Module #3 Plan | CSE 310 – Applied Programming

Name	Date	Teacher
Ethan Trent	10/20/25	Bro McGary

Software Description

I plan to create an interactive chatbot application in Java that demonstrates core programming concepts while providing an engaging user experience. The chatbot will be capable of:

- **Conversational Interaction**: Respond to user questions using a knowledge base stored in a HashMap
- **Multiple Features**: Tell jokes, play word games, provide advice, and answer FAQs
- **Learning Capability**: Store new questions and answers that users teach it, persisting them to a file
- **Chat History**: Maintain a conversation history using an ArrayList for review
- **Extensible Design**: Use inheritance and interfaces to allow different bot personalities (JokeBot, GameBot, AdviceBot)

The project will demonstrate all required Java concepts including variables, conditionals, loops, functions, classes, and the Java Collection Framework (HashMap, ArrayList, TreeSet). Stretch challenges include file I/O for persistence and object-oriented design patterns using inheritance and interfaces.

Module

Mark an **X** next to the module you are planning

Module	Language	
Cloud Databases	Java	X
Data Analysis	Kotlin	
Game Framework	R	
GIS Mapping	Erlang	
Mobile App	JavaScript	
Networking	C#	
Web Apps	TypeScript	
Language – C++	Rust	
SQL Relational Databases	Choose Your Own Adventure	

Create a Schedule

Create a detailed schedule using the table below to complete your selected module during this Sprint. Include details such as what (task), when (time), where (location), and duration. You should also include time to work on your team project. You are expected to spend 16 hours every Sprint working on your individual module, team project, and other activities. Time spent on this individual module should be at least 10 hours.

	First Week	Second Week
Monday	**Week 1**: 2 hrs - Project setup, Java environment review, basic class structure (Chatbot, Message classes)	**Week 2**: 1.5 hrs - Implement file I/O for saving/loading learned responses, test persistence
Tuesday	**Week 1**: 2 hrs - Implement HashMap for Q&A storage, ArrayList for chat history, basic conversation loop	**Week 2**: 1.5 hrs - Create Respondable interface, implement JokeBot and GameBot modules
Wednesday	**Week 1**: 1.5 hrs - Add conditional logic for different response types, implement joke telling	**Week 2**: 2 hrs - Implement inheritance with base Bot class, create AdviceBot extension, code review
Thursday	**Week 1**: 1.5 hrs - Implement word game feature, add TreeSet for topic tracking	**Week 2**: 1.5 hrs - Write chat logs to file, polish user interface, error handling
Friday	**Week 1**: 1 hr - Team project work	**Week 2**: 1.5 hrs - Final testing, README documentation, prepare submission
Saturday	**Week 1**: 1.5 hrs - Mid-week testing, refactor code, add user commands (exit, help, history)	**Week 2**: 1 hr - Team project work

Identify Risks

Identify at least two risks that you feel will make it difficult to succeed in this module. Identify an action plan to overcome each of these risks.

	Risk	Action Plan
1	**Complexity Creep**: Adding too many features (inheritance, interfaces, file I/O) might cause scope to grow beyond the 2-week timeframe	Start with the core required features (variables, conditionals, loops, functions, classes, basic Collections). Implement these solidly first, then add stretch challenges incrementally. If time runs short, focus on demonstrating the concepts clearly rather than building a fully-featured application.
2	**File I/O Learning Curve**: Reading/writing files in Java can be tricky, especially handling exceptions and file paths	Allocate extra time for file I/O implementation early in Week 2. Use Java's built-in Scanner and PrintWriter classes which are simpler than BufferedReader/BufferedWriter. Test file operations thoroughly and handle FileNotFoundException and IOException properly. Start with simple text files before attempting complex data structures.