



KN-Lang Chronicles — Chapter 1

Introduction to KN-Lang:

Welcome to **KN-Lang**, the un-official (and proudly whimsical) programming language of **KNNX Corp** — conceived during one late-night brainstorming session, powered by caffeine and questionable decisions.

KN-Lang is designed for those who love modular logic *and* sarcasm in equal measure. It's simple, object-oriented, class/function-based, and slightly unpredictable—just like your first real-world project.

KN-Lang Core Concepts

Concept	KN-Lang Term	Meaning
Variable	Quark	The smallest building block that holds your data (and occasionally your sanity).
Function	Doodle	A block of logic that does stuff.
Class	Squad	A group of related Doodles and Quarks that stick together.
Loop	SpinCycle	Because you're just tumbling through data anyway.
Comment	// Just Saying:	KN-Lang likes to give its opinion too.
Print Output	Print("...")	Console output command (with a sense of humor).
Input	Input("...")	Reads user input.
Conditionals	If / Else	Standard logic, with dramatic flair.

Why KN-Lang Exists

At KNNX Corp, we don't just test how well you code — we test *how creatively* you think, structure, and adapt. KN-Lang is your playground to demonstrate clean logic, modular thinking, and a bit of fun.

This assignment helps us see:

- Your ability to design modular systems with clear Squad/Doodle separation.
- Your comfort handling input, flow, and data.
- Your creativity and humor while solving problems.

Assignment 1 — *Text Adventure Game in KN-Lang*

Overview

Create a **console-based text adventure** where a player explores a strange, witty world. They'll move through rooms, pick up items, solve puzzles, and (hopefully) escape.

Requirements

1. Game Structure

- Multiple **rooms**, each with unique descriptions.
- Some rooms contain **items, puzzles, or characters**.
- The player explores, interacts, and collects items to progress.

2. Squads (Classes)

- **Player Squad** → tracks player name, inventory, and current room.
- **Room Squad** → holds room description, items, and exit directions.
- **GameManager Squad** → runs the main game loop and handles input.

3. Doodles (Functions)

- **Player Doodles:** pickUp(item), move(direction), showInventory()
- **Room Doodles:** showDetails(), interact()
- **GameManager Doodles:** startGame(), handleCommand()

4. Core Gameplay

- Commands: go <direction>, pick <item>, inventory, look, quit.
- Puzzle rooms: at least 2 requiring item use or logic to solve.
- Invalid inputs should trigger sarcastic responses.

5. Humor

Inject personality! For instance,

“The door stares at you blankly. Perhaps it’s missing something — like a key?”

Sample KN-Lang Snippets

```
// Just Saying: This is the main Player Squad
Squad Player {
    Quark name = "Unnamed Hero";
    Quark inventory = [];
    Quark currentRoom = "Entrance";
    Quark isAlive = True;

    Doodle pickUp(item) {
        inventory.add(item);
        Print("You picked up the " + item + ". Confidence level +10.");
    }
}
```

```
// Just Saying: Room Squad defines room logic
Squad Room {
    Quark name;
```

```
Quark description;
Quark items = [];
Quark exits = {};
```

```
Doodle showDetails() {
    Print(description);
    If (items.length > 0) {
        Print("Items here: " + items.join(", "));
    } Else {
        Print("It's empty... suspiciously empty.");
    }
}

// Just Saying: GameManager runs everything
Squad GameManager {
    Quark rooms = {};
    Quark player = new Player();

    Doodle startGame() {
        Print("Welcome to the Mysterious Land of KN-Lang!");
        player.currentRoom = rooms["Entrance"];

        SpinCycle (player.isAlive) {
            Quark command = Input(">> ");
            handleCommand(command);
        }
    }
}
```

Example Room Setup

```
// Just Saying: The Spooky Dungeon
Quark spookyDungeon = new Room();
spookyDungeon.name = "Spooky Dungeon";
spookyDungeon.description = "Damp walls, moldy smell, and a distant snore...";
spookyDungeon.items.add("rusty key");
spookyDungeon.exits = { "north": "Main Hall", "west": "Locked Door" };

GameManager.rooms["Spooky Dungeon"] = spookyDungeon;
```

Assignment 2 — Quiz Master in KN-Lang

Overview

Design a **console-based quiz game** called *Quiz Master*.

The player meets a snarky, self-aware console that tests their knowledge with different categories and difficulty levels.

Requirements

1. Game Flow

- Player chooses **category** (Science, History, Fun Facts, etc.)
- Chooses **difficulty** (Easy / Medium / Hard).
- Answers a series of questions; score updates dynamically.

2. Squads (Classes)

- **Player Squad** → name and current score.
- **Question Squad** → question text, category, difficulty, correct answer.
- **QuizMaster Squad** → manages questions, scoring, and witty responses.

3. Doodles (Functions)

- chooseCategory(), askQuestion(), evaluateAnswer(), showScore()
- Bonus Doodle: commentOnAnswer(isCorrect) for humor injection.

4. Scoring System

Difficulty	Points (Correct)	Penalty (Wrong)
Easy	+5	-2
Medium	+10	-5
Hard	+15	-7

5. Humor Rules

- **Right answer:** “Well, someone paid attention in school!”
- **Wrong answer:** “Close... if we were grading on imagination.”

6. End of Game

After 10 questions: show total score and rank.

- Score $\geq 80 \rightarrow$ “Quiz Royalty has arrived!”
- 50 – 79 \rightarrow “Quiz Master in training.”
- Below 50 \rightarrow “Better luck next time, genius.”

Sample KN-Lang Snippet

```
// Just Saying: Setting up some sample questions
Quark q1 = new Question();
q1.category = "Science";
q1.difficulty = "Easy";
q1.questionText = "What is the chemical symbol for water?";
q1.correctAnswer = "H2O";

Quark q2 = new Question();
q2.category = "History";
q2.difficulty = "Medium";
q2.questionText = "Who was the first President of the USA?";
```

```
q2.correctAnswer = "George Washington";  
  
// Just Saying: Add questions to the QuizMaster  
Quark quiz = new QuizMaster();  
quiz.questions.add(q1);  
quiz.questions.add(q2);  
quiz.start();
```

🏁 Chapter End — KN-Lang Chronicles: Level 1 Complete

Congratulations! You've just taken your first steps in KN-Lang — a language with equal parts logic and laughter.

Before submitting, please reflect briefly:

1. What part of this assignment challenged you the most, and how did you approach it?
2. If you could extend this project, what feature would you add next in KN-Lang?

Remember: we care less about *perfect syntax* and more about your *problem-solving style*. Be modular, be creative, and—above all—have fun.

👉 **Bonus Tip:** If you started talking back to the console, welcome to KNNX Corp. You fit right in.