**University of The People**

[**CS 1103-01 Programming 2**](https://my.uopeople.edu/course/view.php?id=8111)

***Instructor Name****: Honore Nzambu*

***Programming Assignment Unit-3***

**Project Name:** Clock Application

**Tool:** Eclipse, JDK11,Maven

**Programming Language**: Java

***Environment: DEV***

### **Project Objective:**

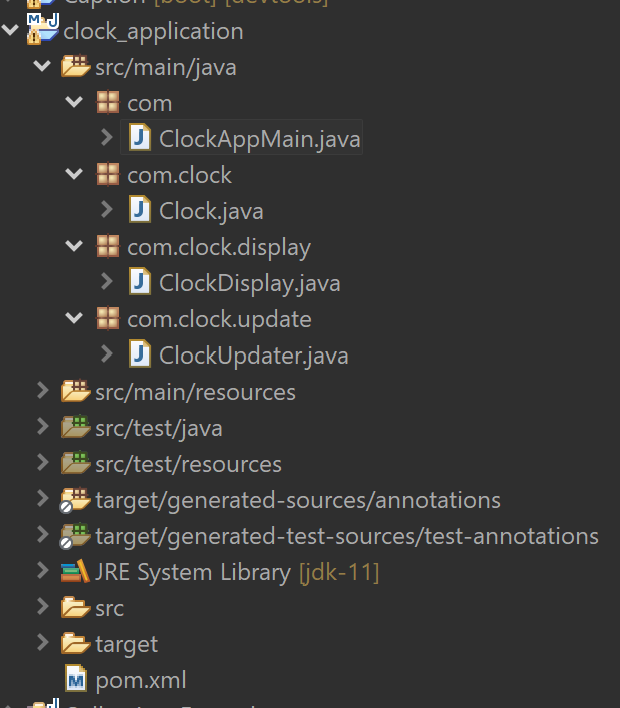
The objective of this project is to develop a simple clock application using Java's thread model. This application will utilize multithreading to concurrently display the current time and date, as well as update it in the background. The key goals of the project are as follows:

1. **Explore the Basics of Java Threads**: Understand and apply the basic concepts of Java threading by creating separate threads for updating and displaying the current time and date.
2. **Thread Synchronization**: Ensure that multiple threads operate seamlessly without conflicts through proper synchronization techniques, allowing safe access to shared resources.
3. **Thread Prioritization**: Demonstrate the use of thread priorities to achieve better timekeeping precision. The clock display thread is given a higher priority over the background updating thread to ensure that the time display is prioritized.
4. **Continuous Time and Date Display**: Implement a system that continuously updates and prints the current time and date in a readable format (HH:mm:ss dd-MM-yyyy), ensuring that the displayed time is synchronized with the background time updates.
5. **Concurrency Handling**: Address concurrency challenges, ensuring that threads work in tandem without causing inconsistencies or race conditions.
6. **Practical Application of Multithreading**: Provide a real-world scenario where the understanding of Java threading, synchronization, and priority assignment is essential in building a practical, time-sensitive application like a digital clock.

### Project Overview:

The final result is a clock application that continuously updates and displays the current time and date. This project showcases multithreading, thread synchronization, and thread prioritization within a simple but practical scenario, reinforcing key programming concepts and best practices in Java.

### Project Structure:



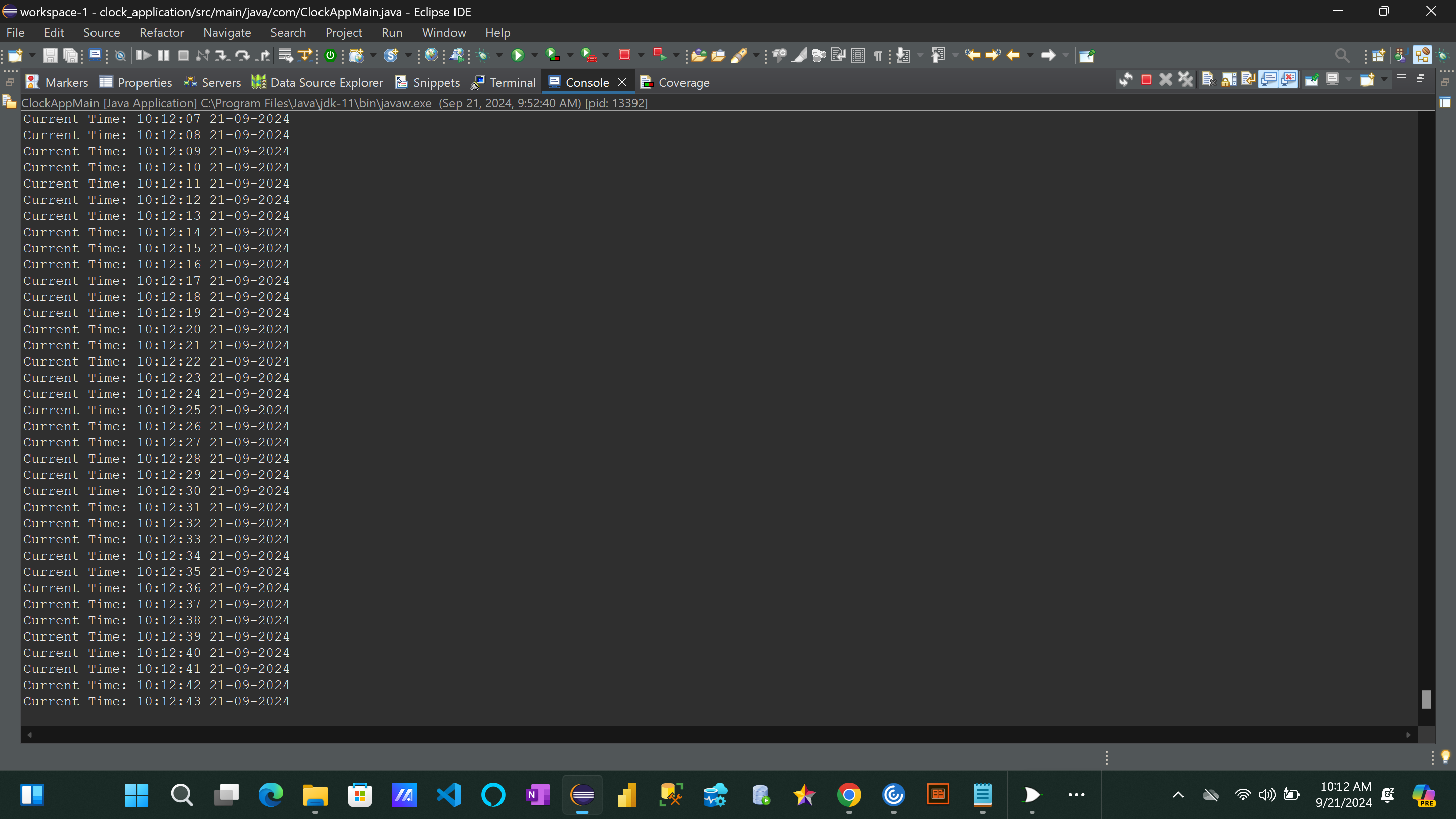
### Explanation of the Code:

1. **Clock Class**:
   * The Clock class contains the shared time object. The updateTime() method updates the current time, while the getCurrentTime() method retrieves the current time in a formatted manner.
   * Both methods are synchronized to ensure proper concurrency and avoid race conditions when multiple threads try to access or modify the currentTime.
2. **ClockDisplay Class**:
   * The ClockDisplay class implements Runnable and is responsible for displaying the time.
   * It retrieves the time from the shared Clock object every second and prints it to the console.
3. **ClockUpdater Class**:
   * The ClockUpdater class, also implementing Runnable, is responsible for updating the currentTime in the Clock class.
   * It updates the time every second by calling the updateTime() method.
4. **ClockApp Class**:
   * This is the main class where the threads are created and managed.
   * Two threads are created: one for updating the time (with lower priority) and one for displaying the time (with higher priority).
   * The threads are started to run concurrently, with thread priorities set to ensure the display thread gets more CPU time for precision.

### Requirements Satisfied:

1. **Thread Implementation**:
   * Threads are used appropriately, with separate threads for updating the time and printing it to the console.
   * Synchronization is added to handle concurrency issues properly, ensuring that there are no conflicts between the threads.
2. **Thread Priorities**:
   * The display thread has a higher priority than the background updating thread, as required.
3. **Readable Time Format**:
   * The current time is displayed in the HH:mm:ss dd-MM-yyyy format, and the clock continuously updates every second.

**Output Screenshot:**



**Reference:**

1.Eck, D. J. (2022). Introduction to programming using java version 9, JavaFX edition. Licensed under CC 4.0.   <https://math.hws.edu/javanotes/>

2.Packages in Java. (n.d.). Board Infinity. <https://www.boardinfinity.com/blog/packages-in-java/>

3.Samoylov, N. (2018). Introduction to programming: Learn to program in java with data structures, algorithms, and logic. Packt Publishing, Limited.