, or library folders (like node_modules). In most IDEs or frameworks you will usually have a folder that is marked at "source" or "src" or something similar.

To submit your work you have a few different options:

- Place your files in a folder, zip the folder, and email it to me
- Place your files in a cloud folder (like Google Drive or OneDrive) and send me an access link
- Create a git repo for your files, push to github, and invite the user **wmpaulroyer** as a collaborator

Presenting Your App

To present your applications, you will need to sign up for a specific time to meet with me.

Since our time is limited and I would like to meet with everyone I will be extending my office hours to accommodate.

We will meet in my office, but if you need to use a lab computer we can walk down to one of the lab computers.

This should be a quick presentation of your applications, so make sure you have it up and running before we meet.

You can sign up here:

[CMPS 3390 - PROJECT 1 PRESENTATION SIGNUPS](#)