



CLOVE User Manual

Version 1.0 | Last Updated: December 04, 2024

Code, Logic, Overcome, Validate, Excel

An Adaptive Learning Platform for Java Programming

Table of Contents

- Welcome to CLOVE
 - What is CLOVE?
 - Key Features
 - Game-Inspired Learning
 - Adaptive Intelligence
 - Progress Tracking
 - Multiple Challenge Types
 - Assessment System
 - Achievement System
- Getting Started
 - Creating Your Account
 - Step 1: Visit the Platform

- [Step 2: Sign Up](#)
- [Step 3: Email Verification](#)
- [Step 4: Log In](#)
- [Onboarding Experience](#)
 - [Step 1: Welcome Screen](#)
 - [Step 2: Character Creation](#)
 - [Step 3: Realm Selection](#)
 - [Step 4: Meet Your Mentor](#)
- [First Steps After Onboarding](#)
 - Recommended First Steps
- [Learning Components](#)
 - [Pre-Assessment](#)
 - [Step-by-Step: Taking a Pre-Assessment](#)
 - [Subtopics](#)
 - [Lessons](#)
 - [Step-by-Step: Completing a Lesson](#)
 - [Practice Exercises](#)
 - [Step-by-Step: Completing Practice Exercises](#)
 - [Challenges](#)
 - [Post-Assessment](#)
 - [Step-by-Step: Taking a Post-Assessment](#)
 - [Retention Tests](#)
 - [Step-by-Step: Taking Retention Tests](#)
- [Your Learning Journey](#)
 - [Phase 1: Getting Started](#)
 - [Phase 2: Learning Subtopics](#)
 - [Phase 3: Final Assessment](#)

- [Phase 4: Retention Testing](#)
 - [Tips for Success](#)
 - [What You'll Achieve](#)
- [Core Features](#)
 - [My Deck](#)
 - [Dashboard](#)
 - [Progress Page](#)
 - [Overall Progress](#)
 - [Performance Metrics](#)
 - [Topic Breakdown](#)
 - [Learning Analytics](#)
 - [Challenges Page](#)
- [Assessments & Retention Tests](#)
 - [Pre-Assessments](#)
 - [Post-Assessments](#)
 - [Retention Tests](#)
- [Challenges](#)
 - [Challenge Types](#)
 - [Code Fixer](#)
 - [Code Completion](#)
 - [Output Tracing](#)
 - [Challenge Features](#)
 - [Adaptive Difficulty](#)
 - [Hints](#)
 - [Timer](#)
 - [Points & Scoring](#)
- [Progress Tracking](#)

- Progress Dashboard
 - Overall Completion
 - Topic Progress
 - Subtopic Breakdown
 - Challenge Statistics
 - Time Spent
 - Mastery Levels
 - Progress Indicators
- Learning Analytics
- Profile & Settings
 - Profile Page
 - Profile Information
 - Achievements
 - Account Settings
 - Personal Information
 - Password Management
 - Profile Photo
 - User Account Management
 - Statistics
- Troubleshooting
 - Common Issues
 - Issue 1: Can't Log In
 - Issue 2: Pre-Assessment Not Unlocking Subtopics
 - Issue 3: Challenges Not Loading
 - Issue 4: Retention Test Not Available
 - Issue 5: Progress Not Updating

- [Getting Help](#)
- [For Developers](#)
 - [System Architecture](#)
 - [Frontend Architecture](#)
 - [Backend Architecture](#)
 - [Deployment Architecture](#)
 - [Core Algorithms](#)
 - [Database Backup](#)
 - [Development Setup](#)
 - [API Documentation](#)
 - [Key Files Reference](#)
- [Additional Resources](#)
 - [Learning Tips](#)
 - [Best Practices](#)
- [Glossary](#)
- [Version History](#)
- [Contact & Support](#)

Happy Learning!

Welcome to CLOVE

What is CLOVE?

CLOVE (Code, Logic, Overcome, Validate, Excel) is an intelligent, adaptive learning platform designed to teach Java programming through immersive, game-inspired challenges. The system personalizes your learning experience by adapting to your performance and learning patterns.

 Key Concept: CLOVE uses AI algorithms to adapt to your learning style, making each experience unique and personalized.

Key Features

Game-Inspired Learning

Three immersive themes:

- **Wizard Academy:** Master the ancient art of Code Magic
- **Detective Agency:** Solve mysteries and catch code criminals
- **Space Station Alpha:** Command systems of a futuristic space station

Adaptive Intelligence

- AI-powered system that adjusts difficulty based on your performance
- Personalized challenge selection
- Smart hint system that activates when needed

Progress Tracking

- Detailed analytics and visualizations
- Real-time progress monitoring
- Performance metrics and insights

Multiple Challenge Types

- **Code Fixer:** Identify and correct bugs
- **Code Completion:** Complete missing code segments

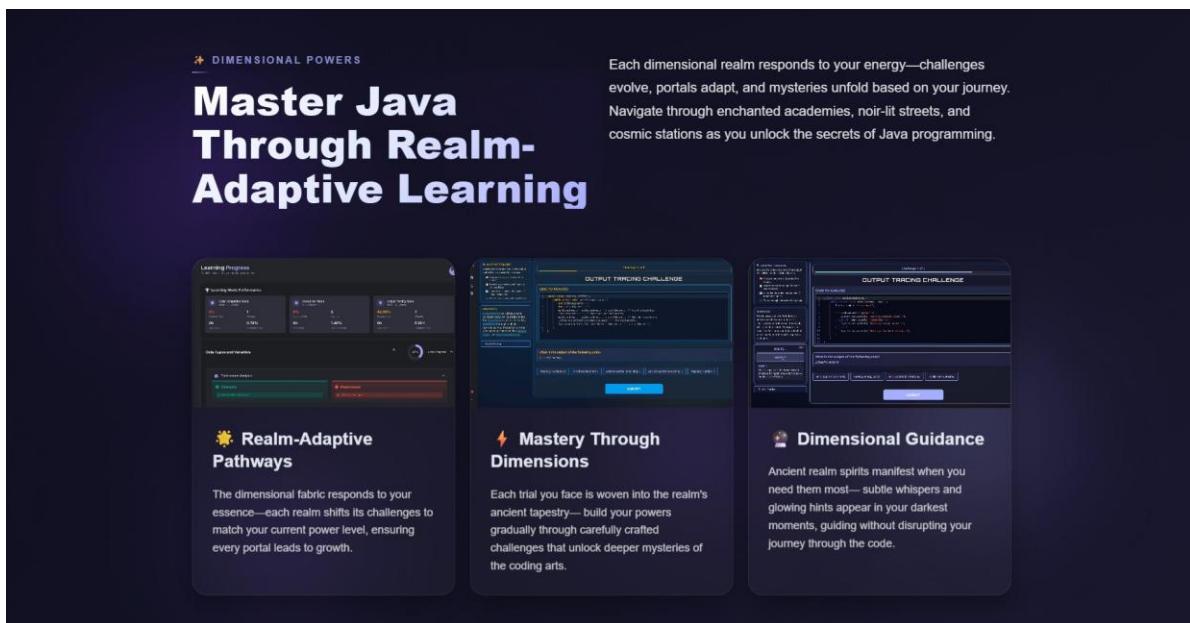
- **Output Tracing:** Predict program output

Achievement System

- Pre/post assessments to measure learning
- Retention tests to verify long-term knowledge
- Detailed performance breakdowns

Achievement System

- Track your progress and unlock achievements
- Earn points and badges
- Build your learning streak



Key Features

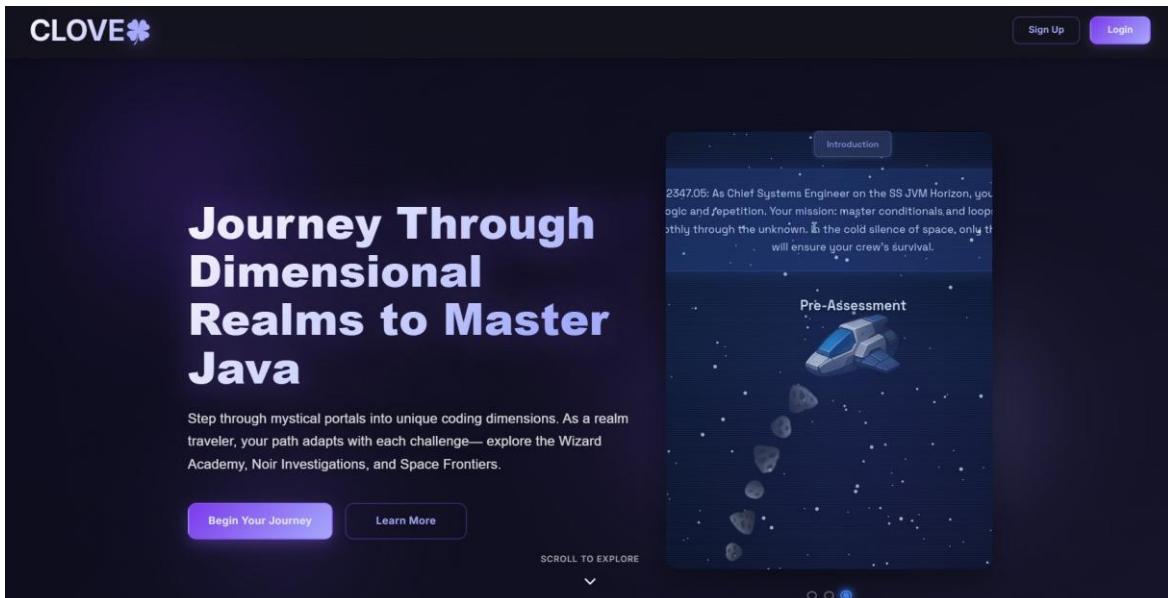
Getting Started

1. Creating Your Account

Follow these steps to create your CLOVE account:

Step 1: Visit the Platform

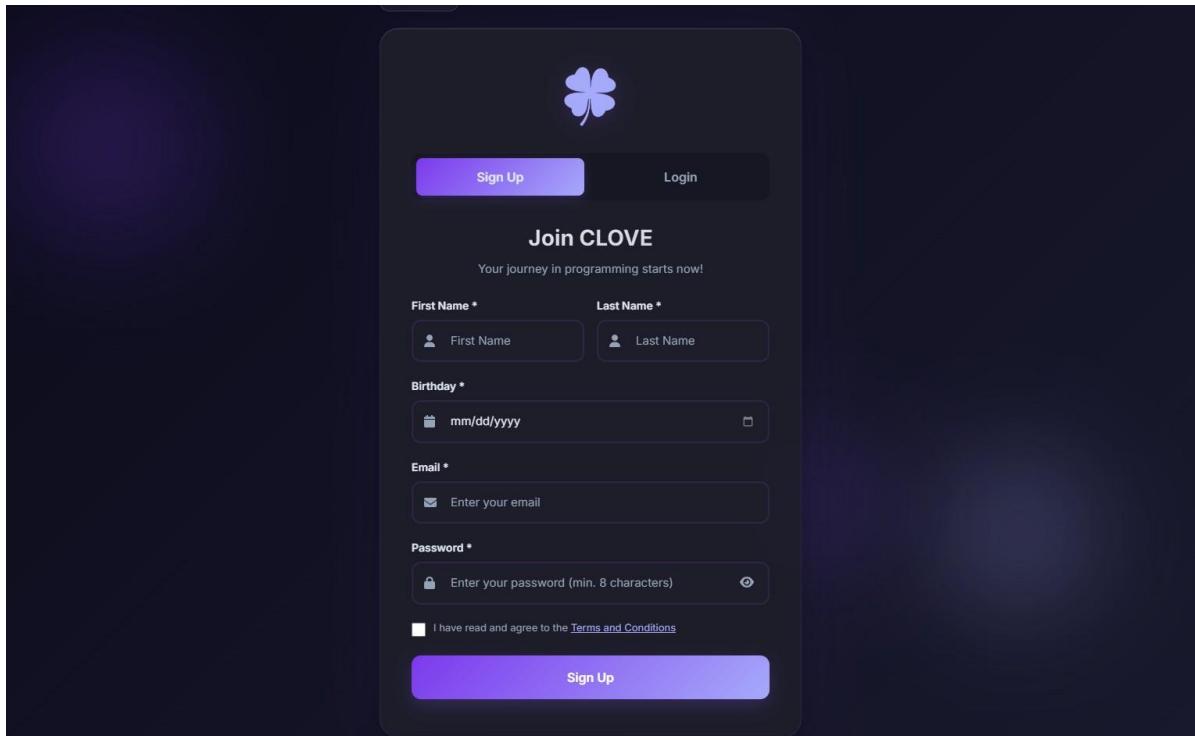
1. Open your web browser
2. Navigate to the CLOVE website (<https://clove-frontend.netlify.app/>)
3. You'll see the landing page with sign-up options



The Sign Up button at the top right

Step 2: Sign Up

1. Click the "Sign Up" button
2. Fill in the registration form:
 - **Email Address:** Use a valid email you can access
 - **Password:** Create a strong password (minimum 8 characters)
 - **First Name:** Your first name
 - **Last Name:** Your last name
 - **Birthday:** Your date of birth

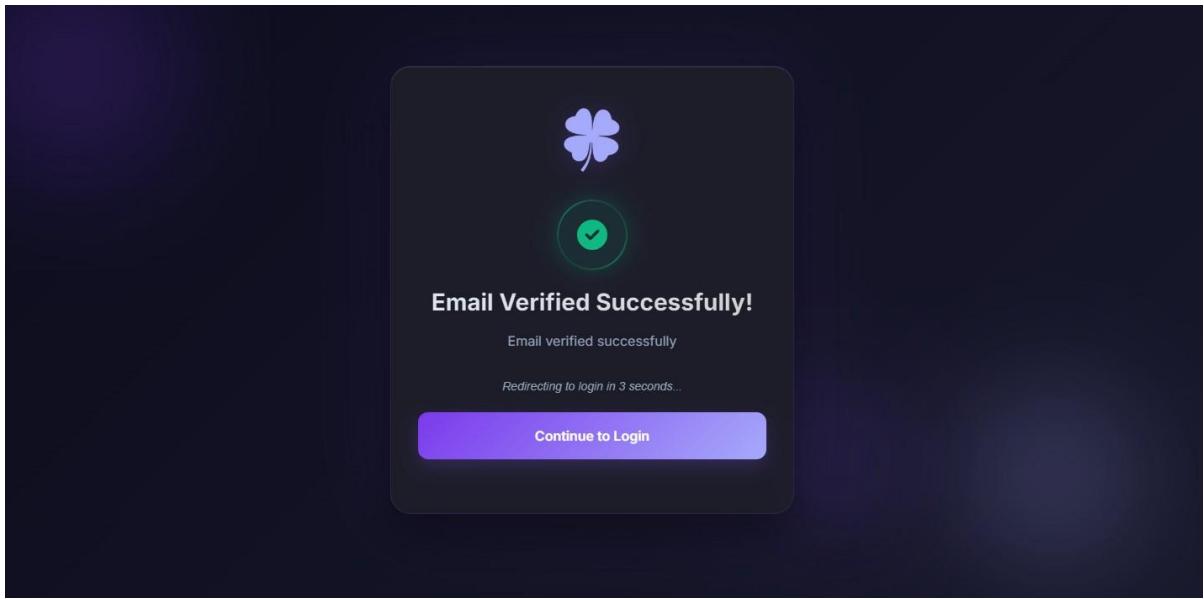


Sign Up Page

⚠ Important: Keep your password secure. Use a combination of letters, numbers, and special characters.

Step 3: Email Verification

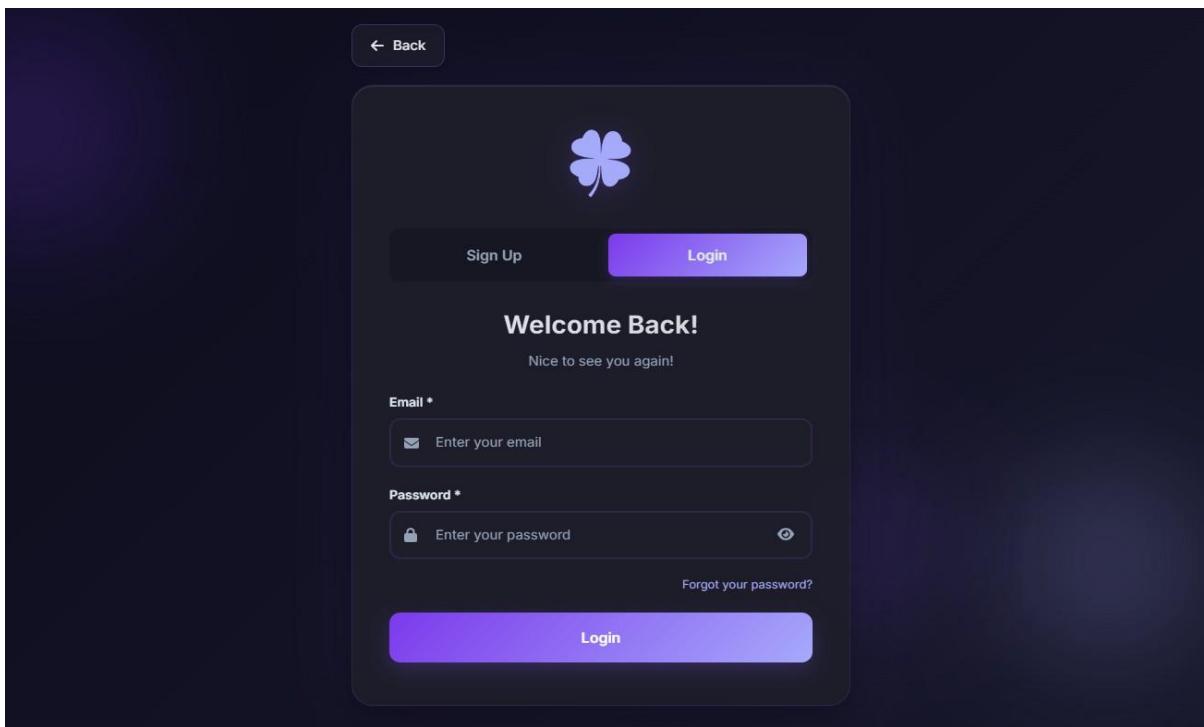
1. Check your email inbox
2. Look for the verification email from CLOVE
3. Click the verification link in the email
4. You'll be redirected back to the platform



User Successfully Verified

Step 4: Log In

1. Return to the CLOVE website
2. Enter your email and password
3. Click "Log In"
4. You'll be taken to the onboarding experience



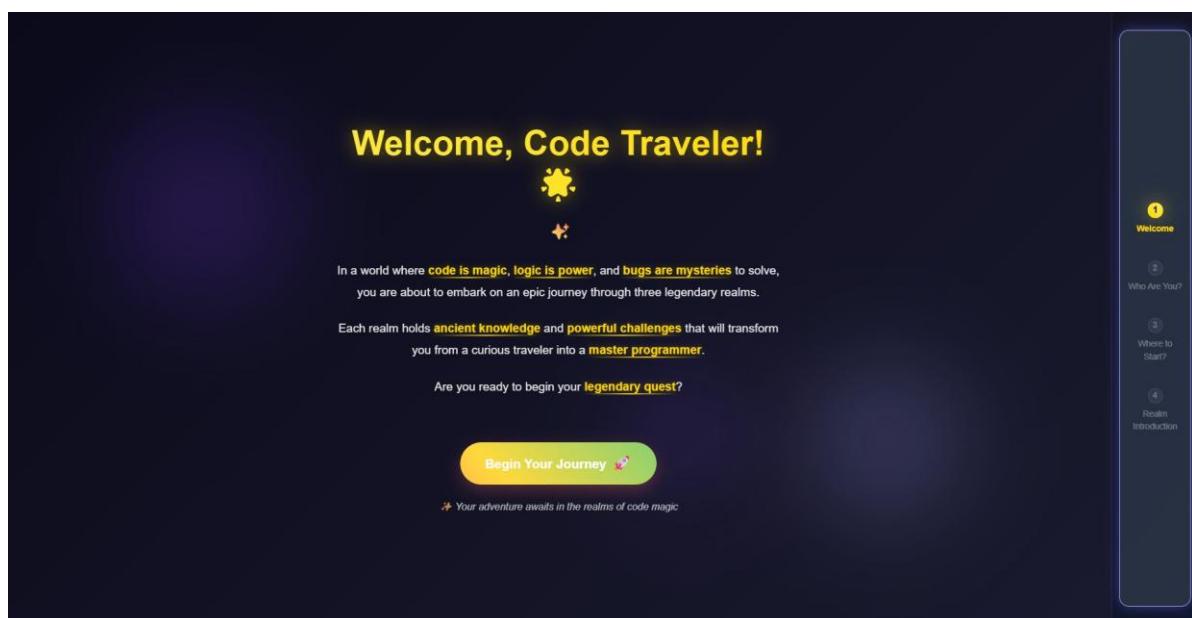
Log In Page

2. Onboarding Experience

When you first log in, you'll embark on a **Traveler's Journey** - an immersive onboarding experience that introduces you to the CLOVE world.

Step 1: Welcome Screen

- **Purpose:** Introduction to your coding adventure
- **What to Do:**
 - Read the story introduction
 - Learn about the journey ahead
 - Click “**Start Your Journey**” to continue



Onboarding Welcome Screen

Step 2: Character Creation

Choose your **Traveler Class** - this represents your learning style:

Who Are You?

Select the type that best describes your personality and approach to coding

What shall we call you, traveler?
(This will become your username)

ExpertCoder007

Welcome, ExpertCoder007 ✨

X What Best Describes You?

Select the personality type that resonates most with your approach to coding

Syntax Sage

Focus on learning code structure, grammar, and writing clean programs

Key Strengths:

- Understanding Syntax
- Structure & Organization
- Pattern Recognition

Logic Investigator

Specialize in problem-solving, debugging errors, and analytical thinking

Key Strengths:

- Breaking Down Problems
- Debugging & Testing
- Critical Analysis

Algorithm Explorer

Master efficient solutions, optimize performance, and learn computational thinking

Key Strengths:

- Algorithm Design
- Code Optimization
- Efficiency Techniques

← Back Continue Journey →

1 Who Are You? Welcome

2 Where to Start? Who Are You?

3 Realm Introduction Where to Start?

4 Realm Introduction Who Are You?

Class	Description	Best For
Syntax Mage	Masters of code structure and form	Detail-oriented learners
Logic Detective	Solvers of complex problems	Analytical thinkers
Algorithm Explorer	Adventurers in algorithms	Creative problem solvers

Who Are You?

Select the type that best describes your personality and approach to coding

What shall we call you, traveler?
(This will become your username)

ExpertCoder007

Welcome, ExpertCoder007 ✨

X What Best Describes You?

Select the personality type that resonates most with your approach to coding

Syntax Sage

Focus on learning code structure, grammar, and writing clean programs

Key Strengths:

- Understanding Syntax
- Structure & Organization
- Pattern Recognition

Logic Investigator

Specialize in problem-solving, debugging errors, and analytical thinking

Key Strengths:

- Breaking Down Problems
- Debugging & Testing
- Critical Analysis

Algorithm Explorer

Master efficient solutions, optimize performance, and learn computational thinking

Key Strengths:

- Algorithm Design
- Code Optimization
- Efficiency Techniques

← Back Continue Journey →

1 Who Are You? Welcome

2 Where to Start? Who Are You?

3 Realm Introduction Where to Start?

4 Realm Introduction Who Are You?

Choose the type that describes your personality

Step 3: Realm Selection

Select your first **Mystical Realm** where you'll begin your learning:

The screenshot shows a dark-themed user interface titled "Where Would You Like to Start?". At the top, a message says "Pick your starting adventure - you can explore all realms later!". Below it, a welcome message reads "Welcome, ExpertCoder007 the Algorithm Explorer! 🚀". On the right, a vertical sidebar lists steps: "Welcome", "Who Are You?", "Where to Start?", and "Realm Introduction". The main area displays three realm cards:

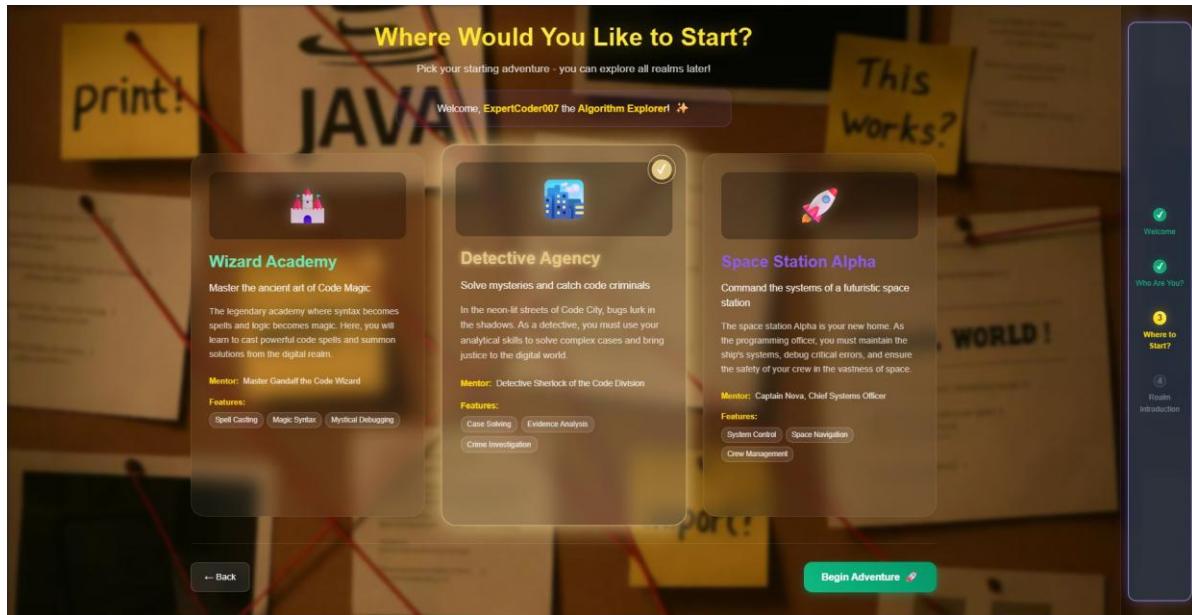
- Wizard Academy**: Fantasy theme. Description: Master the ancient art of Code Magic in a mystical academy. Mentor: Master Gandalf the Code Wizard. Features: Spell Casting, Magic Syntax, Mystical Debugging.
- Detective Agency**: Mystery theme. Description: Solve mysteries and catch code criminals in the neon-lit streets of Code City. Mentor: Detective Sherlock of the Code Division. Features: Case Solving, Evidence Analysis, Crime Investigation.
- Space Station Alpha**: Sci-Fi theme. Description: Command the systems of a futuristic space station. Mentor: Captain Nova, Chief Systems Officer. Features: System Control, Space Navigation, Crew Management.

At the bottom are "Back" and "Begin Adventure" buttons.

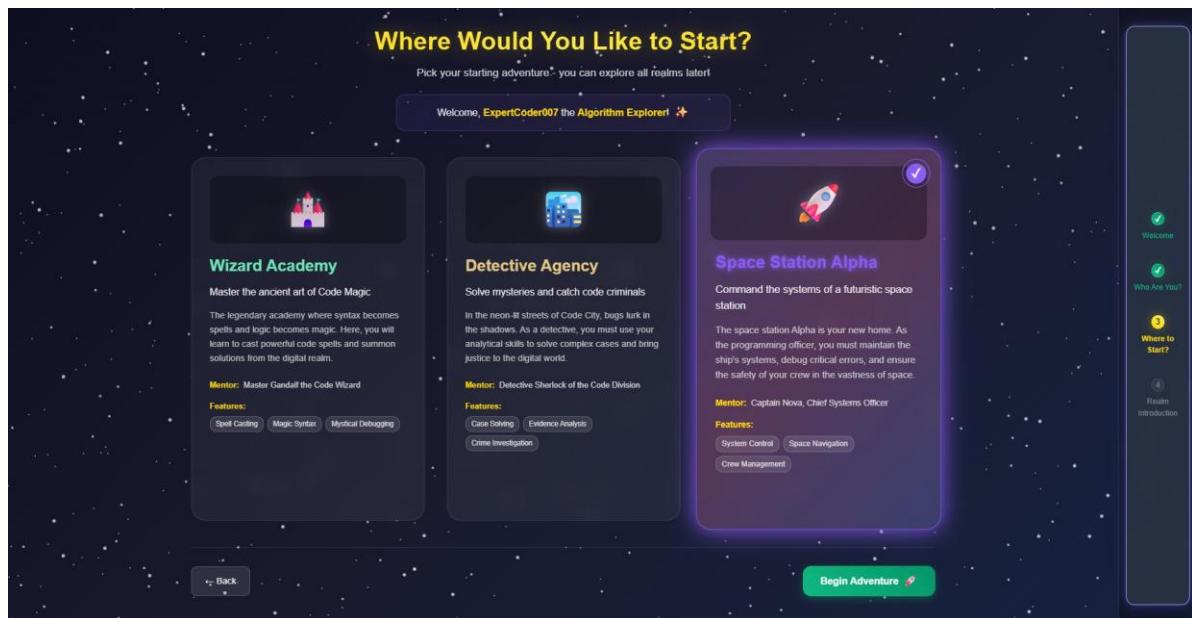
Realm	Theme	Description
🧙 Wizard Academy	Fantasy	Master the ancient art of Code Magic in a mystical academy
🕵 Detective Agency	Mystery	Solve mysteries and catch code criminals in a noir setting
🚀 Space Station Alpha	Sci-Fi	Command the systems of a futuristic space station

This screenshot is identical to the one above, but with a glowing green aura surrounding the "Wizard Academy" card, indicating it is the selected realm. The rest of the interface and the sidebar remain the same.

Wizard Theme



Detective Theme



Space Theme

Step 4: Meet Your Mentor

Each realm has a unique mentor who will guide you:

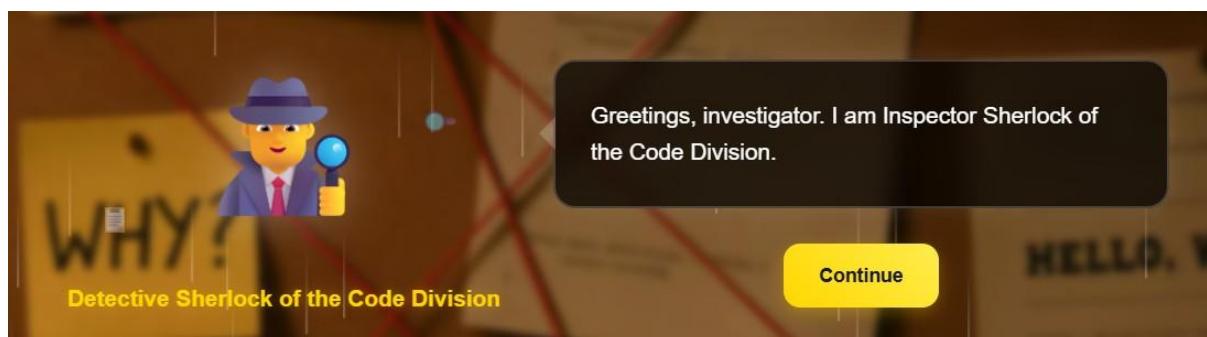
- **Wizard Academy:** Master Gandalf the Code Wizard
- **Detective Agency:** Inspector Sherlock of the Code Division
- **Space Station:** Commander Data, Chief Systems Officer

What Happens:

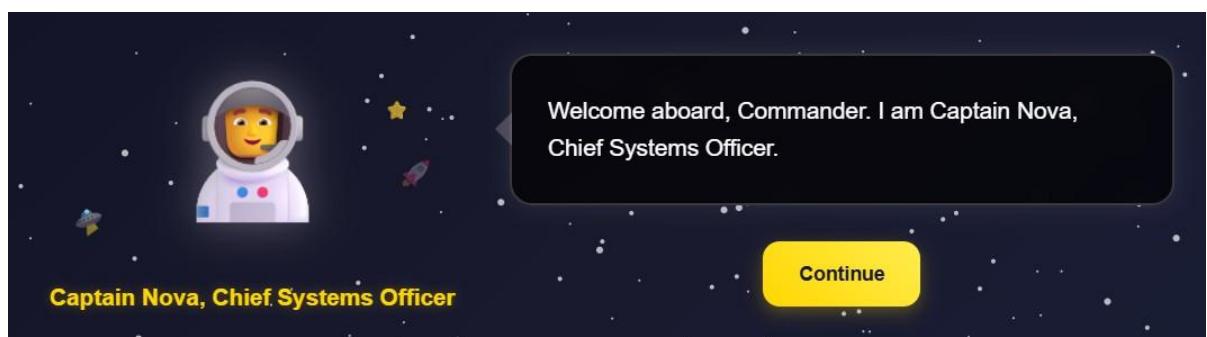
- Interactive dialogue with your mentor
- Tutorial on using the platform
- Introduction to your first lesson



Master Gandalf



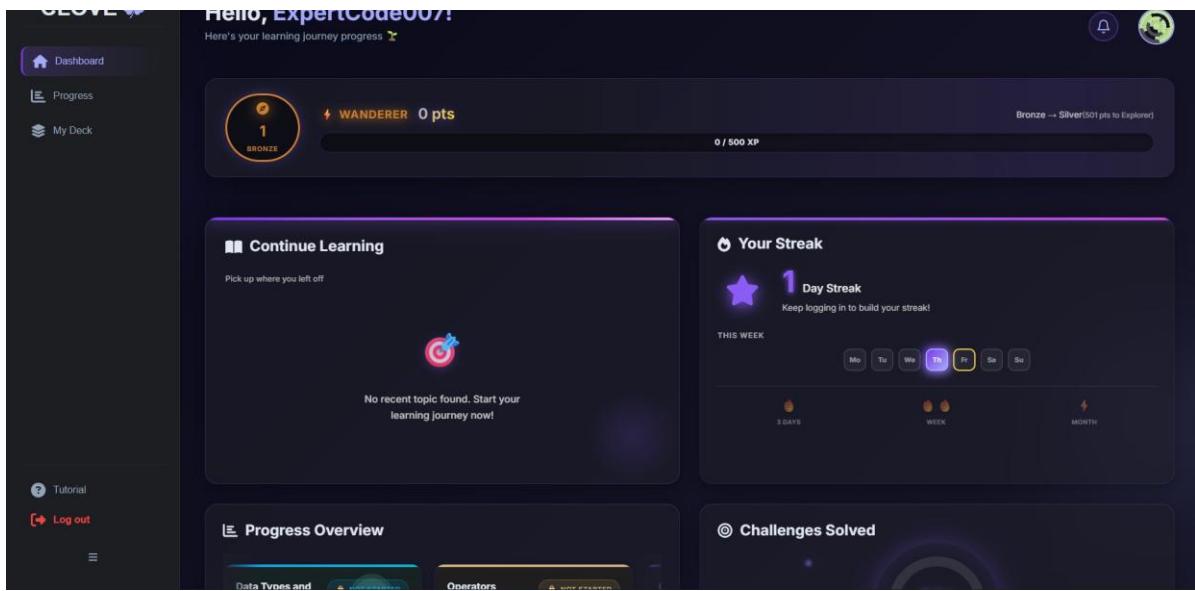
Inspector Sherlock



Captain Nova

3. First Steps After Onboarding

After completing onboarding, you'll be redirected to your **Dashboard**. Here's your action plan:

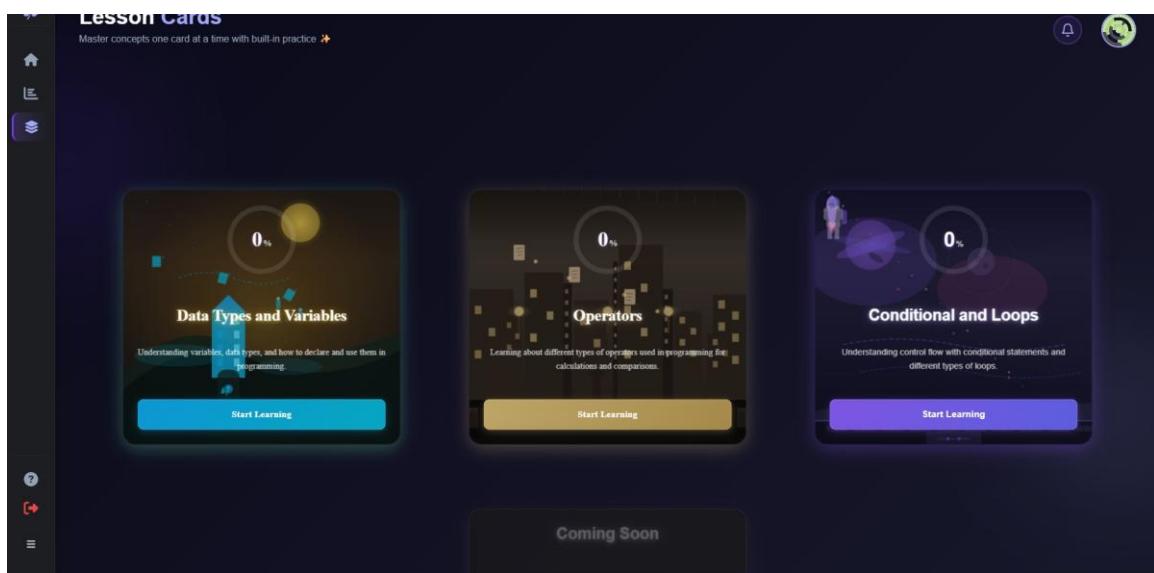


Dashboard Page

Recommended First Steps:

1. Explore “My Deck”

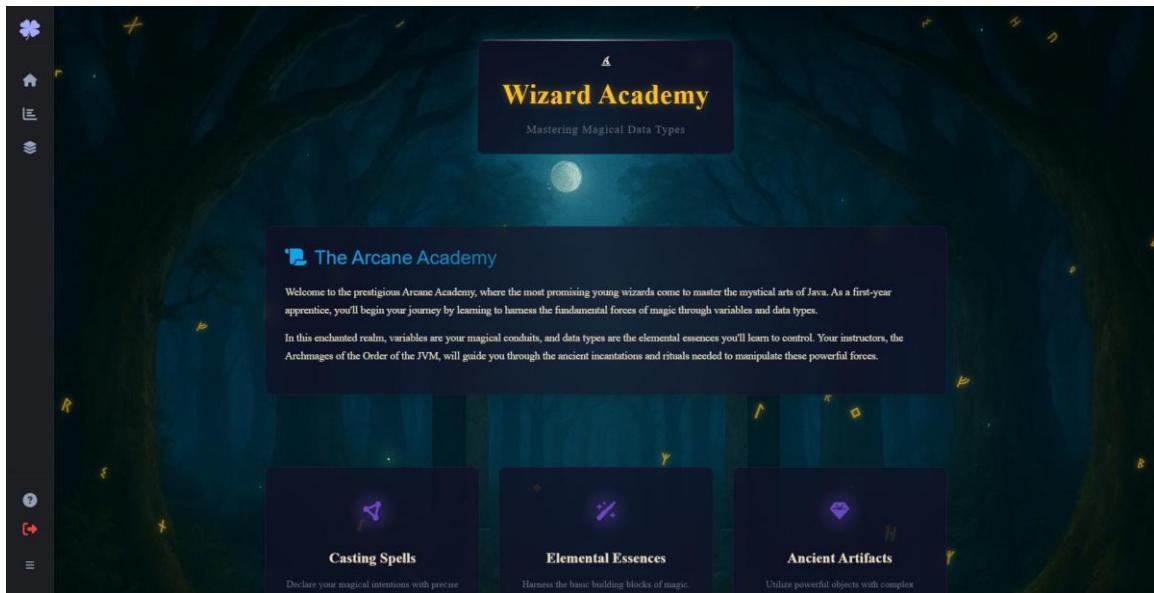
- Click on “My Deck” in the navigation
- Browse available lesson topics
- Notice the progress indicators on each card



My Deck Page

2. Start Your First Topic

- Click on a topic card (usually the first one)
- Read the topic introduction
- Familiarize yourself with the topic structure



Introduction Page

3. Begin Learning

- Start with the first subtopic
- Read the lesson content
- Complete practice exercises
- Attempt challenges

Learning Components

1. Pre-Assessment

Purpose: Evaluate your current knowledge level before starting a topic

Details:

- **When:** Starting New Topic
- **Format:** Multiple-choice questions
- **Duration:** 15 questions

Tips:

- Answer honestly - this helps the system personalize your experience

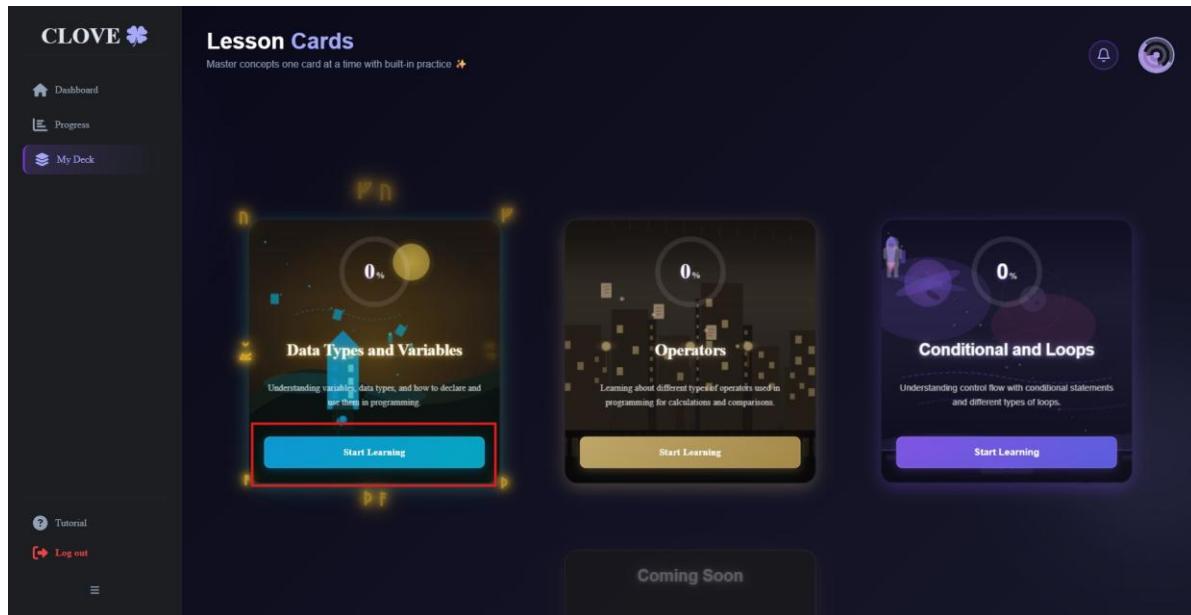
- Don't worry about getting everything right
- The assessment helps identify what you already know

Read Section 6 for detailed assessment test information

Step-by-Step: Taking a Pre-Assessment

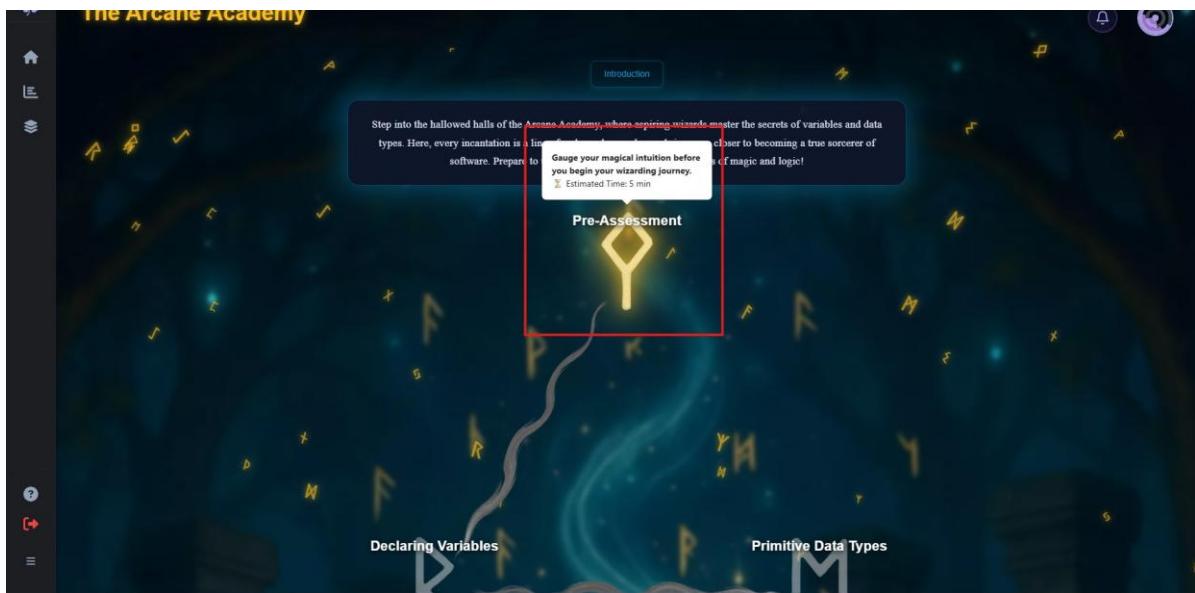
Step 1: Navigate to Topic

1. Log in to your CLOVE account
2. Click on “My Deck” in the main navigation
3. Browse the available topic cards
4. Click on the topic you want to start (usually the first unlocked topic)
5. You’ll see the topic introduction page



Step 2: Access Pre-Assessment

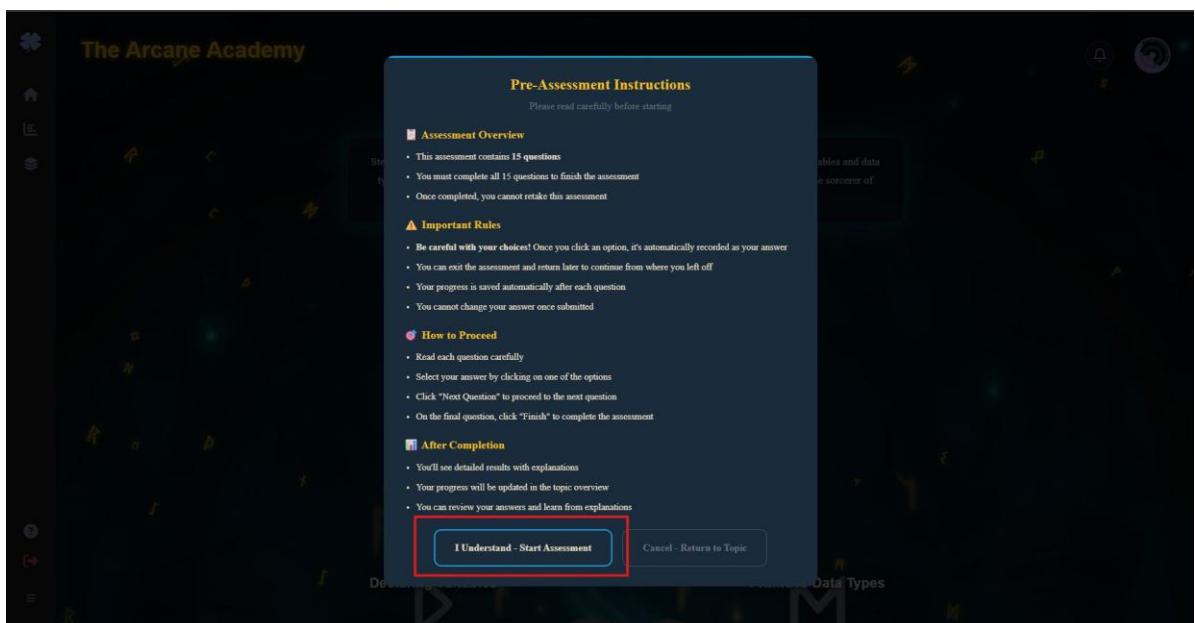
1. On the topic page, locate the “Pre-Assessment” section
2. You’ll see a button labeled “Pre-Assessment”
3. Click the button to begin
4. An instruction dialog will pop up.



Subtopics Page

Step 3: Read Instructions

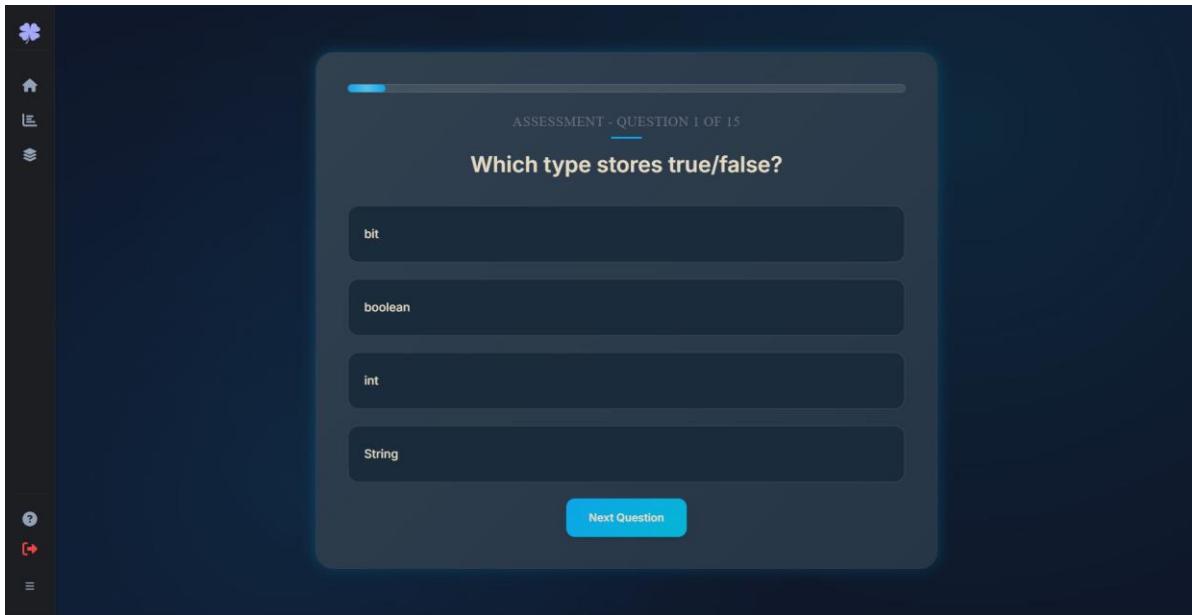
1. Review the pre-assessment instructions screen
2. Understand the format:
 - Multiple-choice questions
 - No time limit (take your time)
 - Answer all questions
3. Click "Start Assessment" when ready



Click "Start Assessment"

Step 4: Answer Questions

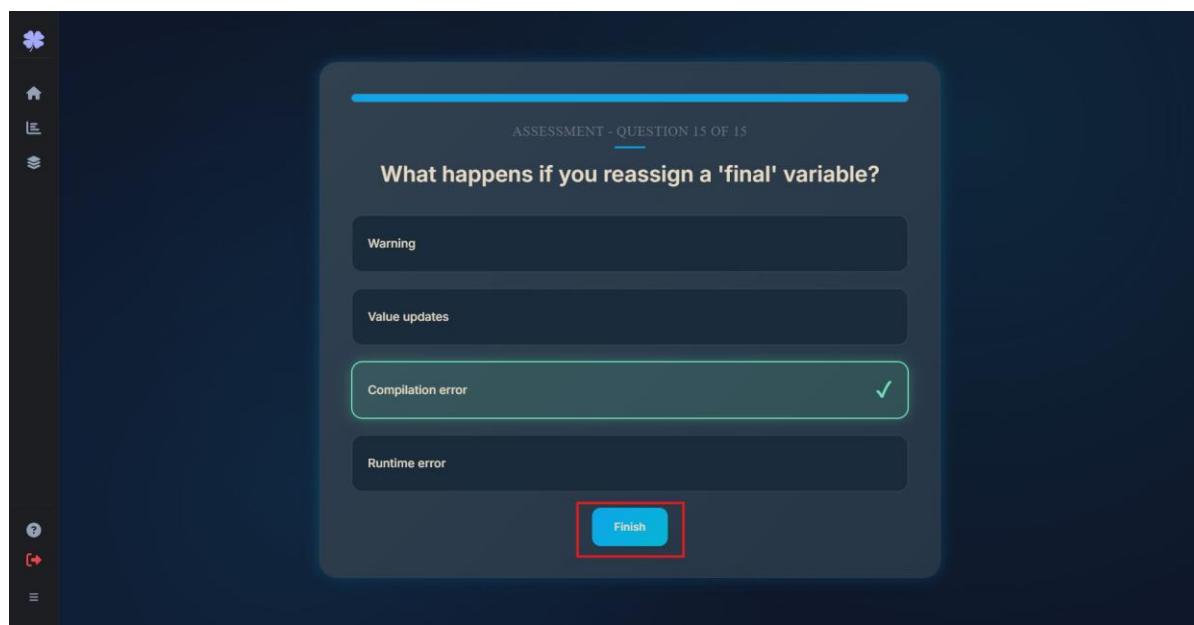
1. **Read Carefully:** Read each question thoroughly
2. **Review Code:** If the question includes code, analyze it carefully
3. **Consider Options:** Review all answer choices before selecting
4. **Select Answer:** Click on your chosen answer (clickable option)
5. **Navigate:** Use “**Next Question**” button to move to next questions. You must answer before moving to next question.
6. **Review Progress:** Check the question counter (e.g., “Question 3 of 15”)



Pre-Assessment Test

Step 5: Submit Assessment

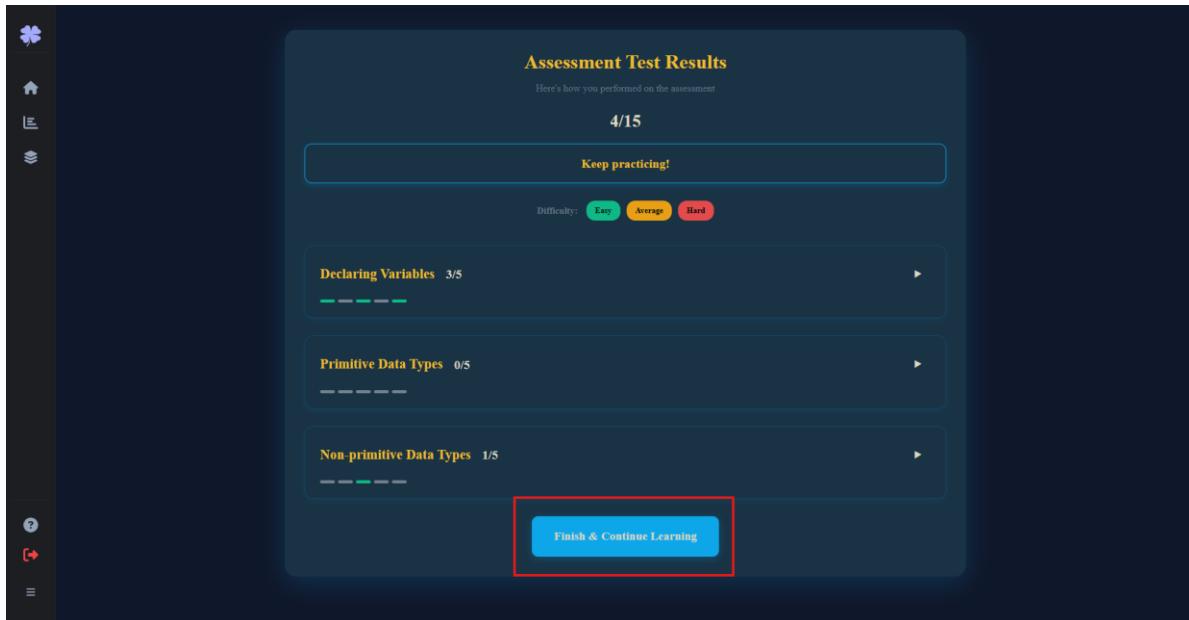
1. Once finished with all answers, click “**Finish**” button
2. Wait for the results to load (usually a few seconds)



Click “Finish”

Step 6: View Results

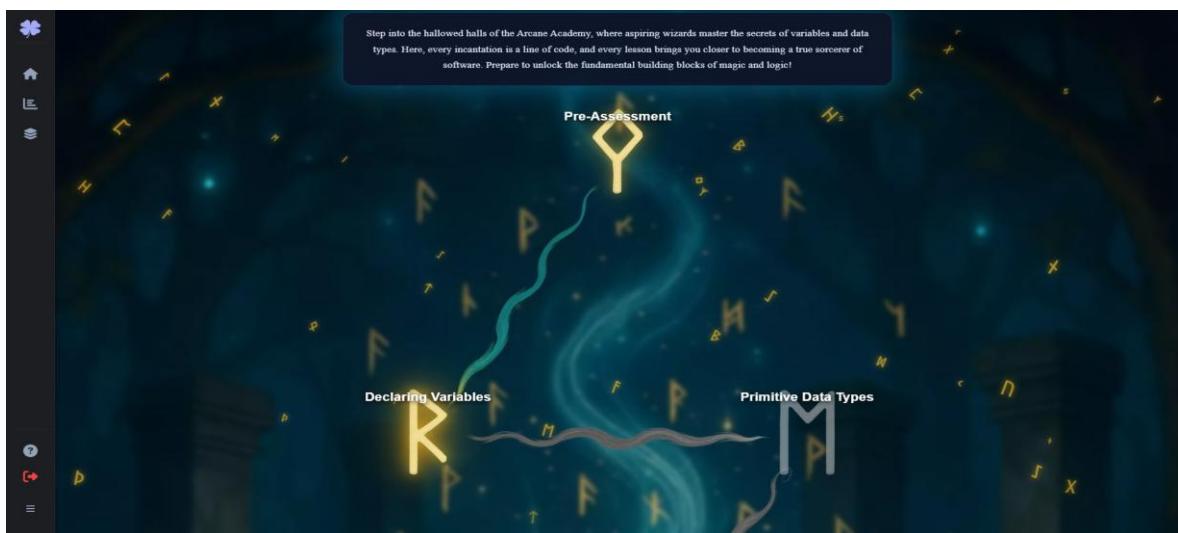
1. **Correct/Total:** View number of correct answers (e.g., "13 out of 15")
2. **Subtopic Breakdown:** See performance by subtopic area
 - Each subtopic shows your score
 - Visual indicators (green/yellow/red) show difficulty of the question
3. Click "**Finish & Continue Learning**" to go back to subtopic page.



Click "Finish & Continue Learning"

Step 7: Unlock First Subtopic

1. After viewing results, you'll automatically return to the subtopic page
2. Notice that the first subtopic are now **unlocked** (no longer grayed out)
3. You can now click it to start learning
4. The pre-assessment status will mark as "**Completed**" and it will show the result every time they click the pre-assessment icon.



Important Notes:

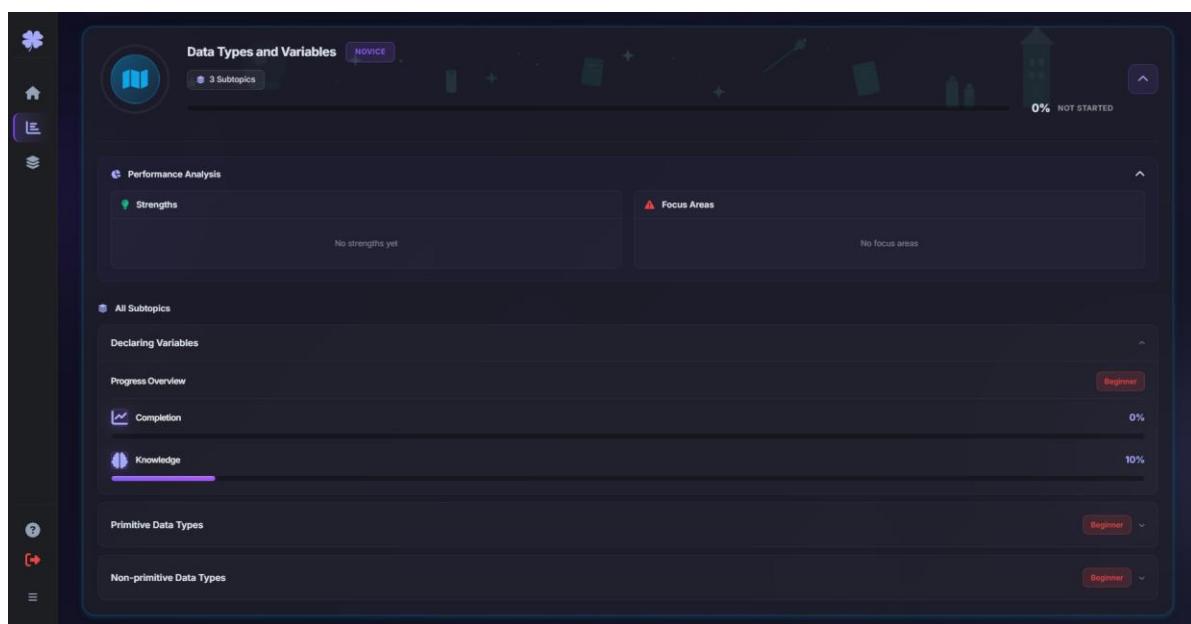
- **⚠ Answer Honestly:** Don't guess randomly - honest answers help personalize your learning
- **⚠ No Time Limit:** Take your time to think through each question
- **⚠ Can Retake:** You can retake the pre-assessment, but it uses the same questions
- **✓ Unlocks Everything:** Completing pre-assessment unlocks all subtopics immediately

2. Subtopics

Structure: Each topic contains multiple subtopics that build upon each other

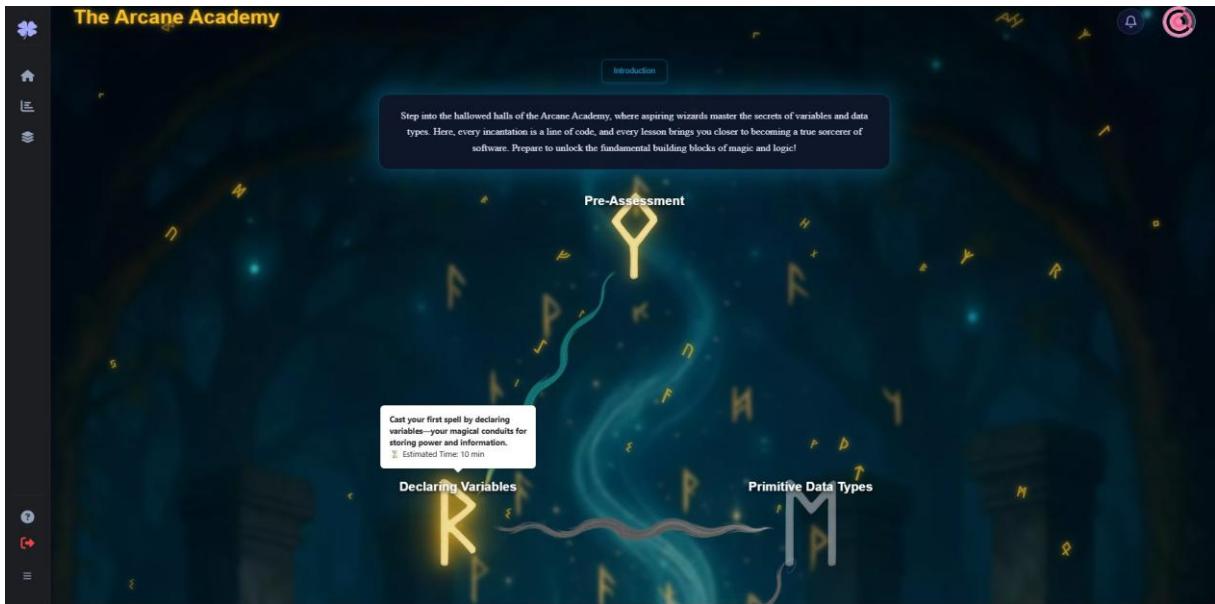
Features:

- **Unlocking:** All subtopics can be unlocked after completing pre-assessment
- **Order:** Work through subtopics sequentially for best results
- **Progress:** Track completion percentage for each subtopic



Navigation:

- At the Topic Page
- Click on any unlocked subtopic to access it
- Unlocked subtopics are colored and glowing



3. Lessons

Content: Educational materials for each subtopic

What's Included:

- **Theory:** Core concepts, explanations and summary
- **Code Snippets:** Code examples with explanations
- **Content Section:** Clickable menu on the right panel

Declaring Variables - Forging Magical Containers

Introduction

In the arcane arts of Java, declaring variables is akin to forging enchanted vessels or containers. Just as a wizard needs a specific chalice for a potent elixir or a dedicated pouch for rare spell components, a program needs defined spaces to hold different kinds of magical information. These 'variables' are named storage locations in the computer's memory, ready to hold specific types of data that your spells will manipulate. Declaring a variable tells the Java runtime: 'Prepare a space in memory, give it this name, and it will hold this type of magic.'

Learning Objectives

- Learn the incantation (syntax) for forging magical data containers in Java.
- Understand how to choose the correct data type for different kinds of magical energy.
- Master the art of giving variables clear, meaningful names using camelCase.
- Recognize the importance of declaring and initializing variables to prevent magical chaos.

Real-world Application

Imagine you're preparing for a grand ritual. You wouldn't just throw all your components onto the floor; you'd carefully organize them.

- You might have a `poisonVolume` container for liquid measurements.
- A `spellName` scroll to inscribe the incantation.
- A `manapoint` crystal to track your magical energy.
- A `hasWandEquipped` charm to know if your magical conduit is ready.

Each of these represents a variable, a designated container for a specific piece of magical data. Without declaring them, your magical workshop (program) would be in chaos, unable to differentiate between a shimmering aether or a mundane pebble.

- 1 Introduction
- 2 Real-world Application
- 3 Syntax
- 4 Example
- 5 How it Works
- 6 Example Code
- 7 Best Practices
- 8 Case (Pitfalls for the Unwary Apprentice)
- 9 Use Cases (When to Forge New Containers)
- 10 Best Practices (Wielder for Aspiring Arcane Engineers)
- 11 Summary

Lesson Page

How to Study:

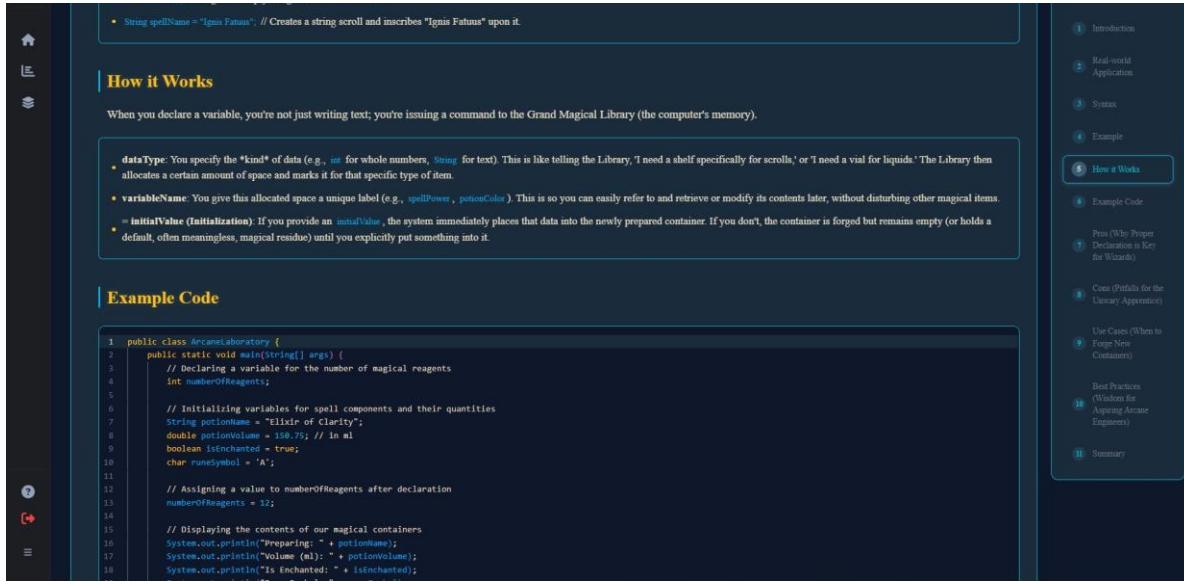
1. Read through the lesson content carefully

CLOVE User Manual

23

2. Review code examples
3. Try to understand each concept before moving on
4. Take notes if helpful
5. Mark the lesson as complete when finished

 Learning Tip: Don't rush through lessons. Take your time to understand each concept fully.



The screenshot shows a Java code editor with a sidebar containing navigation links for a lesson. The main area displays a code snippet with annotations:

```

1 String spellName = "Ignis Fatua"; // Creates a string scroll and inscribes "Ignis Fatua" upon it.

```

How it Works

When you declare a variable, you're not just writing text; you're issuing a command to the Grand Magical Library (the computer's memory).

Annotations explain:

- dataType**: You specify the *kind* of data (e.g., `int` for whole numbers, `String` for text). This is like telling the Library, 'I need a shelf specifically for scrolls,' or 'I need a vial for liquids.' The Library then allocates a certain amount of space and marks it for that specific type of item.
- variableName**: You give this allocated space a unique label (e.g., `spellPower`, `potionColor`). This is so you can easily refer to and retrieve or modify its contents later, without disturbing other magical items.
- = initialValue (Initialization)**: If you provide an `initialValue`, the system immediately places that data into the newly prepared container. If you don't, the container is forged but remains empty (or holds a default, often meaningless, magical residue) until you explicitly put something into it.

Example Code

```

1 public class ArcaneLaboratory {
2     public static void main(String[] args) {
3         // Declaring a variable for the number of magical reagents
4         int numberOfReagents;
5
6         // Initializing variables for spell components and their quantities
7         String potionName = "Elixir of Clarity";
8         double potionVolume = 150.75; // in ml
9         boolean isEnchanted = true;
10        char runesymbol = 'A';
11
12        // Assigning a value to numberOfReagents after declaration
13        numberOfReagents = 12;
14
15        // Displaying the contents of our magical containers
16        System.out.println("Preparing: " + potionName);
17        System.out.println("Volume (ml): " + potionVolume);
18        System.out.println("Is Enchanted: " + isEnchanted);
19    }
}

```

The sidebar contains the following navigation links:

- 1 Introduction
- 2 Real-world Application
- 3 Syntax
- 4 Example
- 5 How it Works
- 6 Example Code
- 7 Pros (Why Proper Declaration is Key for Wizards)
- 8 Cons (Pitfalls for the Unwary Apprentice)
- 9 Use Cases (When to Forge New Containers)
- 10 Best Practices (Wisdom for Aspiring Arcane Engineers)
- 11 Summary

Example Code in Lessons

Step-by-Step: Completing a Lesson

Complete Walkthrough:

Step 1: Access a Subtopic

1. After completing the pre-assessment, navigate to the subtopic page
2. Click on the **first subtopic**
3. The subtopic page will load



Click the first unlocked subtopic

Step 2: Navigate to Lesson

1. The lesson content will display
2. You can click the table of contents at the right side panel.

Table of Contents of the lesson on the right side panel

Declaring Variables - Forging Magical Containers

Introduction

In the arcane arts of Java, declaring variables is akin to forging enchanted vessels or containers. Just as a wizard needs a specific chalice for a potent elixir or a dedicated pouch for rare spell components, a program needs defined spaces to hold different kinds of magical information. These ‘variables’ are named storage locations in the computer’s memory, ready to hold specific types of data that your spells will manipulate. Declaring a variable tells the Java runtime: ‘Prepare a space in memory, give it this name, and it will hold this type of magic.’

Learning Objectives

- Learn the incantation (syntax) for forging magical data containers in Java.
- Understand how to choose the correct data type for different kinds of magical energy.
- Master the art of giving variables clear, meaningful names using camelCase.
- Recognize the importance of declaring and initializing variables to prevent magical chaos.

Real-world Application

Imagine you’re preparing for a grand ritual. You wouldn’t just throw all your components onto the floor; you’d carefully organize them.

- You might have a `potionVolume` container for liquid measurements.
- A `spellName` scroll to inscribe the incantation.
- A `manaPoints` crystal to track your magical energy.

CONTENTS

- 1 Introduction
- 2 Real-world Application
- 3 Syntax
- 4 Example
- 5 How it Works
- 6 Example Code
- 7 Pros (Why Proper Declaration is Key for Wizards)
- 8 Cons (Pitfalls for the Unwary Apprentice)
- 9 Use Cases (When to Forge New Containers)
- 10 Best Practices (Wisdom for Aspiring Arcane Engineers)
- 11 Summary

Step 3: Read the Introduction

1. Start by reading the **lesson title** and **introduction**
2. Understand what concepts will be covered
3. Note any **learning objectives** listed

Step 4: Study the Content Sections

1. **Read Each Section:** Work through the lesson content section by section
2. **Understand Concepts:**

- Read explanations carefully
- Don't skip ahead if something is unclear
- Re-read difficult sections if needed

3. Review Examples:

- Study each code example provided
- Understand what the code does
- Note the syntax and structure

4. Check Visual Aids:

- Review diagrams or illustrations
- Understand how they relate to the text

5. Scroll Progressively: Scroll down to read all content

Step 5: Analyze Code Examples

1. Read the Code: Look at each code snippet carefully

2. Understand Syntax:

- Note keywords, operators, and structure
- Understand variable names and types
- See how statements are organized

3. Read Explanations:

- Read the explanation below or beside the code
- Understand what each part does
- Connect the code to the concept

4. Try to Predict: Before reading the explanation, try to understand what the code does

5. Review Multiple Times: Re-read code examples if needed

The screenshot shows a user interface for code analysis. On the left, there's a sidebar with icons for home, help, and navigation. The main area has a title 'Example Code' and a code editor with the following Java code:

```

1 public class ArcaneLaboratory {
2     public static void main(String[] args) {
3         // Declaring a variable for the number of magical reagents
4         int numberOfReagents;
5
6         // Initializing variables for spell components and their quantities
7         String potionName = "Elixir of Clarity";
8         double potionVolume = 150.75; // in ml
9         boolean isEnchanted = true;
10        char runeSymbol = 'A';
11
12        // Assigning a value to numberOfReagents after declaration
13        numberOfReagents = 12;
14
15        // Displaying the contents of our magical containers
16        System.out.println("Preparing: " + potionName);
17        System.out.println("Volume (ml): " + potionVolume);
18        System.out.println("Is Enchanted: " + isEnchanted);
19        System.out.println("Rune Symbol: " + runeSymbol);
20        System.out.println("Total Reagents: " + numberOfReagents);
21    }
22 }

```

Below the code editor is an 'OUTPUT' section showing the console output:

```

Preparing: Elixir of Clarity
Volume (ml): 150.75
Is Enchanted: true
Rune Symbol: A
Total Reagents: 12

```

To the right of the main area is a 'CONTENTS' sidebar with a table of contents:

1	Introduction
2	Real-world Application
3	Syntax
4	Example
5	How it Works
6	Example Code
7	Pros (Why Proper Declaration is Key for Wizards)
8	Cons (Pitfalls for the Unwary Apprentice)
9	Use Cases (What to Forge New Containers)
10	Best Practices (Wisdom for Aspiring Arcane Engineers)
11	Summary

Sample Code Snippet

Step 6: Take Notes (Optional but Recommended)

1. **Open a Notepad:** Keep a notebook or digital note-taking app open
2. **Write Key Concepts:** Jot down important concepts
3. **Copy Code Examples:** Write down code examples you find useful
4. **Note Questions:** Write down any questions you have
5. **Create Summary:** Summarize each section in your own words

Step 7: Review and Understand

1. **Check Understanding:** After reading, ask yourself:
 - Do I understand the main concept?
 - Can I explain it in my own words?
 - Do I understand the code examples?
2. **Re-read if Needed:** If something is unclear, re-read that section
3. **Look for Connections:** Connect new concepts to things you already know
4. **Don't Rush:** Take your time - understanding is more important than speed

Step 8: Mark as Complete

1. After finishing the lesson content, scroll to the bottom
2. Look for a “**Start Practice**” button

The screenshot shows the CLOVE learning interface. On the left is a vertical sidebar with icons for navigation. The main content area has a dark background with light-colored text. At the top, there's a summary of key takeaways, followed by a 'Start Practice' button which is highlighted with a red box. To the right, there's a sidebar with a list of topics and sub-topics, each preceded by a small numbered icon.

- 1 Real-world Application
- 2 Syntax
- 3 Example
- 4 How it Works
- 5 Example Code
- 6 Best (Why Proper Declaration is Key for Wizards)
- 7 Cons (Pitfalls for the Unwary Apprentice)
- 8 Use Cases (When to Forge New Containers)
- 9 Best Practices (Wisdom for Aspiring Arcane Engineers)
- 10 Summary

Click “Start Practice”

Step 9: Move to Practice

1. After completing the lesson, you're ready for practice
2. Click on the “**Start Practice**” tab or section

3. Start with the first practice exercise
4. Apply what you learned in the lesson

Best Practices:

- **Read Actively:** Don't just skim - engage with the content
- **Take Your Time:** There's no rush - understanding is key
- **Review Code:** Spend time understanding code examples
- **Take Notes:** Writing helps retention
- **Ask Questions:** If confused, re-read or take a break
- **Complete Fully:** Don't skip sections - they build on each other

Common Mistakes to Avoid:

- Rushing through content without understanding
- Skipping code examples
- Not taking notes
- Moving on when confused
- Not marking lessons as complete

4. Practice Exercises

Purpose: Apply what you've learned in a low-pressure environment

Features:

- **Types:** Practice problems with different game modes
- **Feedback:** Immediate feedback on your answers
- **Retry:** You can practice multiple times
- **No Penalties:** Practice doesn't affect your score

PRACTICE CHALLENGES
Complete interactive coding challenges to test your understanding and improve your programming skills.

SCENARIO:
This section displays the scenario description for each challenge. The scenario provides context and explains what you need to accomplish in the current challenge.

CODE FIXER CHALLENGE

INSTRUCTION:

- Edit the code to fix all syntax errors and logical issues
- Make sure the code produces the expected output
- Check for missing semicolons, brackets, and other syntax elements
- Verify that variable declarations and method calls are correct

EDIT THE CODE:

```
public class FateInscriber {
    public static void main(String[] args) {
        String fateMessage = "Prophecy of Dawn";
        System.out.println("Initial Fate: " + fateMessage);
        //I'm editing the code.
        fateMessage = "Prophecy of Dusk";
        System.out.println("Altered Fate: " + fateMessage);
    }
}
```

Expected Output:
Initial Fate: Prophecy of Dawn

SUBMIT

Code Fixer practice problems with 2 remaining attempts

Best Practices:

- Complete all practice exercises
- Review incorrect answers
- Practice until you're comfortable
- Use practice to prepare for challenges

Step-by-Step: Completing Practice Exercises

Complete Walkthrough:

Step 1: Navigate to Practice Section

1. After completing a lesson, click on the “Start Practice” tab or section
2. You’ll see a three practice exercises for the current subtopic
3. Exercises in different challenge types (Code Fixer, Code Completion, Output Tracing)
4. Look on the first practice exercise to begin

The screenshot shows a user interface for a "CODE FIXER CHALLENGE". On the left, there's a sidebar with navigation icons and a "PRACTICE CHALLENGES" section. The main area has a "SCENARIO" box describing a task to fix code for a "PortalConfigurator" class. Below it is an "INSTRUCTION" box with four items: "Edit the code to fix all syntax errors and logical issues", "Make sure the code produces the expected output", "Check for missing semicolons, brackets, and other syntax elements", and "Verify that variable declarations and method calls are correct". Underneath is a "EDIT THE CODE:" text area containing Java code for the "PortalConfigurator" class. At the bottom, there's an "Expected Output:" box showing the expected console output: "Portal Destination: XYZ-789".

Code Fixer Practice Challenge

Step 2: Read the Practice Question

1. **Read the Question:** Carefully read the practice question or problem statement
2. **Understand the Task:**
 - What are you being asked to do?
 - What information is provided?
 - What is the expected outcome?
3. **Review Code (if applicable):**
 - If code is provided, read and understand it

- Identify what the code does
 - Note any specific requirements
4. **Check Examples:** Look for example inputs/outputs if provided

Step 3: Analyze the Question

- Identify Key Concepts:** What lesson concepts does this exercise test?
- Break Down the Problem:**
 - What are the steps to solve this?
 - What information do you need?
 - What's the logic required?
- Think Through Solution:**
 - Mentally work through the solution
 - Consider different approaches
 - Think about edge cases

Step 4: Review Answer Options

- Read All Options:** Read through all available answer choices
- Eliminate Obvious Wrong Answers:** Cross out clearly incorrect options
- Compare Options:** Compare remaining options carefully
- Consider Each Option:** Think about why each option might be correct or wrong

Step 5: Select Your Answer

- Choose Your Answer:** Click on the answer you believe is correct, or drag and drop your answer, or fix the code syntax.
- Confirm Selection:** Your answer provided should be reflected
- Double-Check:** Review your selection before submitting
- Click Submit:** Click the "Submit" button

The screenshot shows a user interface for a coding challenge. At the top, there are three items to check: 'Edit the code to fix all syntax errors and logical issues', 'Make sure the code produces the expected output', 'Check for missing semicolons, brackets, and other syntax elements', and 'Verify that variable declarations and method calls are correct'. Below this is a 'SCENARIO:' section with a description: 'This section displays the scenario description for each challenge. The scenario provides context and explains what you need to accomplish in the current challenge.' There are two buttons: 'Back to Lessons' and 'Go to Challenge'. The main area is titled 'EDIT THE CODE:' and contains the following Java code:

```

1 public class PortalConfigurator {
2     public static void main(String[] args) {
3         String destinationCoordinates = "XYZ-789";
4         System.out.println("Portal Destination: " + destinationCoordinates);
5         //I'm editing now...
6     }
7 }

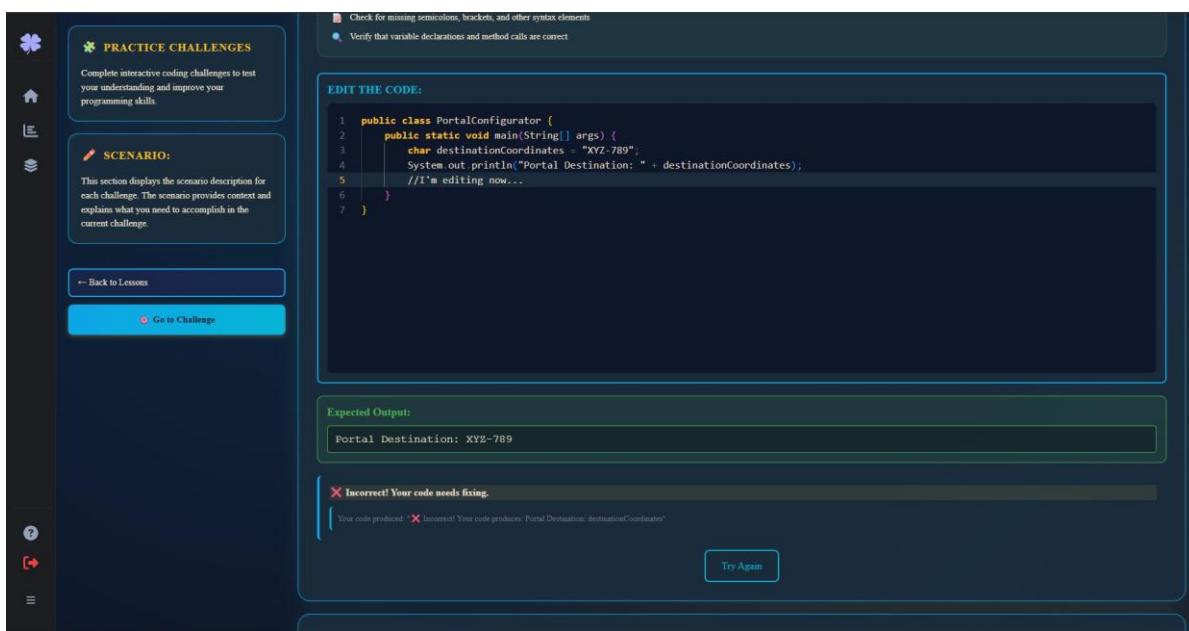
```

Below the code editor is an 'Expected Output:' field containing 'Portal Destination: XYZ-789'. At the bottom is a large blue 'SUBMIT' button, which is highlighted with a red border.

Click "Submit"

Step 6: Review Feedback

1. **Immediate Feedback:** After submitting, you'll see immediate feedback
2. **Correct Answer:**
 - If correct, you'll see a success message
 - May see points or positive reinforcement
3. **Incorrect Answer:**
 - If incorrect, you'll see which answer was correct
 - Explanation will be provided
 - Learn from the mistake
4. **Read Explanation:**
 - Always read the explanation, even if you got it right
 - Understand why the correct answer is correct
 - Learn from the reasoning



Step 7: Learn from Mistakes

1. **If Incorrect:**
 - Read the explanation carefully
 - Understand why your answer was wrong
 - Understand why the correct answer is right
 - Review the related lesson content if needed
2. **Review Concepts:**
 - Go back to the lesson if the concept is unclear
 - Take notes on what you learned
 - Make sure you understand before moving on

Step 8: Continue to Next Exercise

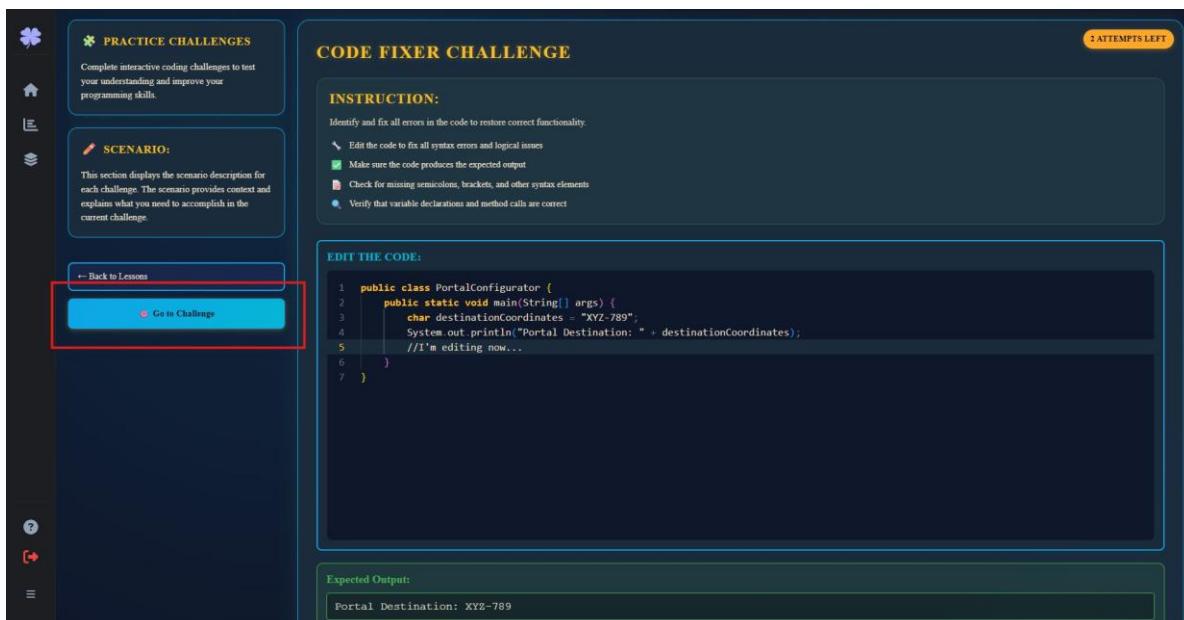
1. After reviewing feedback, look for next practice problems
2. Repeat steps 2-7 for each exercise
3. Complete all practice exercises in the subtopic

Step 9: Retry if Needed

1. **Can Retry:** Practice exercises can usually be retaken
2. **If Struggling:**
 - If you got many wrong, consider retrying
 - Review the lesson content again
 - Try the exercises again after understanding better
3. **No Penalties:** Retrying doesn't hurt your score (practice is for learning)

Step 10: Complete All Practices

1. **Finish All:** Complete all practice exercises in the subtopic
2. **Review Summary:** Some subtopics show a practice summary
3. **Move to Challenges:** Once comfortable with practice, try challenges



Click "Go to Challenge"

Tips for Success:

- **Take Your Time:** No time limit - think carefully
- **Read Explanations:** Always read feedback explanations
- **Learn from Mistakes:** Mistakes are learning opportunities
- **Retry if Needed:** Don't hesitate to retry exercises
- **Review Lessons:** Go back to lessons if concepts are unclear
- **Complete All:** Don't skip practice exercises

What Practice Helps With:

- Reinforcing lesson concepts
- Building confidence
- Identifying areas that need more study
- Preparing for challenges
- Understanding application of concepts

5. Challenges

Purpose: Test your skills with interactive coding challenges

Challenge Types:

1. **Code Fixer:** Identify and correct bugs in provided code
2. **Code Completion:** Complete missing code segments
3. **Output Tracing:** Predict program output and trace execution

Features:

- Adaptive difficulty based on your performance
- Smart hints when you're struggling
- Optional timer for added pressure
- Points and scoring system

The screenshot shows a challenge interface titled "OUTPUT TRACING CHALLENGE". At the top right, it says "Challenge 1 of 5". The main area is labeled "CODE TO ANALYZE:" and contains the following Java code:

```
1 public class ElementalConduit {
2     public static void main(String[] args) {
3         int flowRate = 200;
4         String conduitState = "Open";
5         boolean stabilizerEngaged = false;
6
7         if (flowRate > 150) {
8             flowRate = flowRate / 2;
9             if (conduitState.equals("Open")) {
10                 stabilizerEngaged = true;
11                 conduitState = "Stabilized";
12             } else {
13                 stabilizerEngaged = false;
14             }
15     }
16 }
```

Below the code, there is a question: "What is the output of the following code?". A note says "Selected: 3 option(s)". Below the question are five options: "Final Flow Rate: 100", "Stabilizer Engaged: false", "Final Flow Rate: 150", "Conduit State: Stabilized!", "Stabilizer Engaged: true", and "Conduit State: Open!". At the bottom is a blue "SUBMIT" button.

See Section 6 for detailed challenge information

6. Post-Assessment

Purpose: Verify knowledge after completing all subtopics

Details:

- **When:** After finishing the entire topic
- **Format:** Multiple-choice questions (same questions in pre-assessments)
- **Duration:** 15 questions

Tips:

- Answer honestly - This helps the system track your growth
- Don't Worry About Getting Everything Right - It's about understanding progress
- The Post-Assessment Helps You See Improvement - Compare your pre-assessment to this one

Read Section 6 for detailed assessment test information

Step-by-Step: Taking a Post-Assessment

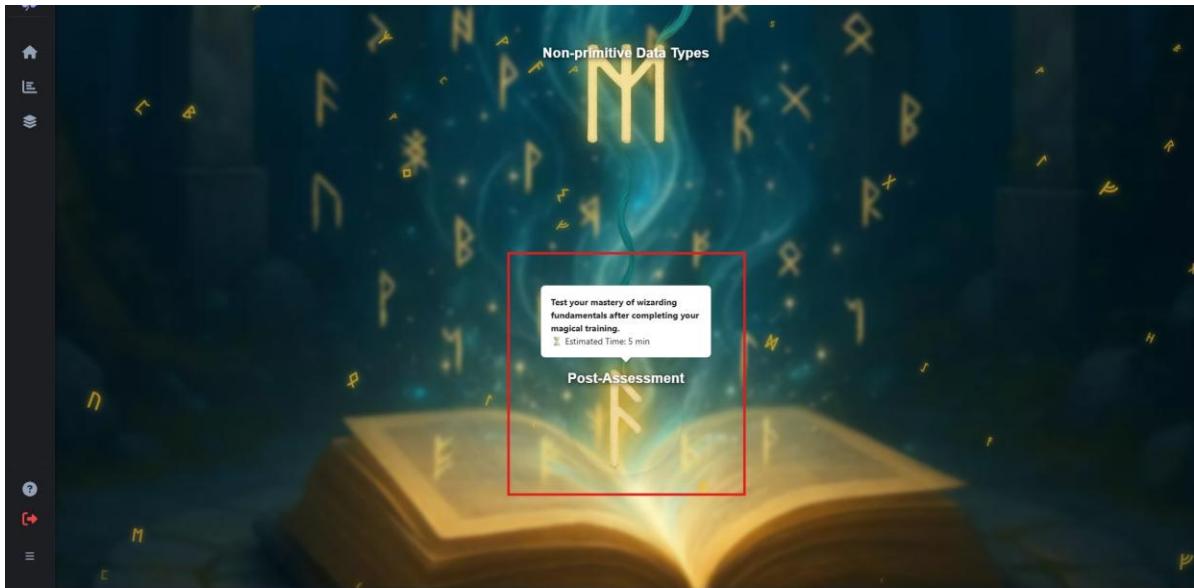
Complete Walkthrough:

Step 1: Complete All Requirements

1. **Finish All Subtopics:** Complete all subtopics in the topic
 - Complete all lessons
 - Finish all practice exercises
 - Attempt challenges (optional but recommended)
2. **Verify Completion:** Check that all subtopics show as "Completed"
3. **Return to Topic Page:** Navigate back to the main topic page

Step 2: Access Post-Assessment

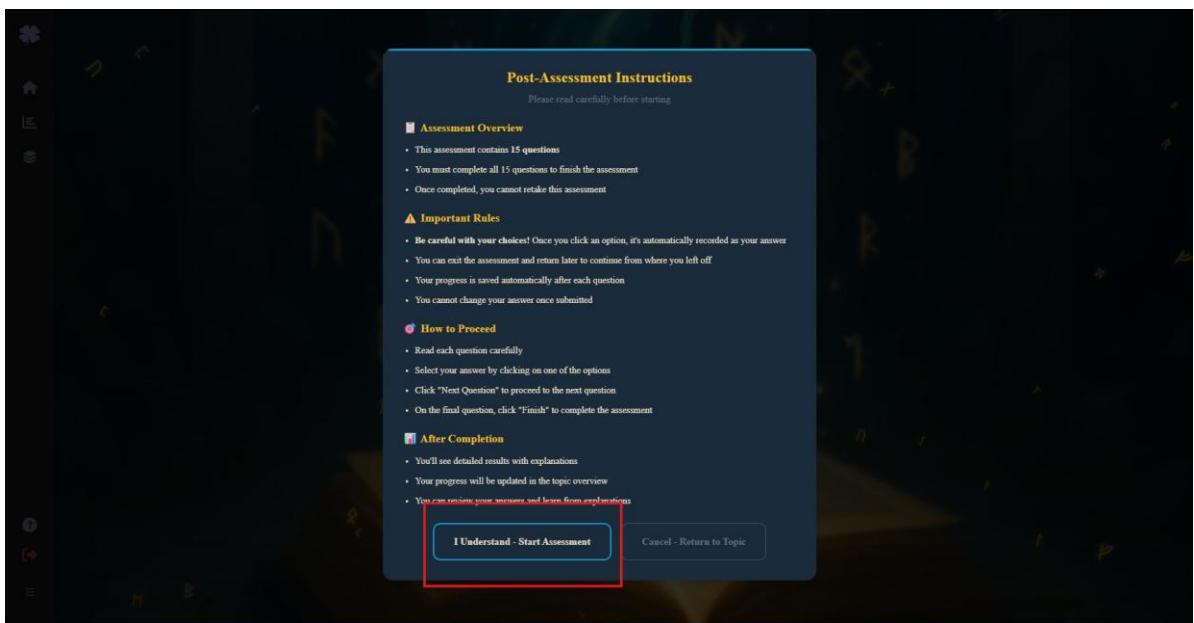
1. On the topic page, locate the "**Post-Assessment**" section
2. The section should now be **available** (no longer locked)
3. Click the button to begin
4. An instruction dialog will pop up.



Post-Assessment

Step 3: Understand the Format

- 1. Same Questions:** The post-assessment uses the **same questions** as the pre-assessment
- 2. Purpose:** This allows you to compare your improvement
- 3. No Time Limit:** Take your time to answer thoughtfully
- 4. Answer All:** Make sure to answer all questions

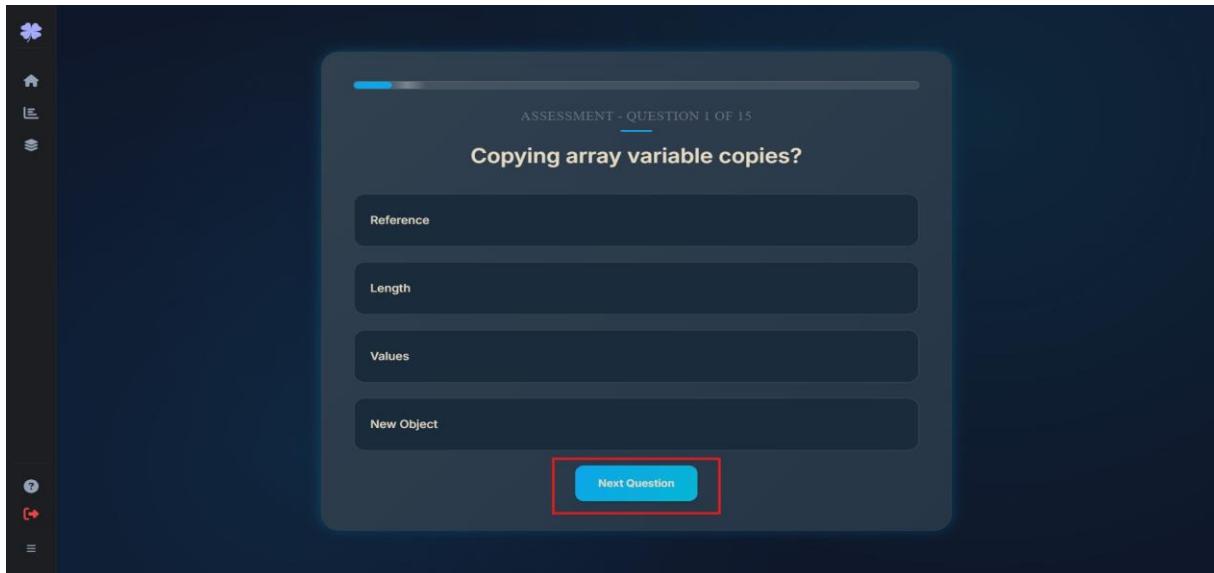


Click "Start Assessment"

Step 4: Answer Questions Thoughtfully

- 1. Read Each Question:** Read questions carefully (you may remember them from pre-assessment)
- 2. Apply Your Learning:**
 - Use what you learned from lessons

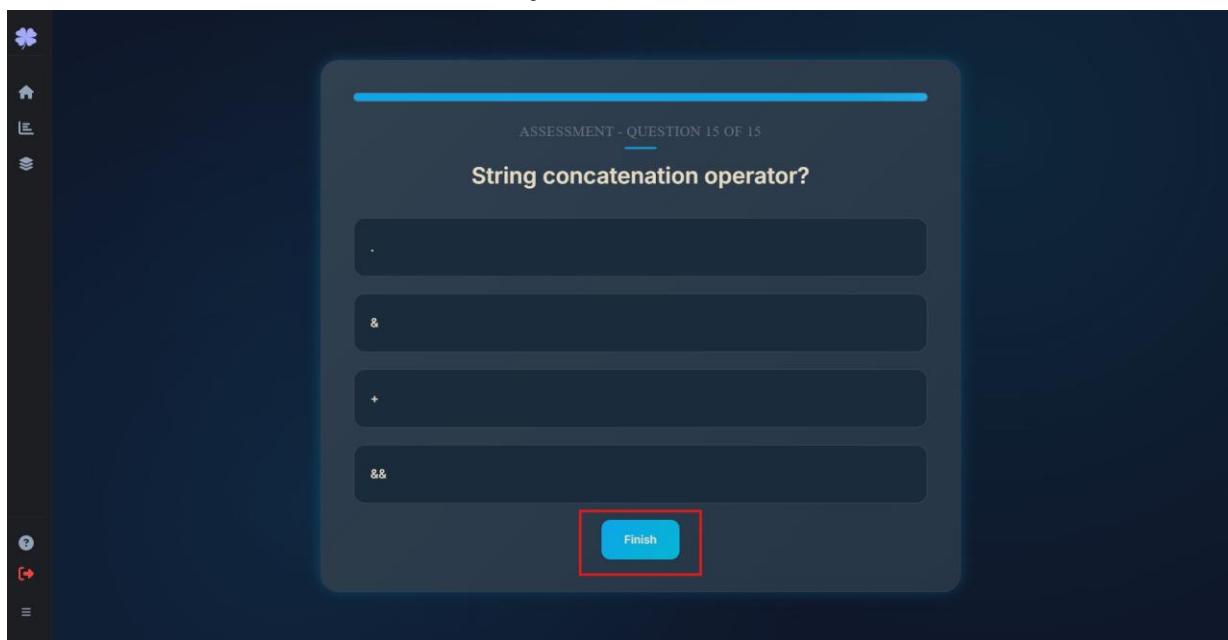
- Apply concepts from practice exercises
 - Think about the subtopics you completed
- Select Answers:** Choose your answers based on your new knowledge
 - Navigate:** Use “Next Question” button to move to the next question
 - Review Progress:** Check the question counter



Click “Next Question”

Step 5: Submit Post-Assessment

1. Once finished with all answers, click “Finish” button
2. Wait for the results to load (usually a few seconds)

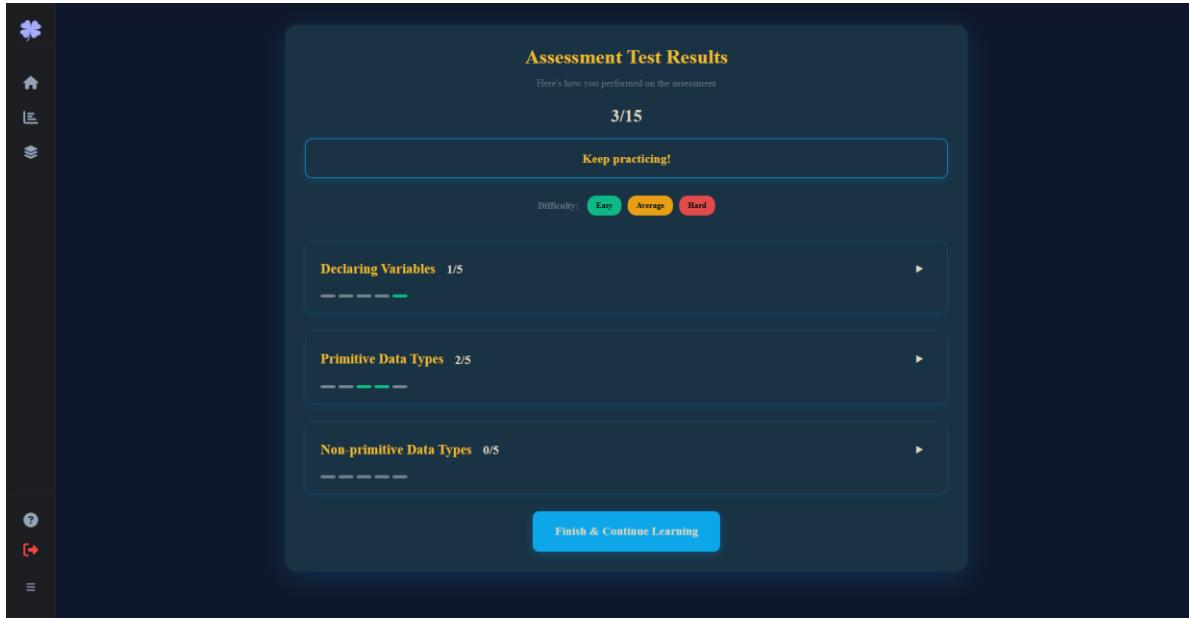


Click “Finish”

Step 6: View Your Results

- Correct Answers:** View number correct (e.g., “17 out of 20”)
- Subtopic Breakdown:** See performance by subtopic area

- Each subtopic shows your score
 - Visual indicators (green/yellow/red) show difficulty of the question
- Click “Finish & Continue Learning” to go back to subtopic page



Click “Finish & Continue Learning”

Step 7: Compare with Pre-Assessment

- Improvement Calculation:** See how much you improved
 - Example: Pre: 60% * Post: 85% □ 25% improvement
- Subtopic Comparison:** See improvement in each subtopic area

Step 8: Analyze Your Learning

- Areas of Improvement:**
 - See which subtopics you improved in most
 - Identify concepts you mastered
- Areas Needing Review:**
 - Note subtopics where improvement was minimal
 - Consider reviewing those areas again
- Overall Progress:**
 - Celebrate your improvement!
 - Note your learning journey

Step 9: Complete the Topic

- After viewing results, you've **completed the topic!**
- The topic status will update to “**Completed**”
- The topic card in “My Deck” will show as completed

Step 10: Retention Test Availability

1. After completing the topic, **retention tests** become available
 - **Stage 1:** Available **10 hours** after completion
 - **Stage 2:** Available **5 days** after completion
 2. Look for the retention test notification in the upper right corner of the page.
 3. See Section 5 for detailed retention test instructions
-

7. Retention Tests

Purpose: Test long-term knowledge retention

Timing Schedule:

- **Stage 1:** Available **10 hours** after completing a topic
- **Stage 2:** Available **5 days (120 hours)** after completing a topic

Format:

- 15 questions per stage
- Countdown timer showing availability
- Results tracking and analytics

Read Section 6 for detailed retention test information

Step-by-Step: Taking Retention Tests

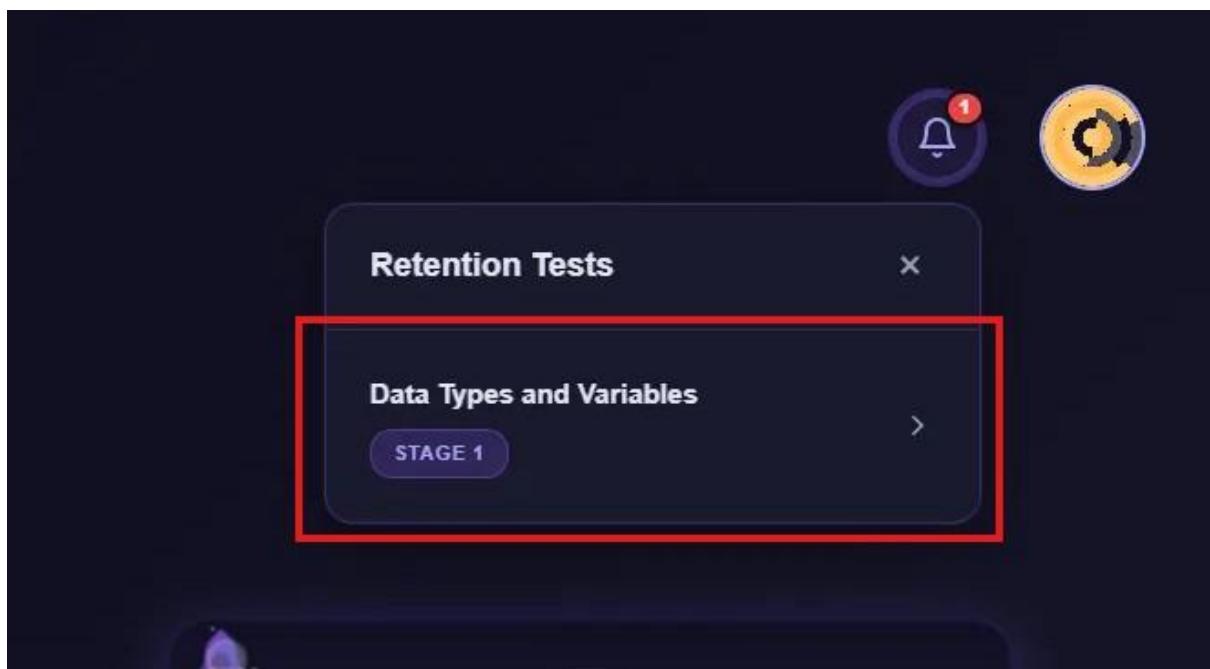
Complete Walkthrough:

Step 1: Complete Topic Requirements

1. **Finish Pre-Assessment:** Complete the pre-assessment for the topic
2. **Complete All Subtopics:**
 - Finish all lessons
 - Complete all practice exercises
 - Attempt challenges
3. **Complete Post-Assessment:** Finish the post-assessment
4. **Topic Marked Complete:** The topic should show as “Completed”

Step 2: Wait for Availability Window

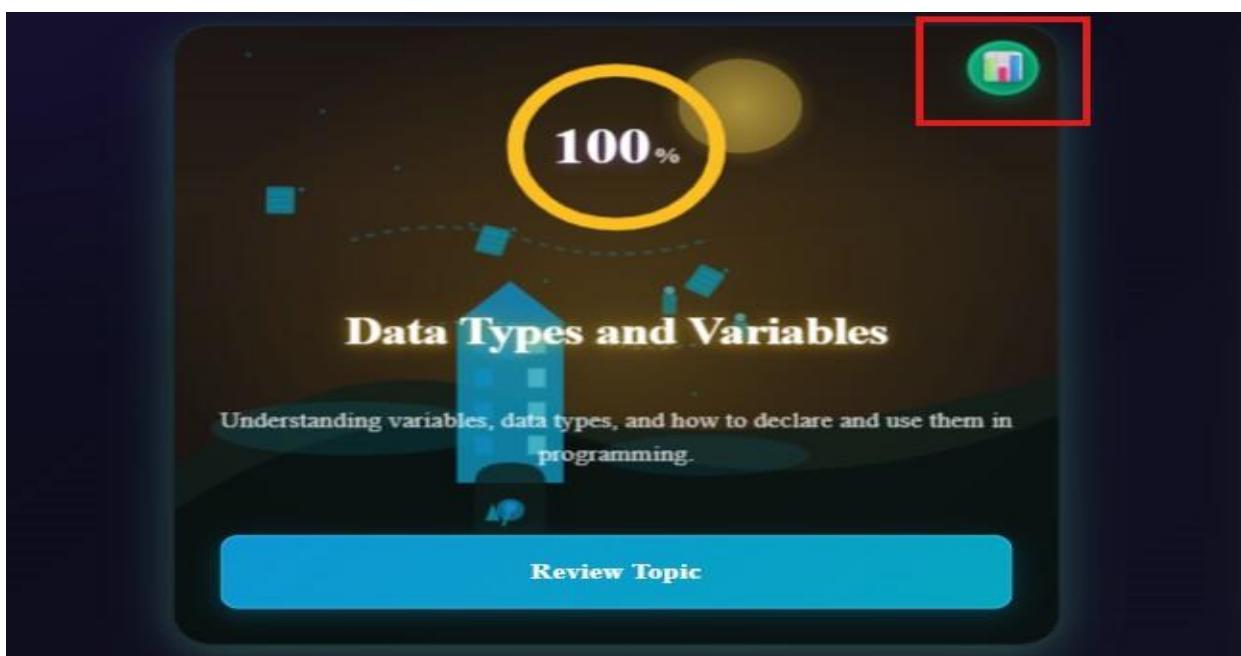
1. **Stage 1 Timing:** Wait **10 hours** after topic completion
2. **Stage 2 Timing:** Wait **5 days (120 hours)** after topic completion
3. **Countdown Timer:** Check the countdown on the topic card
4. **Notification:** You may receive a notification when tests become available



Retention Test Notification

Step 3: Locate Retention Test Icon (If stage 1 is Completed)

1. Go to "My Deck" page
2. Find the completed topic card
3. Look for the retention test icon () on the card
4. The icon appears after topic completion



Click the retention test icon

Step 4: Check Availability

1. Click the Icon: Click on the retention test icon
2. View Popup: A popup or modal will appear showing:

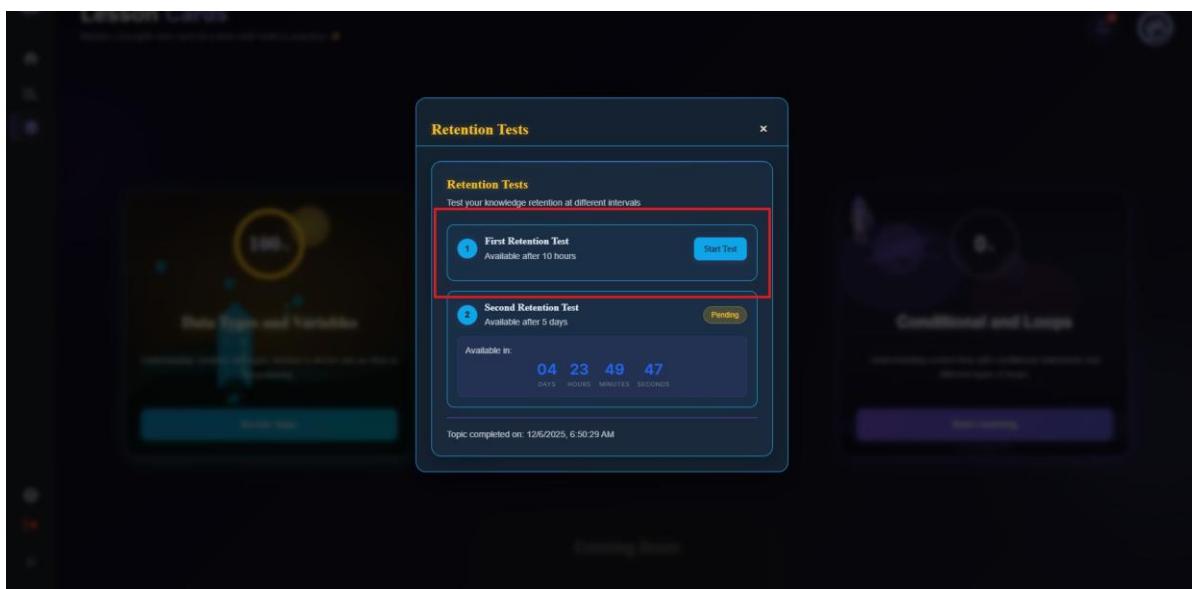
- Stage 1 availability status
- Stage 2 availability status
- Countdown timers (if not yet available)
- “Start Test” buttons (if available)

3. Check Status:

- **Available:** Button says “Start Test”
- **Pending:** Shows countdown timer
- **Completed:** Shows “View Results” or “Completed” badge

Step 5: Start Stage 1 Test (10 Hours)

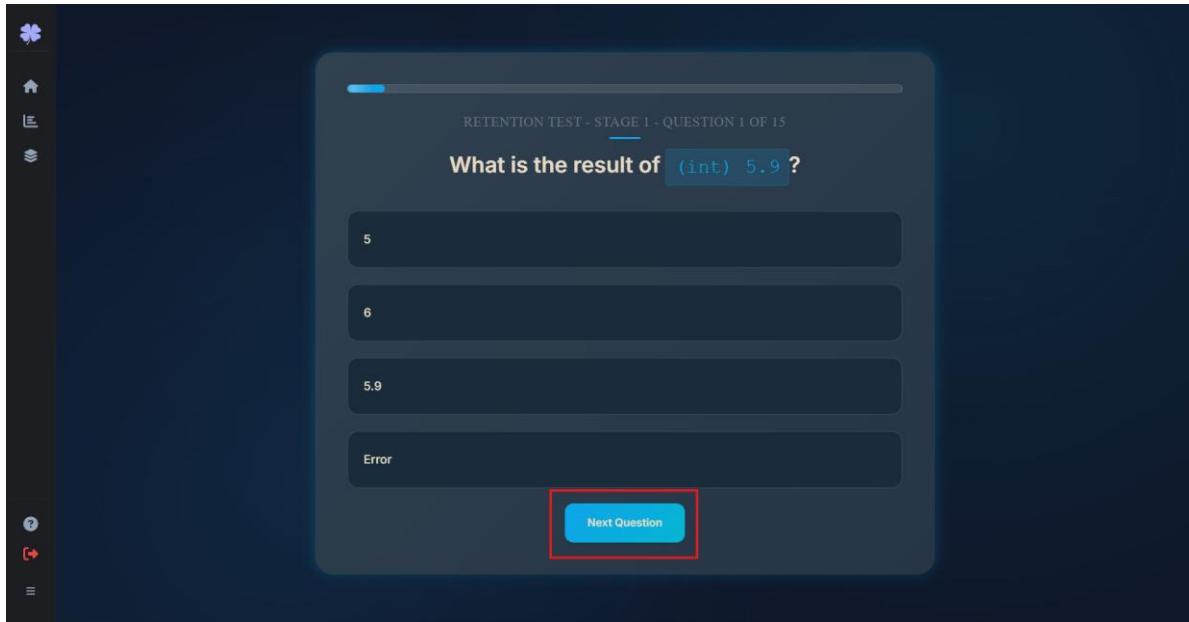
- When Available:** After 10 hours, Stage 1 becomes available
- Click “Start Test”:** Click the “Start Test” button for Stage 1
- Read Instructions:** Review any instructions shown
- Begin:** Click “Begin” or “Start Retention Test” to proceed



Click “Start Test”

Step 6: Answer Questions

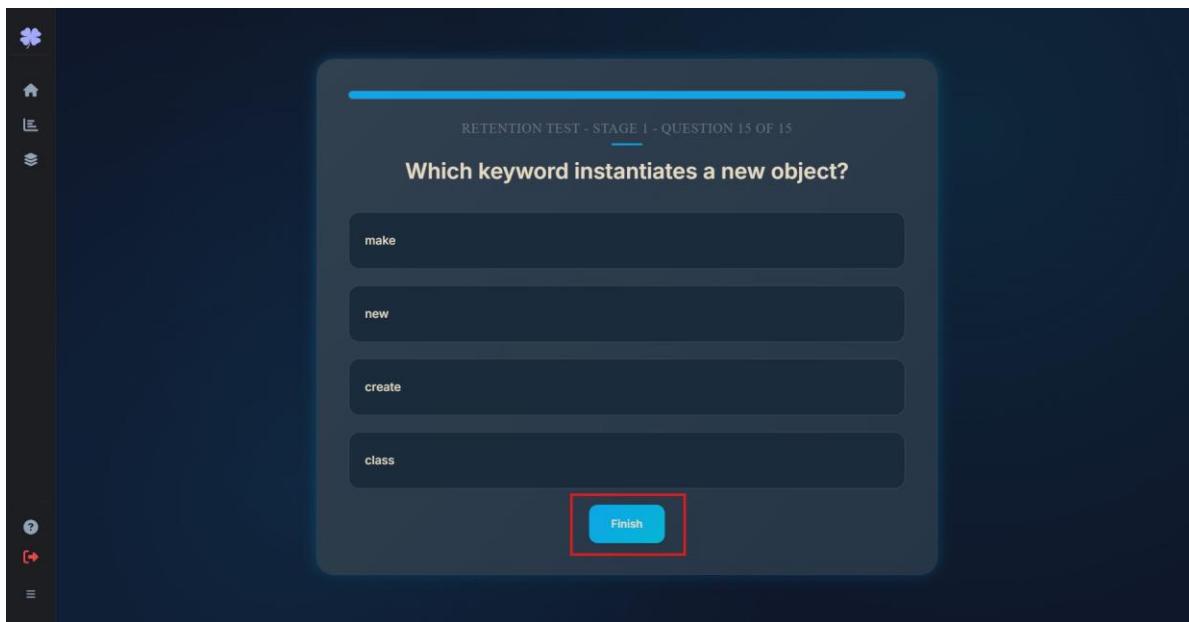
- Read Questions:** Read each question carefully
- Recall Learning:** Remember what you learned from the topic
- Select Answers:** Choose your answers (multiple choice format)
- Navigate:** Use “Next Question” button to move to the next question
- Progress:** Check question counter (e.g., “Question 5 of 15”)
- Answer All:** Make sure to answer all 15 questions



Click "Next Question"

Step 7: Submit Stage 1 Test

1. Once finished with all answers, click “Finish” button
2. **Wait for Results:** Results will load automatically



Click "Finish"

Step 8: View Stage 1 Results

1. **Score Display:** See your Stage 1 score (e.g., “80%”)
2. **Correct/Total:** View “12 out of 15 correct”
3. **Subtopic Breakdown:** See performance by subtopic

4. **Question Review:** Review each question and your answers
5. **Explanations:** Read explanations for each question
6. **Complete:** Click “Finish & Return to My Deck” button

The screenshot shows the 'Retention Test Results' page. At the top, it says 'Here's how you performed on the retention test' with a score of '3/15'. Below this is a 'Keep practicing!' button. A difficulty scale from 'Easy' to 'Hard' is shown. Three sections are listed: 'Declaring Variables' (1/5), 'Primitive Data Types' (2/5), and 'Non-primitive Data Types' (0/5). Each section has a progress bar. At the bottom is a 'Finish & Return to My Deck' button, which is highlighted with a red box.

Retention Test Result

Step 9: Wait for Stage 2 (5 Days)

1. **Wait Period:** Wait **5 days (120 hours)** after topic completion
2. **Countdown:** Check the countdown timer on the topic card
3. **Notification:** You may get notified when Stage 2 becomes available
4. **Continue Learning:** Work on other topics while waiting

The screenshot shows the 'Retention Tests' status view. It displays '1 of 2 retention tests completed' and lists two tests: 'First Retention Test' (Completed) and 'Second Retention Test' (Pending). The second test has a countdown timer showing 'Available in: 04 23 40 47' (Days Hours Minutes Seconds). The background shows a blurred lesson card for 'Data Structures and Variables'.

Retention Tests Status View

Step 10: Take Stage 2 Test (5 Days)

1. **After 5 Days:** Stage 2 becomes available
2. **Repeat Steps 5-8:** Follow the same process as Stage 1
3. **Answer Questions:** Complete all 15 questions
4. **Submit:** Submit your answers
5. **View Results:** See your Stage 2 score

Step 11: Compare Stage 1 vs Stage 2

1. **View Comparison:** See both Stage 1 and Stage 2 scores together
 2. **Analyze Retention:**
 - **High Scores Both:** Excellent retention!
 - **Stage 1 High, Stage 2 Lower:** Some forgetting occurred
 - **Both Low:** May need to review the topic
 3. **Subtopic Analysis:** See which areas retained best
 4. **Learning Insights:** Understand your long-term retention patterns
-

Your Learning Journey

Scenario: You're starting a new topic from scratch. Follow this complete workflow:

Phase 1: Getting Started

Step 1: Select Your Topic

1. Log in to CLOVE
2. Navigate to "My Deck"
3. Browse available topic cards
4. Click on the topic you want to start (e.g., "Topic 1: Data Types and Variables")
5. Read the topic introduction

Step 2: Take Pre-Assessment

1. Click "Pre-Assessment"
2. Answer all questions honestly (15 questions)
3. Submit the assessment
4. View your baseline score (e.g., "60%")

5. Review subtopic breakdown
 6. **Result:** First subtopic are now unlocked!
-

Phase 2: Learning Subtopics

For Each Subtopic (Repeat for all subtopics):

Step 3: Start a Subtopic

1. Click on **Subtopic 1** from the subtopic list
2. The subtopic page loads with tabs: Lesson, Practice, Challenges

Step 4: Complete the Lesson

1. Click the “**Lesson**” tab
2. Read the lesson introduction
3. Study each section carefully:
 - Read explanations
 - Review code examples
 - Understand concepts
4. Take notes (optional but recommended)
5. Mark lesson as complete

Step 5: Complete Practice Exercises

1. Click the “**Practice**” tab
2. Start with Practice Exercise 1
3. Read the question carefully
4. Select your answer
5. Submit and review feedback
6. Read explanations (especially for incorrect answers)
7. Repeat for all practice exercises
8. Retry exercises if needed

Step 6: Attempt Challenges (Optional but Recommended)

1. Click the “**Challenges**” tab
2. Select a challenge (Code Fixer, Code Completion, or Output Tracing)
3. Read the challenge scenario

4. Solve the challenge
5. Submit your solution
6. Review feedback and learn from mistakes
7. Attempt more challenges if desired

Step 7: Mark Subtopic Complete

1. After completing lesson and practice, the subtopic is marked complete
 2. Move to the next subtopic
 3. Repeat Steps 3-7 for all subtopics
-

Phase 3: Final Assessment

Step 8: Complete Post-Assessment

1. After completing **all subtopics**, return to the topic page
 2. Click "**Post-Assessment**"
 3. Answer all questions (same questions as pre-assessment)
 4. Submit the assessment
 5. View your post-assessment score (e.g., "85%")
 6. **Compare with Pre-Assessment:** See your improvement!
- Pre: 60% * Post: 85% □ 25% improvement!
 7. Review subtopic breakdown
 8. **Topic is now Complete!**
-

Phase 4: Retention Testing

Step 9: Wait for Stage 1 Retention Test (10 Hours)

1. After completing the topic, wait **10 hours**
2. Check the topic card in "My Deck"
3. Look for the retention test icon ()
4. Click the icon to see countdown timer
5. When timer reaches zero, Stage 1 becomes available

Step 10: Take Stage 1 Retention Test

1. Click "**Start Retention Test**" for Stage 1
2. Answer all 15 questions
3. Submit the test

4. View results (e.g., "80% - 12 out of 15 correct")
5. Review question-by-question breakdown

Step 11: Wait for Stage 2 Retention Test (5 Days)

1. Wait **5 days (120 hours)** from topic completion
2. Check countdown timer on topic card
3. When available, Stage 2 button will appear

Step 12: Take Stage 2 Retention Test

1. Click "**Start Retention Test**" for Stage 2
 2. Answer all 15 questions
 3. Submit the test
 4. View results
 5. **Compare Stage 1 vs Stage 2:** See long-term retention
-

Tips for Success

- Consistency:** Study regularly (daily if possible)
- Don't Rush:** Take time to understand concepts
- Practice:** Complete all practice exercises
- Review:** Review incorrect answers
- Challenges:** Attempt challenges to test understanding
- Take Notes:** Write down important concepts
- Ask Questions:** Re-read lessons if confused

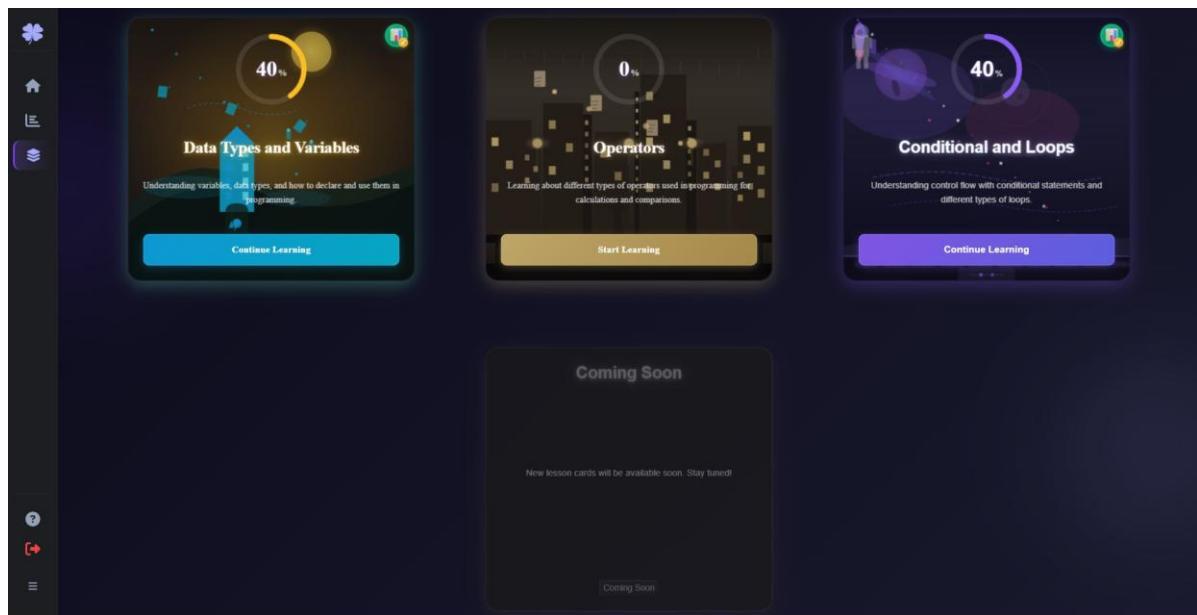
What You'll Achieve

- **Baseline Knowledge:** Pre-assessment shows starting point
- **Learning Progress:** Post-assessment shows improvement
- **Short-term Retention:** Stage 1 shows 10-hour retention
- **Long-term Retention:** Stage 2 shows 5-day retention
- **Complete Understanding:** Mastery of all subtopics
- **Topic Completion:** Full topic completion badge

Core Features

My Deck

Location: Main navigation → “My Deck”



Features:

- **Topic Cards:** Visual cards for each available topic
- **Progress Indicators:** See completion percentage at a glance
- **Retention Test Icon:** appears after completing retention tests
- **Quick Access:** Click any card to start or continue learning



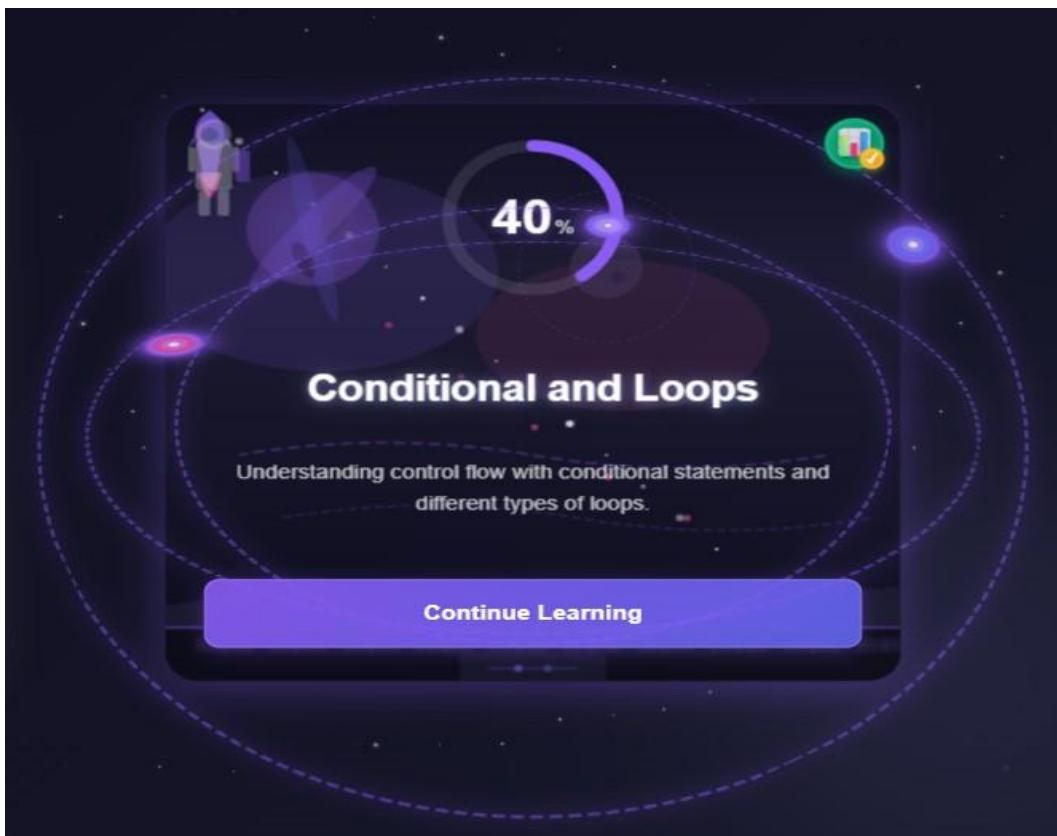
Card for the “Conditional and Loops” topic

How to Use:

1. Navigate to “My Deck” from the main menu
2. Browse available topics
3. Click on a topic card to open it
4. View topic introduction (first time only)
5. Access assessments and subtopics

Topic Card Information:

- Topic name and description
- Progress percentage (circular progress ring)
- Completion status
- Retention test availability (if applicable)
- Theme-based visual design



Hover effect animation of the Card "Conditional and Loops" space themed

Dashboard

Location: Main navigation → “Dashboard”

Features:

- **Overview:** Quick snapshot of your learning progress
- **Recent Activity:** See what you've done recently
- **Quick Actions:** Fast access to continue learning
- **Statistics:** Key metrics at a glance
- **Achievements:** Recent achievements and badges

Hello, ExpertGuru007!
Here's your learning journey progress 🚀

WANDERER 0 pts

0 / 500 XP

Bronze → Silver (501 pts to Explorer)

Continue Learning
Pick up where you left off
LAST STUDIED

Data Types and Variables
Keep the momentum going! 💪

Resume Topic →

Progress Overview

Your Streak
1 Day Streak
Keep logging in to build your streak!
THIS WEEK

3 DAYS WEEK MONTH

Challenges Solved

Recent Topic and Log In Streak

Progress Overview

- Data Types and Variables: NOT STARTED
- Operators: NOT STARTED
- Conditional Loops: NOT STARTED

Challenges Solved

0% SOLVED

You've solved 0 out of 405 challenges. Keep up answering the challenges!

Achievements

Category	Achievement	Description	Progress
Centurion	Getting Warmed Up	Maintain a 3-day streak	1/3
	Week Warrior	Maintain a 7-day streak	1/7
	Unstoppable	Maintain a 30-day streak	1/30
	Challenge Starter	Solve 10 challenges	0/10
	Challenge Master	Solve 50 challenges	0/50
First Steps	Centurion	Solve 100 challenges	0/100
	First Steps	Complete your first topic	0/1
	Knowledge Seeker	Complete 2 topics	0/2
	Master of All	Complete all 3 topics	0/3
	Sharp Shooter	Achieve 75% average accuracy	0/75

Progress Overview and Challenges Solved

Achievements

Category	Achievement	Description	Progress
Centurion	Getting Warmed Up	Maintain a 3-day streak	1/3
	Week Warrior	Maintain a 7-day streak	1/7
	Unstoppable	Maintain a 30-day streak	1/30
	Challenge Starter	Solve 10 challenges	0/10
	Challenge Master	Solve 50 challenges	0/50
First Steps	Centurion	Solve 100 challenges	0/100
	First Steps	Complete your first topic	0/1
	Knowledge Seeker	Complete 2 topics	0/2
	Master of All	Complete all 3 topics	0/3
	Sharp Shooter	Achieve 75% average accuracy	0/75

Completed Topics

You haven't completed any topics yet. Keep learning and your completed topics will appear here!

Achievements and Completed Topics

Dashboard Sections:

Section	What It Shows
Progress Summary	Overall completion percentage
Recent Topics	Topics you've worked on recently
Challenges Completed	Number of challenges solved
Learning Streak	Consecutive days of learning
Points Earned	Total points accumulated
Quick Links	Fast access to My Deck, Challenges, Progress

Progress Page

Location: Main navigation → “Progress”

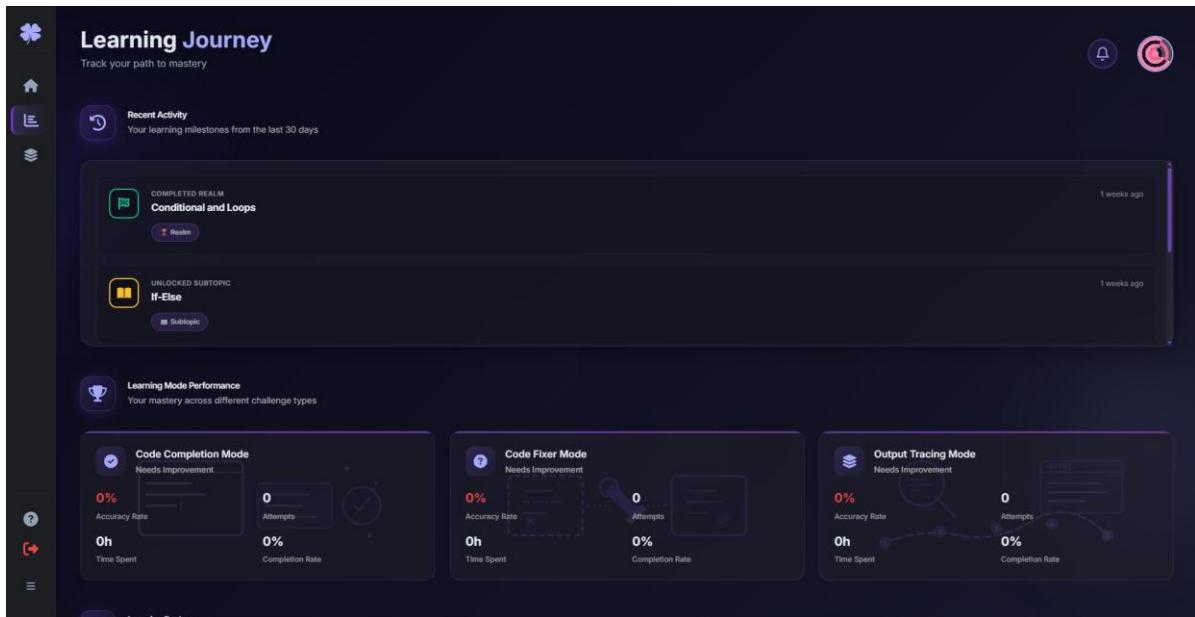
Features:

Overall Progress

- Visual charts showing completion rates
- Progress bars for each topic
- Percentage indicators

Performance Metrics

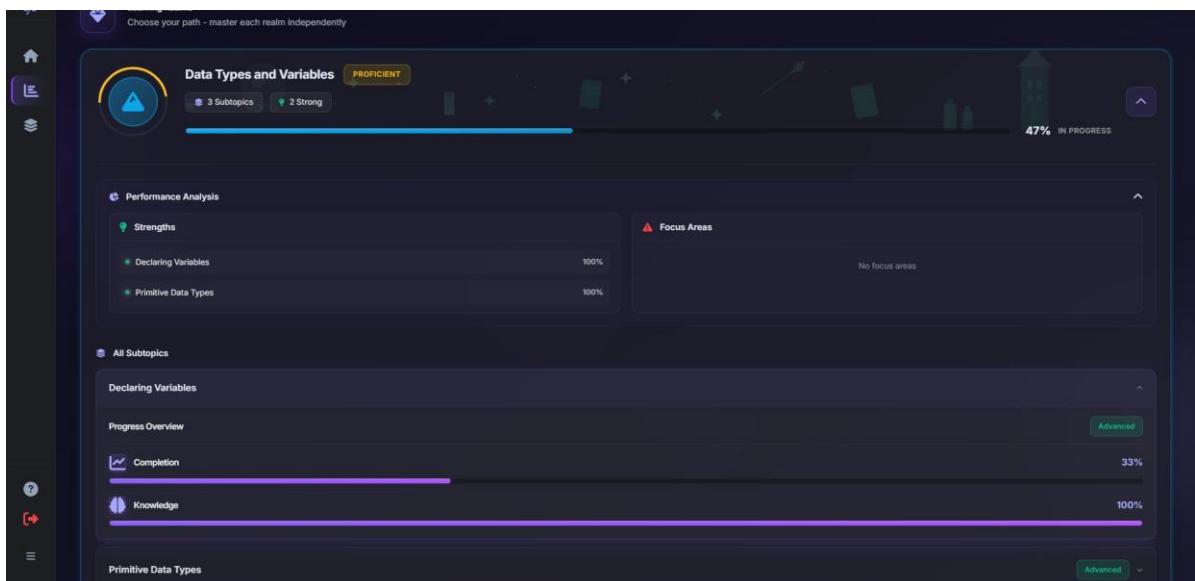
- **Challenge Statistics:**
 - Total challenges completed
 - Success rate percentage
 - Average score
 - Points earned
- **Time Spent:** Total learning time tracked
- **Mastery Levels:** Knowledge level for each subtopic (0-100%)



Recent Activity and data analytics of three challenge types

Topic Breakdown

- Detailed progress for each topic
- Subtopic-level progress
- Completion status for each component



Topic and subtopic progress report

Learning Analytics

- Challenge completion rates over time
- Assessment score trends (pre vs post)
- Retention test results
- Time to mastery metrics
- Learning efficiency scores
- Improvement trends

How to Use Analytics:

1. Visit the Progress page
 2. Review overall statistics in the summary section
 3. Click on a specific topic to drill down
 4. Compare performance over time using charts
 5. Identify areas for improvement
 6. Set learning goals based on data
-

Challenges Page

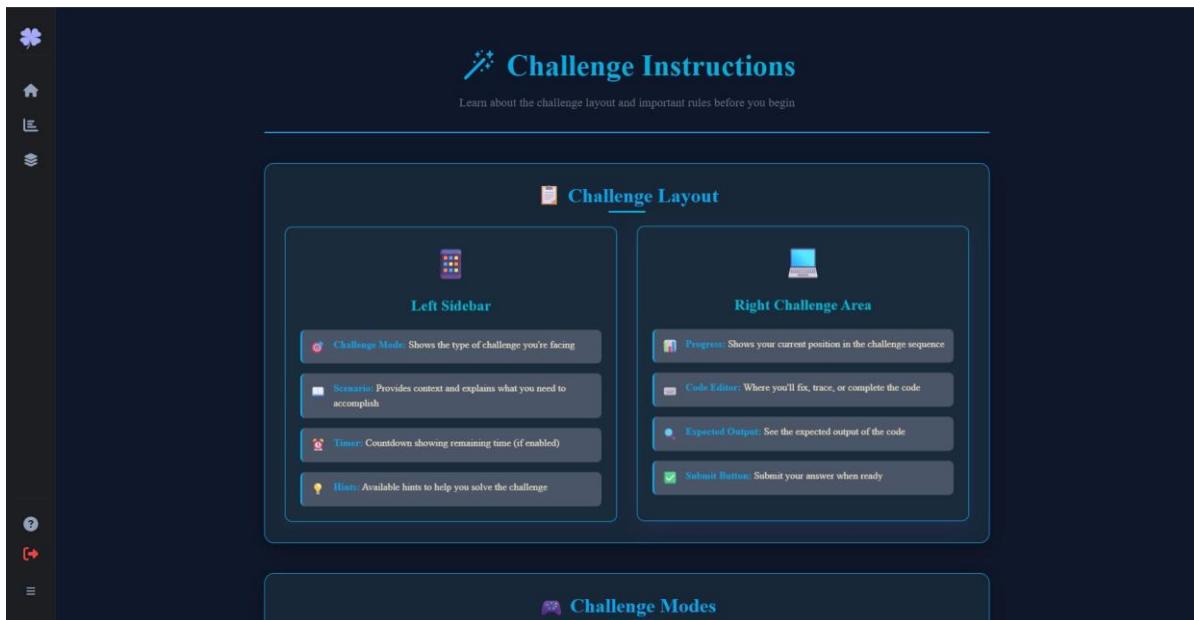
Location: Main navigation → “Challenges”

Features:

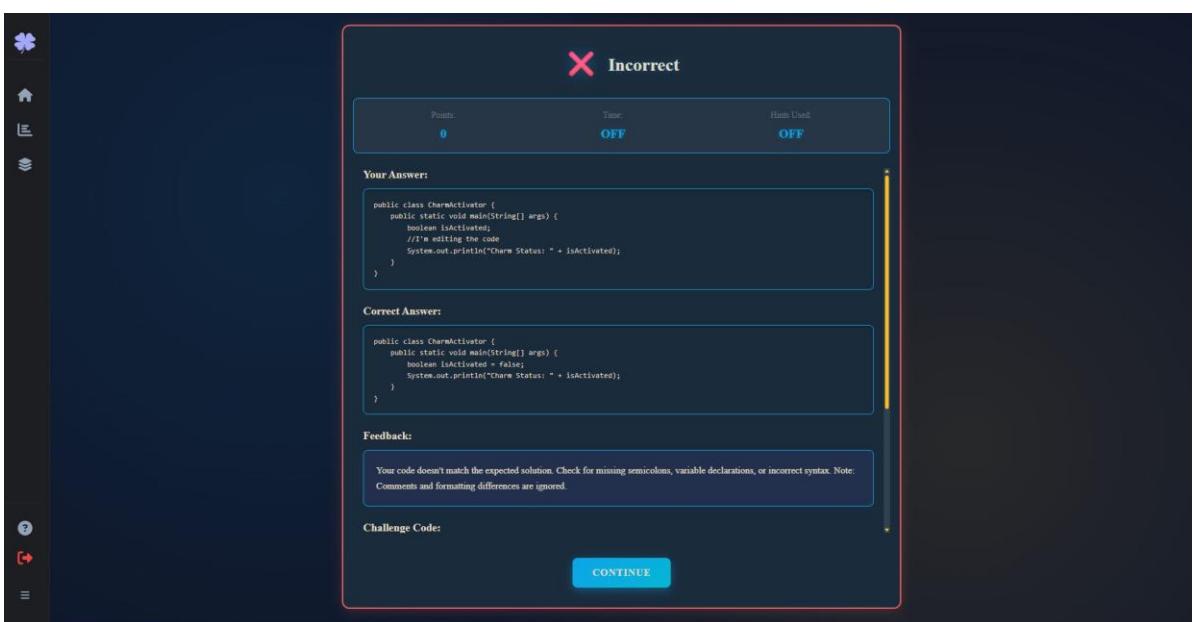
- By difficulty: Easy, Medium, Hard
- By type: Code Fixer, Code Completion, Output Tracing

Step-by-Step Guide:

1. **Browse:** Navigate to the Challenges page
2. **Review:** Read the challenge instructions
3. **Understand:** Review the scenario and story context
4. **Prepare:** Understand what you need to do
5. **Start:** Click “Start Challenge” button
6. **Analyze:** Determine the challenge type presented
7. **Complete:** Use the code editor and output console to solve the challenge
8. **Submit:** Submit your answer when ready
9. **Review:** View results and feedback



Challenge instructions



Challenge feedback and insights

💡 Take Note: Time failed and cancelled challenge are counted as wrong regardless of the answer.

Assessments & Retention Tests

Pre-Assessments

Purpose: Evaluate your baseline knowledge before starting a topic

Key Information:

- **Format:** Multiple-choice questions
- **Timing:** No time limit (take your time)
- **Feedback:** Immediate results after submission
- **Scoring:** Score and percentage displayed
- **Unlocking:** Completing unlocks the first subtopic

Taking a Pre-Assessment:

1. **Navigate:** Go to a topic page
2. **Locate:** Find the Topic you wanted to start
3. See: Pre-assessment icon (the icon which is highlighted and glowing)
4. **Click:** Begin the assessment
5. **Read:** Read each question carefully
6. **Answer:** and Submit: Choose your desired answer
7. **Done:** Complete the 15 questions
8. **Results:** View your score and feedback

Understanding Results:

- **Score:** Your percentage score
- **Breakdown:** Performance by subtopic area
- **Recommendations:** Areas to focus on during learning

See the breakdown analysis of your answers

💡 Tip: Answer honestly. The assessment helps personalize your learning experience.

Post-Assessments

Purpose: Measure learning improvement after completing all subtopics

Features:

- **Same Questions:** Uses the same questions as pre-assessment
- **Comparison:** You can compare scores to see improvement
- **Detailed Breakdown:** Performance by subtopic
- **Completion Required:** Must complete to finish the topic

Taking a Post-Assessment:

1. **Complete:** Finish all three subtopics in a topic
2. **Locate:** Find the “Start Post-Assessment” button
3. **Start:** Begin the assessment
4. **Answer:** Answer all questions
5. **Submit:** Submit when complete
6. **Review:** See detailed breakdown

Understanding Improvement:

- **Score Comparison:** Pre vs Post scores side-by-side

- **Improvement Percentage:** How much you've improved
 - **Subtopic Analysis:** See improvement in each area
 - **Areas Mastered:** Topics you've fully understood
-

Retention Tests

Purpose: Test long-term knowledge retention at specific intervals

Retention Test Schedule:

Stage	Availability	Questions	Purpose
Stage 1	10 hours after topic completion	15 questions	Short-term retention
Stage 2	5 days (120 hours) after topic completion	15 questions	Long-term retention

Features:

- **Countdown Timer:** Shows when test becomes available
- **Two Stages:** Test retention at different time intervals
- **Results Tracking:** View performance over time
- **Detailed Feedback:** Question-by-question breakdown

Accessing Retention Tests:

Step-by-Step:

1. **Complete Topic:** Finish pre + post assessments + all subtopics
2. **Wait:** Wait for the availability window
3. **Look for Icon:** Find the retention test icon () on topic card
4. **Click:** Click the icon to view availability
5. **Check Status:** See countdown timer if not yet available
6. **Start:** Click "Start Test" when available
7. **Complete:** Answer all 15 questions
8. **Submit:** Submit your answers
9. **Review:** View results and explanations

Viewing Results:

- **Access:** Click retention test icon on completed topics
- **Scores:** View scores for both stages

- **Breakdown:** See detailed question-by-question results
- **Tracking:** Monitor retention over time
- **Comparison:** Compare Stage 1 vs Stage 2 performance

What Retention Tests Tell You:

- **Short-term Retention (Stage 1):** How well you remember after 10 hours
- **Long-term Retention (Stage 2):** How well you remember after 5 days
- **Learning Effectiveness:** Whether your study methods are working
- **Areas to Review:** Which topics need more reinforcement

⚠️ Important: Retention tests are only available during specific time windows. Make sure to complete them when available!

Challenges

Challenge Types

CLOVE offers three types of interactive coding challenges, each designed to test different skills:

1. Code Fixer

Objective: Identify and fix bugs in provided code

The screenshot shows the 'CODE FIXER' challenge interface. At the top right, it says 'Challenge 4 of 5'. The main area is titled 'CODE FIXER CHALLENGE'. Below that is a 'EDIT THE CODE:' section containing the following Java code:

```

1 public class DustCollector {
2     public static void main(String[] args) {
3         int crystalDust = 3000000000;
4         System.out.println("Crystal Dust: " + crystalDust);
5         //editing
6     }
7 }
```

Below the code editor is a 'SUBMIT' button. To the left of the code editor is a sidebar with the following sections:

- CODE FIXER**: Instructions to identify and fix all errors in the code to restore correct functionality.
- SCENARIO:** A `DustCollector` tracks `crystalDust`. The literal value assigned to `crystalDust` is too large for an `int` variable, leading to a compilation error. Fix the code to correctly store this large integer value.
- HINTS :** Buttons for 'Get Hint' and two hints:
 - HINT 1:** Hint 1: An 'int' has a defined maximum positive value (approx. 2.1 billion).
 - HINT 2:** Hint 2: If a whole number literal exceeds this 'int' limit, it's considered a different type.
- Back to Practice**

Code Fixer

How to Play:

1. **Read:** Read the scenario and problem description
2. **Analyze:** Analyze the provided code carefully
3. **Identify:** Identify the bug(s) in the code
4. **Fix:** Fix the code in the Monaco code editor
5. **Test:** Mentally test your solution
6. **Submit:** Submit your solution
7. **Feedback:** Receive immediate feedback

Tips for Success:

- Read error messages carefully if provided
- Test your logic mentally before submitting
- Check for syntax errors (missing semicolons, brackets)
- Look for logic errors (wrong operators, incorrect conditions)
- Consider edge cases
- Review variable names and types

Common Bug Types:

- Syntax errors (missing punctuation)
- Logic errors (incorrect conditions)
- Type mismatches
- Off-by-one errors
- Null pointer exceptions

2. Code Completion

Objective: Complete missing code segments

The screenshot shows a challenge interface titled "Challenge 1 of 5". The main area is labeled "CODE COMPLETION CHALLENGE". It features a Monaco code editor with the following code:

```
public class ScrollEvaluation {  
    public static void main(String[] args) {  
        String scrollCondition = [ 2 ];  
        boolean isPristine;  
  
        if (scrollCondition == '0') {  
            isPristine = true;  
        } else {  
            isPristine = false;  
        }  
        System.out.println("Scroll Condition: " + scrollCondition);  
        System.out.println("Is Pristine: " + isPristine);  
    }  
}
```

To the left of the code editor is a sidebar with instructions and a scenario:

- CODE COMPLETION**: Fill in the missing code blocks to complete the program logic as described in the scenario.
- Drag and drop the code choices into the marked positions
- Click on filled choices to remove them if needed
- Ensure the code produces the expected output
- Check for proper syntax and logic flow

SCENARIO:
A `ScrollEvaluator` checks the condition of a magical scroll. The `scrollCondition` variable needs to be declared and initialized to hold a character representing its status, and `isPristine` needs to be declared as a boolean.

At the bottom of the sidebar is a "Back to Practice" button.

Below the code editor is an "Expected Output" section showing:
Scroll Condition: 0
Is Pristine: true

A large blue "SUBMIT" button is at the bottom right.

Code Completion

How to Play:

1. **Understand:** Understand the problem requirements
2. **Identify:** Identify what code is missing
3. **Analyze:** Analyze the existing code structure
4. **Write:** Write the missing code
5. **Integrate:** Ensure it integrates with existing code
6. **Submit:** Submit your solution

Tips for Success:

- Pay attention to variable names and types
- Match the coding style of existing code
- Consider edge cases and boundary conditions
- Ensure proper indentation and formatting
- Check return types and method signatures
- Verify your code compiles

What to Look For:

- Method signatures
- Variable declarations
- Loop structures
- Conditional statements
- Return statements

3. Output Tracing

Objective: Predict program output and trace execution

OUTPUT TRACING

Analyze the code and predict the output that will be produced when it runs.

- Analyze the code and predict the output
- Select the correct output from the choices below
- Consider the order and context of program outputs
- Trace through the code step by step

SCENARIO:

An `OrbIdentifier` system assigns a unique ID to each magical orb. The `orbID` variable holds a single character. Your task is to trace the code and determine the full Orb ID that is displayed.

HINTS :

Get Hint

HINT 1:
Hint 1: Identify the data type used for `'orbID'`.

HINT 2:
Hint 2: Recall how characters are typically printed in Java.

Challenge 4 of 5

OUTPUT TRACING CHALLENGE

CODE TO ANALYZE:

```
1 public class OrbIdentifier {
2     public static void main(String[] args) {
3         char orbID = 'A';
4         System.out.println("Orb ID: " + orbID);
5     }
6 }
```

What is the output of the following code?

Selected: 1 option(s)

A **Orb ID: A** Orb ID: orbID Error

SUBMIT

Output Tracing

How to Play:

1. **Read:** Read the code carefully from top to bottom
2. **Trace:** Trace through the execution step by step
3. **Track Variables:** Track variable values as they change
4. **Follow Loops:** Follow loop iterations carefully
5. **Predict:** Predict the final output
6. **Select:** Select the correct answer from options
7. **Submit:** Submit your prediction

Tips for Success:

- Follow variable values through execution
- Track loop iterations (count carefully!)
- Consider operator precedence
- Watch for type conversions
- Track method calls and returns
- Be careful with array indices

Common Pitfalls:

- Off-by-one errors in loops
- Forgetting operator precedence
- Missing type conversions
- Incorrect array indexing
- Not tracking variable updates

Challenge Features

Adaptive Difficulty

The system automatically adjusts challenge difficulty based on your performance:

How It Works:

- **Beginners:** Start with easier challenges
- **Improving:** Difficulty increases as you improve
- **Struggling:** System may provide easier challenges
- **Mastery:** Advanced challenges for experienced learners

Benefits:

- Always appropriately challenged
- Prevents frustration from too-difficult challenges
- Prevents boredom from too-easy challenges
- Optimal learning pace

Hints

Contextual help that activates when you're struggling:

Features:

- **Activation:** Appears when you're keep getting incorrect
- **Contextual:** Hints are specific to your current challenge
- **Strategic Use:** Use hints wisely (may affect scoring)

The screenshot shows the Clove Hints System interface. On the left, there's a vertical sidebar with icons for home, challenges, and other navigation. The main area has a dark background with light-colored boxes for content.

OUTPUT TRACING

Analyze the code and predict the output that will be produced when it runs.

- Analyze the code and predict the output
- Select the correct output from the choices below
- Consider the order and content of program outputs
- Trace through the code step by step

SCENARIO:

An `OrbIdentifier` system assigns a unique ID to each magical orb. The `orbID` variable holds a single character. Your task is to trace the code and determine the full Orb ID that is displayed.

HINTS

No Hints Left

HINT 1:
Hint 1: Identify the data type used for 'orbID'.

HINT 2:
Hint 2: Recall how characters are typically printed in Java.

HINT 3:
Hint 3: The '+' symbol combines a literal string with the value of the variable.

CODE TO ANALYZE:

```
1 public class OrbIdentifier {  
2     public static void main(String[] args) {  
3         char orbID = 'A';  
4         System.out.println("Orb ID: " + orbID)  
5     }  
6 }
```

What is the output of the following code?

Selected: 1 option(s)

A **Orb ID: A** **Orb ID: orbID** **Error**

[← Back to Practice](#)

Hints System

💡 Tip: Try to solve challenges without hints first. Use hints as a learning tool, not a crutch.

Timer

Optional time limits for added challenge:

Features:

- **Adaptive Timing:** Time limits adjust based on difficulty
- **Streak Activation:** Timer activates during performance streaks
- **Optional:** Can be disabled in some cases
- **Visual Indicator:** See time remaining

The screenshot shows the CLOVE user interface with a dark theme. On the left, there's a vertical toolbar with icons for clover, home, list, and settings. The main area has two panels. The left panel contains a "CODE COMPLETION" section with instructions to fill missing code blocks, a "SCENARIO" section describing a SentinelSystem class, and an "ARCANE TIMER" section showing 09:31 with a progress bar at 95%. The right panel shows a "CODE COMPLETION" title, a code editor with numbered lines, and an "Expected Output" section with sample values.

CODE COMPLETION

Fill in the missing code blocks to complete the program logic as described in the scenario.

- Drag and drop the code choices into the marked positions
- Click on filled choices to remove them if needed
- Ensure the code produces the expected output
- Check for proper syntax and logic flow

SCENARIO:

A `SentinelSystem` manages `sentinelEnergy`, `shieldStatus`, and `dangerProtocol`. `sentinelEnergy` is an integer. `shieldStatus` and `dangerProtocol` are set conditionally in an `if-else` structure. Complete the declaration of `sentinelEnergy` and the assignments for `shieldStatus` and `dangerProtocol` in the `else` block.

ARCANE TIMER

09:31

95%

← Back to Practice

CODE COMPLETION

COMPLETE THE CODE:

```
1 public class SentinelSystem
2     public static void main
3         [ 1 ] sentinelEnergy
4         String shieldStatus
5         boolean dangerProtocol
6
7         if (sentinelEnergy
8             shieldStatus =
9             dangerProtocol
10        } else {
11            [ 2 ] = "Shielded"
12            [ 3 ] = true;
13        }
14    }
```

Expected Output:

Energy: 120
Status: Shields Online.
Protocol: false

Timer System

Timer Benefits:

- Adds challenge and excitement

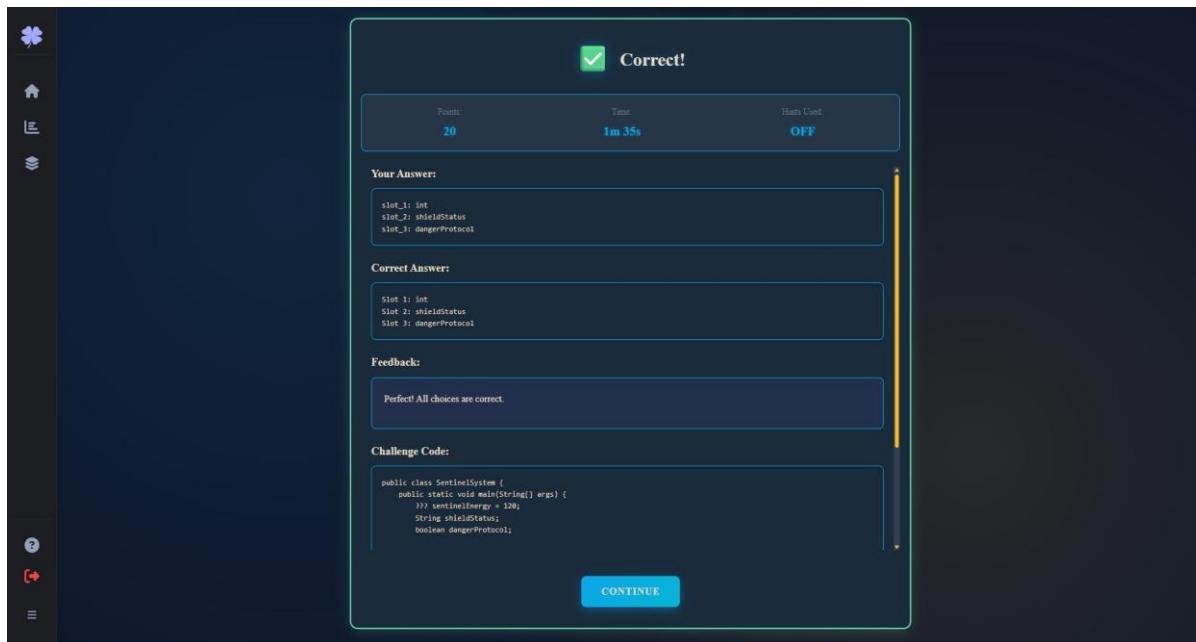
- Simulates real-world coding scenarios
- Improves time management skills
- Builds pressure-handling ability

Points & Scoring

Earn points and track your performance:

Scoring Factors:

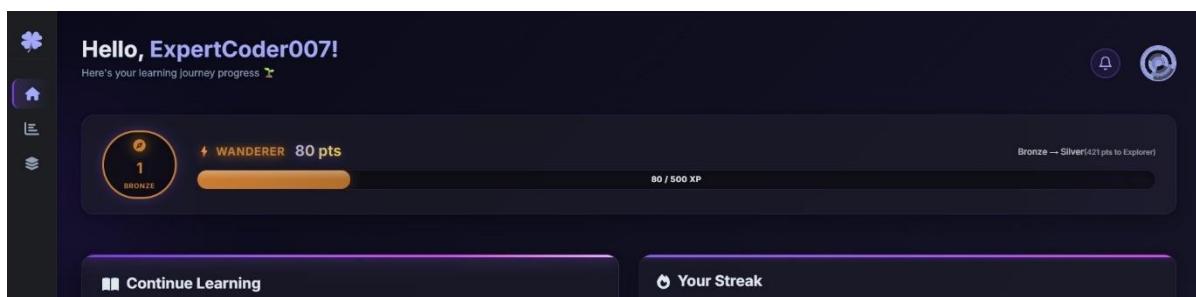
- **Correctness:** Primary factor in scoring
- **Difficulty:** Higher difficulty = more points
- **Speed:** Bonus points for quick completion
- **Hints Used:** Points may be deducted for hints



Points System at the upper corner

Points Display:

- See points earned after each challenge
- Track total points in your profile

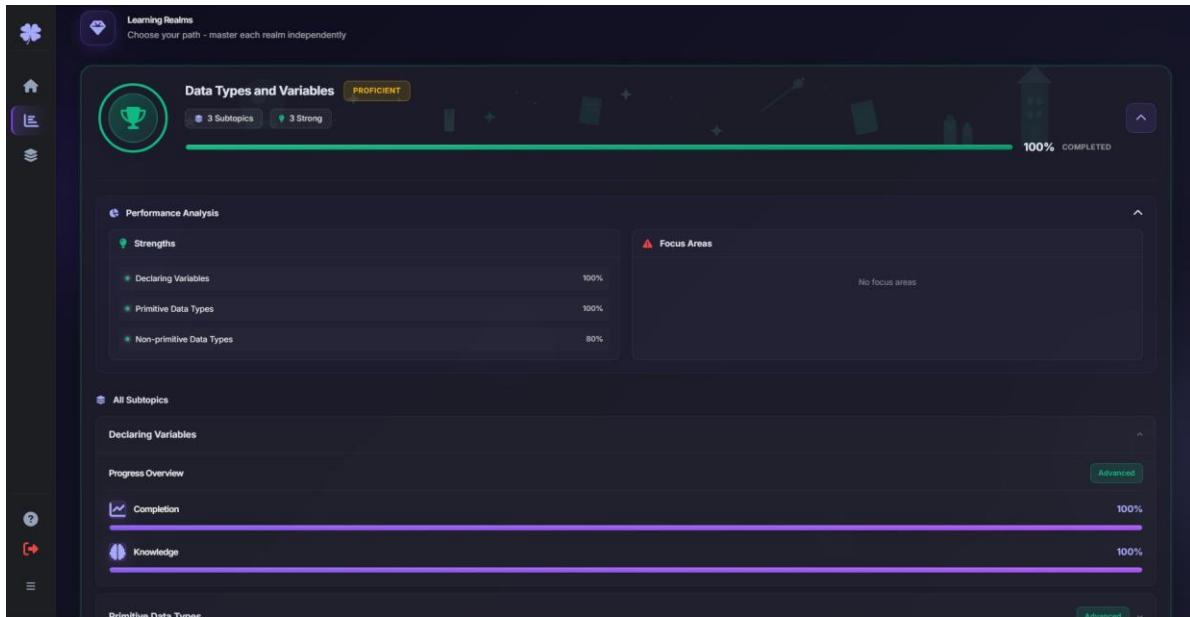


Points earned by you

Progress Tracking

Progress Dashboard

Location: Progress tab in main navigation



Topics and Subtopics progress

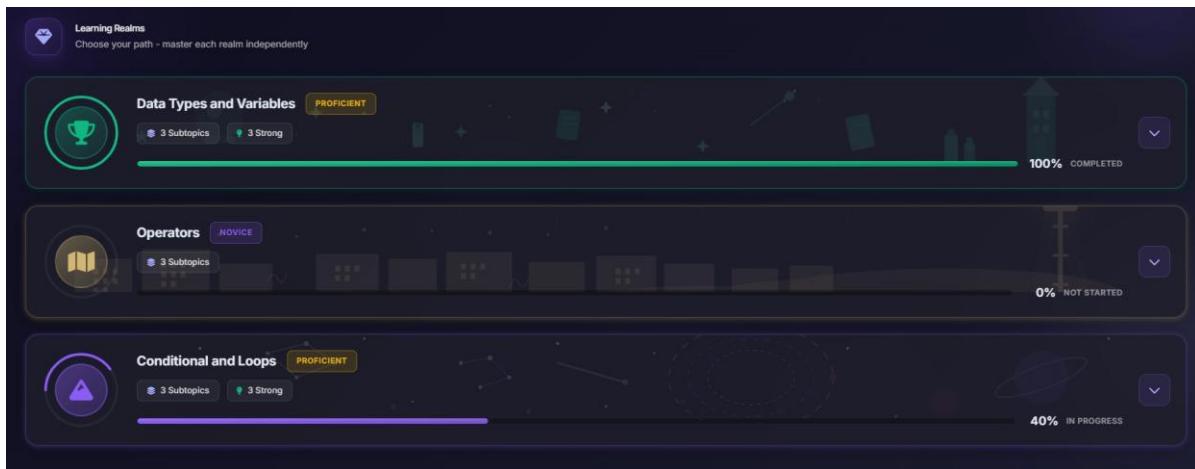
What You Can See:

Overall Completion

- **Percentage:** Overall completion percentage across all topics
- **Visual Chart:** Pie chart or progress bar
- **Topics Completed:** Number of topics finished
- **Topics In Progress:** Number of topics currently being worked on

Topic Progress

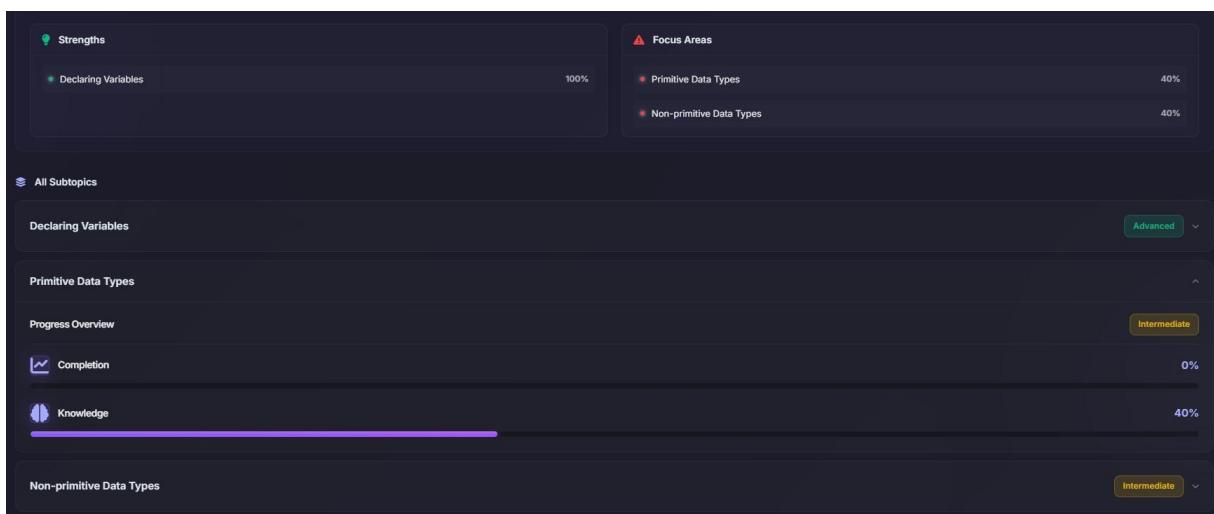
- **Individual Topics:** Progress for each topic separately
- **Progress Bars:** Visual representation of completion
- **Status Indicators:**
 - Trophy Icon: Completed (100%)
 - Mountain Icon: In Progress (1-99%)
 - Map Icon: Not Started (0%)



Topic Progress

Subtopic Breakdown

- **Detailed View:** See progress within each topic
- **Component Status:**
 - Pre-assessment status
 - Each subtopic completion
 - Post-assessment status
 - Retention test completion

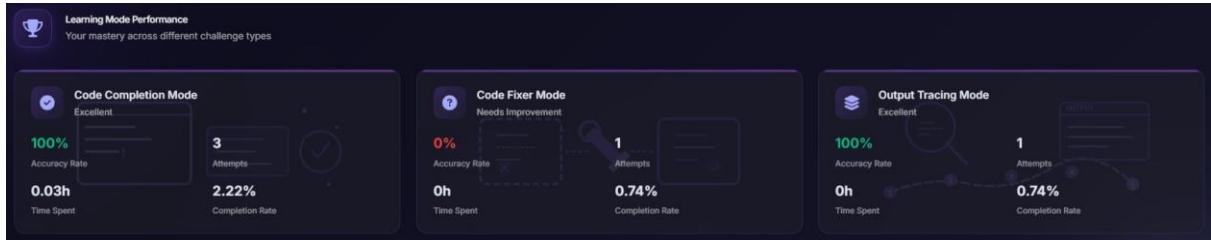


Subtopic Progress and Analytics

Challenge Statistics

- **Total Completed:** Number of challenges solved
- **Success Rate:** Percentage of successful completions

- **Average Score:** Mean score across all challenges
- **By Type:** Statistics for each challenge type
- **By Difficulty:** Statistics by difficulty level



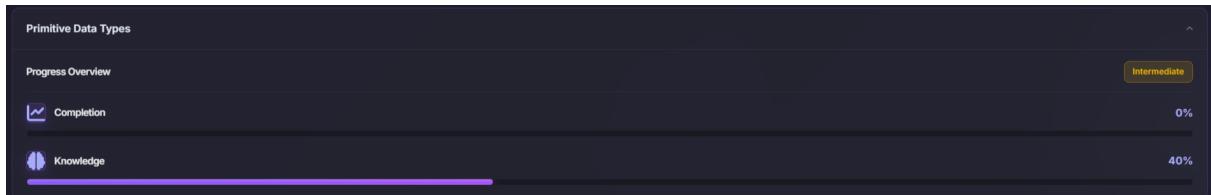
Challenge Type Analytics

Time Spent

- **Total Time:** Cumulative learning time

Mastery Levels

- **Knowledge Level:** 0-100% for each subtopic
- **Mastery Classification:**
 - Beginner: 0-33%
 - Intermediate: 34-66%
 - Advanced: 67-100%



Knowledge Level

Progress Indicators

Charts and Graphs:

- **Line Charts:** Show progress over time
- **Bar Charts:** Compare performance across topics
- **Pie Charts:** Show completion distribution
- **Progress Rings:** Circular progress indicators

Learning Analytics

Metrics Tracked:

Metric	Description	How It Helps
Challenge Completion Rate	% of challenges completed successfully	Shows overall performance
Assessment Scores	Pre vs Post comparison	Measures learning improvement
Retention Test Results	Stage 1 vs Stage 2 scores	Tracks long-term retention

Metric	Description	How It Helps
Time to Mastery	Time taken to reach mastery	Measures learning efficiency
Learning Efficiency	Knowledge gained per hour	Optimizes study time
Improvement Trends	Performance over time	Shows learning trajectory

How to Use Analytics:

1. **Visit Progress Page:** Navigate to the Progress tab
2. **Review Summary:** Check overall statistics
3. **Drill Down:** Click on specific topics for details
4. **Compare:** Compare performance over time
5. **Identify:** Find areas needing improvement
6. **Set Goals:** Use data to set learning goals
7. **Track:** Monitor progress toward goals

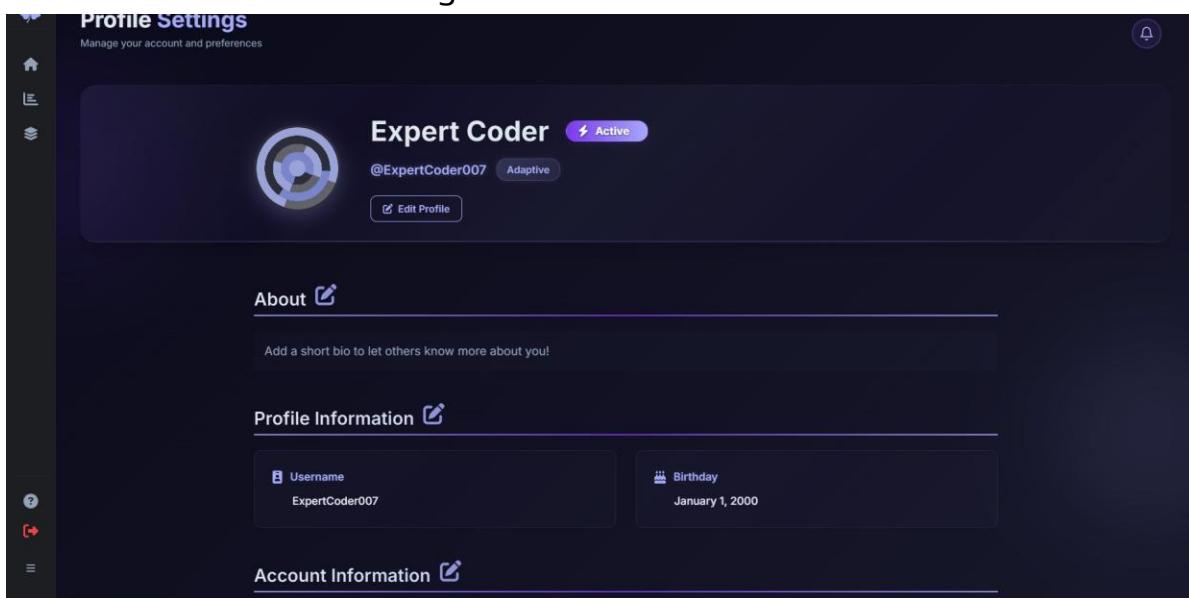
Interpreting Data:

- **Rising Trends:** Good - you're improving
- **Flat Trends:** May need to adjust learning approach
- **Declining Trends:** Review and focus on fundamentals
- **High Variability:** May indicate inconsistent study habits

Profile & Settings

Profile Page

Location: Profile icon in navigation → “Profile”



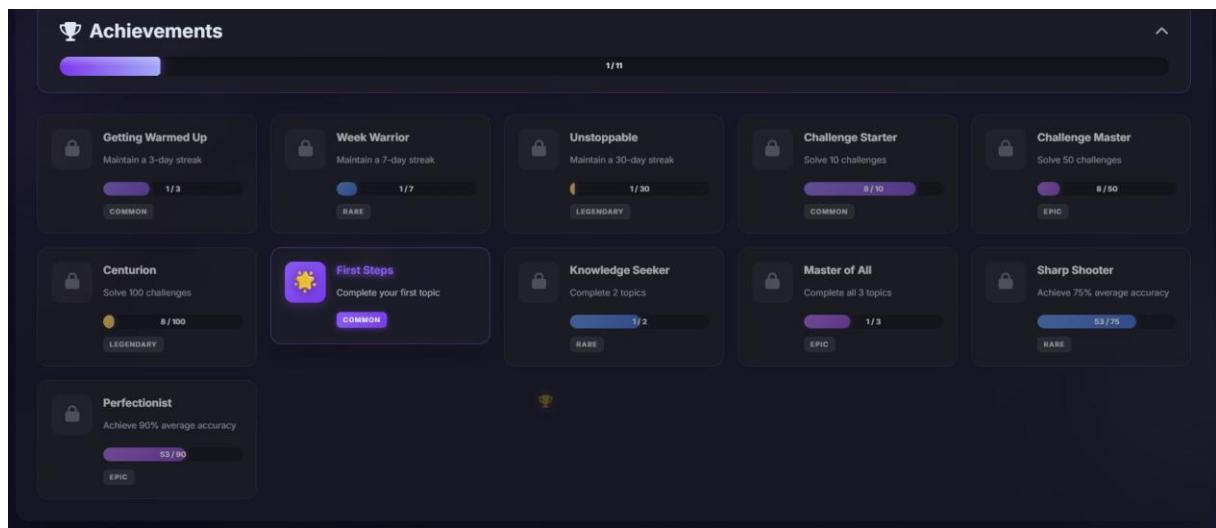
Profile Page

Profile Information

- **Name:** Your first and last name
- **Email:** Your registered email address
- **Profile Photo:** Your uploaded profile picture
- **Bio:** Personal bio (optional)
- **Traveler Class:** Your chosen class from onboarding
- **Current Realm:** Your selected realm

Achievements

- **Badges:** Unlocked achievement badges
- **Milestones:** Reached learning milestones
- **Special Accomplishments:** Notable achievements



Achievements Badge

Account Settings

Personal Information

- **Edit Name:** Update first and last name
- **Change Email:** Update email address (requires verification)
- **Update Birthday:** Modify date of birth
- **Edit Bio:** Add or update personal bio

About 

Add a short bio to let others know more about you!

Profile Information 

 Username
ExpertCoder007

 Birthday
January 1, 2000

Account Information 

 Email
adaptive2@clove.com

 Password

Profile and Account Information

Password Management

- **Change Password:** Update your account password
- **Password Requirements:**
 - Minimum 8 characters
 - Mix of letters, numbers, and special characters
 - Cannot be your email or name

Account Information 

 Email
adaptive2@clove.com

 Password

New Password 

Confirm New Password 

Current Password (required for changes) 

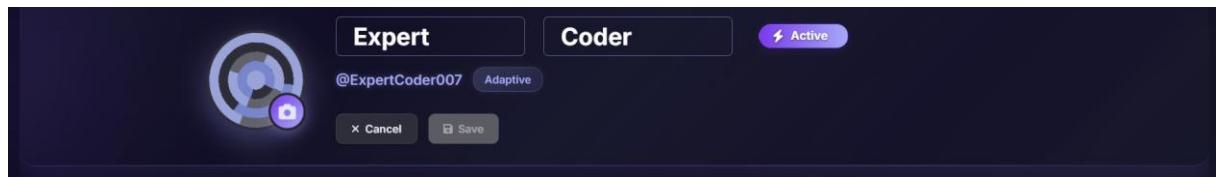
 Cancel  Save

Password Form

Profile Photo

- Click Icon: Camera icon on the lower side of the profile photo

- **Upload Photo:** Add a profile picture
- **Change Photo:** Replace existing photo
- **Remove Photo:** Delete profile picture
- **Supported Formats:** JPG, PNG, GIF
- **Size Limit:** Typically 5MB maximum



Change Profile Picture

User Account Management

- **Delete Account:** Permanently delete your account.

Statistics

What's Tracked:

Learning Statistics:

- **Total Topics Completed:** All finished topics
- **Total Challenges Solved:** All completed challenges
- **Total Points Earned:** Cumulative points
- **Current Learning Streak:** Consecutive days
- **Longest Streak:** Best streak achieved
- **Login Days This Week:** Active days this week
- **Total Login Days:** All-time active days

Performance Statistics:

- **Accuracy Rates:** Overall correctness percentage
- **Average Challenge Score:** Mean challenge performance
- **Assessment Improvement:** Pre vs Post comparison
- **Retention Test Scores:** Stage 1 and Stage 2 results

- **Hours Spent Learning:** Total learning time
- **Completion Rates:** By mode (lessons, practice, challenges)

Achievement Statistics:

- **Badges Earned:** Number of achievement badges
 - **Milestones Reached:** Learning milestones achieved
 - **Special Accomplishments:** Notable achievements unlocked
-

Troubleshooting

Common Issues Issue

1: Can't Log In

Possible Causes:

- Incorrect email or password
- Account not verified
- Account locked due to too many failed attempts
- Browser cache issues
- Server connection problems

Solutions:

1. Check Credentials:

- Double-check your email address
- Verify your password (check caps lock)
- Try typing password in a text editor first to verify

2. Email Verification:

- Check your email inbox for verification link
- Check spam/junk folder
- Request new verification email if needed

3. Forgot Password:

- Click "Forgot Password" link
- Enter your email address

- Check email for reset link
- Follow instructions to reset password

4. Account Locked:

- Wait 15 minutes if locked
- Account unlocks automatically
- Try logging in again after waiting

5. Browser Issues:

- Clear browser cache
- Try a different browser
- Disable browser extensions temporarily
- Check if JavaScript is enabled

6. Contact Support:

- If issues persist, contact support
 - Provide details about the problem
 - Include error messages if any
-

Issue 2: Pre-Assessment Not Unlocking Subtopics

Possible Causes:

- Assessment not fully completed
- Assessment not submitted
- Browser refresh needed
- Server sync delay

Solutions:

1. **Verify Completion:** Ensure you submitted the assessment
2. **Check Results:** View your assessment results page
3. **Refresh Page:** Try refreshing the browser page
4. **Log Out/In:** Log out and log back in
5. **Wait:** Wait a few minutes for server sync
6. **Contact Support:** If problem persists

Issue 3: Challenges Not Loading

Possible Causes:

- Slow internet connection
- Browser cache issues
- Server connection problems
- JavaScript errors

Solutions:

1. **Check Internet:** Verify your internet connection
 2. **Clear Cache:** Clear browser cache and cookies
 3. **Try Different Browser:** Test in another browser
 4. **Refresh:** Hard refresh (Ctrl+F5 or Cmd+Shift+R)
 5. **Check Console:** Look for JavaScript errors in browser console
 6. **Server Status:** Check if server is operational
-

Issue 4: Retention Test Not Available

Possible Causes:

- Not enough time has passed
- Topic not fully completed
- Pre/post assessments not completed
- System calculation error

Solutions:

1. **Check Countdown:** Look at the countdown timer
2. **Verify Completion:** Ensure all requirements are met:
 - Pre-assessment completed
 - All subtopics completed
 - Post-assessment completed
3. **Wait:** Wait for the availability window
4. **Check Icon:** Look for retention test icon () on topic card
5. **Refresh:** Refresh the page to update status

Issue 5: Progress Not Updating

Possible Causes:

- Browser cache
- Data not synced to server
- Server delay
- Browser session issues

Solutions:

1. **Refresh:** Refresh the page (F5)
 2. **Clear Cache:** Clear browser cache
 3. **Wait:** Wait a few minutes for server sync
 4. **Log Out/In:** Log out and log back in
 5. **Check Network:** Verify internet connection is stable
-

Getting Help

Support Options:

1. Check Documentation

- Review this user manual
- Check the README.md for technical details
- Look for FAQ section (if available)

2. Contact Support

- Use the contact form on the platform
- Email support (if email provided)
- Include detailed problem description
- Attach screenshots if helpful

3. Report Bugs

- Use the bug report feature
- Describe the issue clearly
- Include steps to reproduce
- Note your browser and OS

4. Community Forum

- Search for similar issues
- Ask questions
- Help other users

For Developers

System Architecture

Frontend Architecture

Technology Stack:

- **Framework:** React 19 with Vite
- **Styling:** SCSS with Bootstrap 5
- **State Management:** React Context API
- **Routing:** React Router DOM v7
- **Code Editor:** Monaco Editor
- **Charts:** Recharts
- **Animations:** Framer Motion
- **Build Tool:** Vite 6

Key Directories:

```
clove-frontend/
├── src/
│   ├── features/      # Feature-based modules
│   ├── components/    # Reusable UI components
│   ├── contexts/      # React context providers
│   ├── hooks/         # Custom React hooks
│   └── styles/        # SCSS styling system
```

Backend Architecture

Technology Stack:

- **Framework:** FastAPI (Python)
- **Database:** PostgreSQL
- **ORM:** SQLAlchemy 2.0 (async)
- **Authentication:** JWT tokens with refresh
- **API:** RESTful with OpenAPI/Swagger docs
- **Migrations:** Alembic
- **Email:** Brevo integration

Key Directories:

```
clove-backend/
├── app/
│   ├── api/          # API endpoints
│   ├── core/         # Core algorithms (BKT, RL)
│   ├── crud/         # Database operations
│   ├── db/           # Database models and session
│   ├── schemas/      # Pydantic schemas
│   └── services/     # Business logic services
```

Deployment Architecture

Deployment Architecture:

- Frontend: Netlify (React App)
- Backend: Render.com (FastAPI)
- Database: Render.com (PostgreSQL)

Flow: Frontend ** Backend ** Database

- **Frontend:** Deployed on Netlify
- **Backend:** Deployed on Render.com
- **Database:** PostgreSQL hosted on Render.com

Core Algorithms

1. Bayesian Knowledge Tracing (BKT)

Purpose: Models student knowledge state and predicts learning outcomes

Location: `clove-backend/app/core/bkt.py`

Parameters:

- `p_T = 0.1` (Transition probability - likelihood of learning)
- `p_G = 0.2` (Guess probability - chance of correct answer without knowledge)
- `p_S = 0.1` (Slip probability - chance of incorrect answer despite knowledge)

Key Method: `update_knowledge(knowledge_prob, is_correct)`

How It Works:

- Updates knowledge probability after each challenge attempt
- Predicts likelihood of correct answer on next attempt
- Tracks mastery progression over time

2. Q-Learning (Reinforcement Learning)

Purpose: Optimizes challenge selection strategy based on student performance

Location: `clove-backend/app/core/rl.py`

Parameters:

- `a = 0.1` (Learning rate)
- `y = 0.9` (Discount factor)
- `ε = 0.1` (Initial exploration rate, decays to 0.1)

State Space: (Mastery Level, Timer Active, Hint Active)

Actions: Challenge types (code_fixer, code_completion, output_tracing)

Rewards: Based on correctness, hints used, timing, and streak
Key Methods:

- `select_action(state)` : Chooses best challenge for student
- `update_q_value(current_state, action, reward, next_state)` : Learns from experience

3. Adaptive Challenge Selection

Purpose: Intelligently selects the most appropriate challenge for each student

Location: `clove-backend/app/services/selection.py`

Mastery Classification:

- Beginner: 0-33% knowledge
- Intermediate: 34-66% knowledge
- Advanced: 67-100% knowledge

Features:

- Automatically adjusts challenge difficulty based on mastery level
- Tracks correct/incorrect streaks to activate timers and hints
- Considers student's historical performance

Key Functions:

- `_select_adaptive_challenge()` : Selects challenge using AI algorithms
- `_select_non_adaptive_challenge()` : Fallback selection method

Database Backup

Automated Backups:

Schedule:

- **Frequency:** Daily at 2 AM UTC
- **Storage:** Google Drive
- **Cleanup:** Automatic (keeps last 3 days)
- **Manual:** Available via GitHub Actions

Backup Types:

- **Full:** Complete database backup
- **Schema Only:** Just table structures
- **Data Only:** Just data, no structures

Setup: See [DATABASE_AUTOMATIC_BACKUP_QUICK_SETUP.md](#) for detailed instruction

Development Setup

Prerequisites:

- Node.js 18+
- Python 3.11+
- PostgreSQL
- Git

Frontend Setup:

```
cd clove-frontend  
npm install  
npm run dev
```

Backend Setup:

```
cd clove-backend  
python -m venv venv  
source venv/bin/activate # Windows: venv\Scripts\activate  
pip install -r requirements.txt  
alembic upgrade head  
uvicorn app.main:app --reload
```

Environment Variables:

- Create `.env` file in `clove-backend/`
 - Add database URL, JWT secrets, API keys
 - See `.env.template` for required variables
-

API Documentation

Access: Visit `/docs` on the backend server (when running locally)

Features:

- Interactive API documentation (Swagger UI)
- Test endpoints directly from browser
- View request/response schemas
- Authentication testing interface

Endpoints:

- `/api/auth/*` - Authentication endpoints
 - `/api/users/*` - User management
 - `/api/topics/*` - Topic operations
 - `/api/challenges/*` - Challenge operations
 - `/api/assessments/*` - Assessment operations
-

Key Files Reference

Frontend Key Files:

- `src/App.jsx`: Main application router
- `src/features/`: Feature-based modules
- `src/components/`: Reusable components
- `src/context/`: Context providers (Auth, MyDeck, etc.)

Backend Key Files:

- `app/main.py`: FastAPI application entry point
- `app/api/`: API endpoint definitions
- `app/core/`: Core algorithms (BKT, RL, utils)
- `app/crud/`: Database CRUD operations
- `app/db/models/`: SQLAlchemy database models
- `app/services/`: Business logic services

Additional Resources

Learning Tips

1. **Consistency:** Practice regularly for best results
2. **Take Notes:** Write down important concepts
3. **Review:** Revisit completed topics periodically
4. **Use Hints Wisely:** Try first, then use hints if stuck
5. **Complete Assessments:** Don't skip pre/post assessments
6. **Take Retention Tests:** Verify long-term learning

Best Practices

1. **Start with Pre-Assessment:** Always take it seriously
2. **Read Lessons Thoroughly:** Don't rush through content
3. **Practice Regularly:** Complete practice exercises
4. **Attempt Challenges:** Test your knowledge
5. **Review Mistakes:** Learn from incorrect answers
6. **Track Progress:** Monitor your improvement

Glossary

Term	Definition
Assessment	Evaluation test (pre or post) to measure knowledge
BKT	Bayesian Knowledge Tracing - algorithm that models student knowledge
Challenge	Interactive coding exercise (Code Fixer, Code Completion, Output Tracing)
Mastery Level	Your knowledge level for a specific subtopic (0-100%)
Post-Assessment	Test taken after completing all subtopics to measure learning
Pre-Assessment	Test taken before starting a topic to evaluate baseline knowledge

Q-Learning	Reinforcement learning algorithm for challenge selection
Realm	Thematic environment (Wizard Academy, Detective Agency, Space Station)
Retention Test	Test taken after topic completion to verify long-term retention (Stage 1: 10h, Stage 2: 5 days)
Subtopic	A sub-section within a main topic
Topic	Main learning unit containing multiple subtopics
Traveler Class	Your chosen learning style (Syntax Mage, Logic Detective, Algorithm Explorer)

Version History

Version	Date	Changes
1.0	December 2024	Initial user manual release

Contact & Support

For questions, issues, or feedback:

- **Check Documentation:** Review this manual first
- **Technical Details:** See README.md for developers
- **Contact Support:** Use platform contact form (cloveconnect@gmail.com)
- **Report Bugs:** Use bug report feature

Happy Learning!

This manual is a living document and will be updated as new features are added.

CLOVE - Code, Logic, Overcome, Validate, Excel
All Rights Reserved @ 2025