

# **CodeVent: Where Coding meets Adventure**

**Description/Overview** - CodeVent is an educational Java console application that transforms learning programming into an engaging adventure game. Designed for beginners, this interactive platform teaches fundamental Java concepts through a gamified experience featuring colorful console interfaces, interactive lessons, and challenging quizzes.

The application addresses the common struggle students face when learning programming - the complexity and dryness of traditional tutorials. By framing programming concepts as exciting adventures and challenges, CodeVent makes learning Java intuitive, memorable, and enjoyable.

## **Key Features:**

- 16 Comprehensive Java Lessons covering syntax to advanced concepts
- Interactive Quiz System with immediate feedback and scoring
- Gamified Learning Experience with adventure-themed progression
- Colorful Console Interface with ASCII art and animations
- Structured Learning Path from beginner to intermediate topics

## **OOP Concepts applied -**

### **Encapsulation**

- All class fields are declared as `private` with controlled access through public getters and setters
- The `Quiz` class encapsulates scanner management and quiz logic
- `Help` class encapsulates all tutorial content and display methods

- Lesson classes encapsulate their specific educational content and display logic

## Inheritance

- Lesson classes inherit common structure and behavior through shared design patterns
- Quiz system uses consistent method signatures across different topic implementations
- Color management and console utilities follow inherited patterns

## Polymorphism

- Dynamic method dispatch through the lesson selection system
- Overloaded constructor methods in multiple classes
- Interface-like behavior through consistent startMenu() methods across classes
- Runtime binding in quiz answer validation system

## Abstraction

- Abstract game flow control separated from concrete implementation
- High-level menu systems abstracting complex user interactions
- Modular design allowing independent development of lessons, quizzes, and UI components
- Separation of educational content from game mechanics

#### 4. Program Structure -CodeVent/

```
|—— CodeVent.java (Main Controller)
|   |—— Manages overall application flow
|   |—— Displays main menu and ASCII art
|   |—— Coordinates between all components
|—— Lessons.java (Lesson Manager)
|   |—— Handles lesson selection menu
|   |—— Routes to individual lesson classes
|   |—— Manages lesson progression
|—— Quiz.java (Assessment System)
|   |—— Manages quiz selection interface
|   |—— Implements 16 topic-specific quizzes
|   |—— Handles scoring and feedback
|   |—— Provides immediate answer validation
└—— Help.java (Tutorial System)
    |—— Comprehensive game tutorial
    |—— Learning guidance and tips
    |—— Interactive help menu system
```

## **Relationships**

- CodeVent aggregates Lessons, Quiz, and Help
- Lessons composes individual lesson implementations
- Quiz contains multiple quiz topic implementations
- All classes follow the Single Responsibility Principle

### **5. How to Run the Program -**

-Java JDK 8 or higher

-Command line terminal or IDE

### **Step-by-Step Instructions:**

1. **Download the Project:** `git clone https://github.com/frzj1103-lgt/CodeVent.git`
  2. **Go to cmd and type :** `cd` CodeVent
  3. **Compile the Program:** `javac CodeVent.java`
  4. **Run the Application:** `java` CodeVent
- 5. Using an IDE:**
- Open the project in Eclipse, IntelliJ, or VS Code
  - Locate `CodeVent.java` in the `CodeVent` package
  - Run the main class (`CodeVent.java`) directly

### **Navigation:**

1. Select options using number keys (1-4)
2. Follow on-screen instructions for lessons and quizzes
3. Use '0' to exit menus and return to previous screens

## 6. Sample output -



7.

## Authors:

**Boñon, Franz Jacob** - In line with applying the logics and the principles of OOP, as well as implementing the ASCII colorful codes and animations.

**Factor, Brent Jeric** - For creating all the quizzes in the project and analyzing each logic on how to connect it with the main class.

**Perez, Justin Raven** - By applying the lectures in the project and using oop principles in creating the program.

### **Acknowledgements:**

- W3Schools for comprehensive Java documentation and learning resources
- Java Platform for robust console application capabilities
- Educational Researchers whose work inspired gamified learning approaches
- Open Source Community for best practices in software design and documentation

### **Future Enhancements**

Expanding the topic and concepts into professional-level codes, since the project's scope is until arrays only, as well as adding a user login to the program to make it more usable when it comes to many users.

### **References**

W3Schools Java Tutorials

Oracle Java Documentation

Gamification in Education Research Papers

### **Technical References:**

Java Console Application Best Practices

ANSI Color Codes for Terminal Applications

Object-Oriented Design Principles

Software Architecture Patterns