Derek McPeak

Douglas Lundin

CSC245

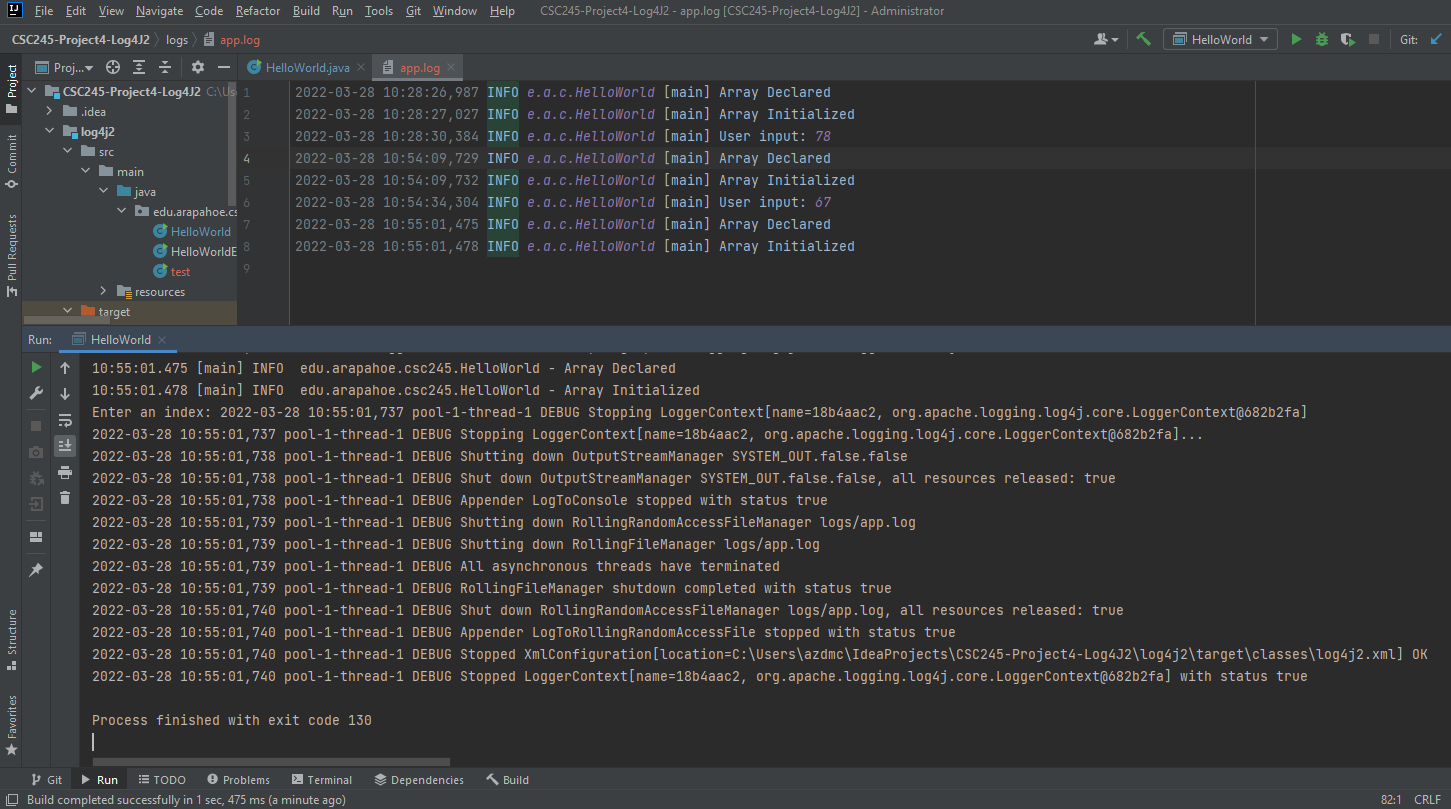
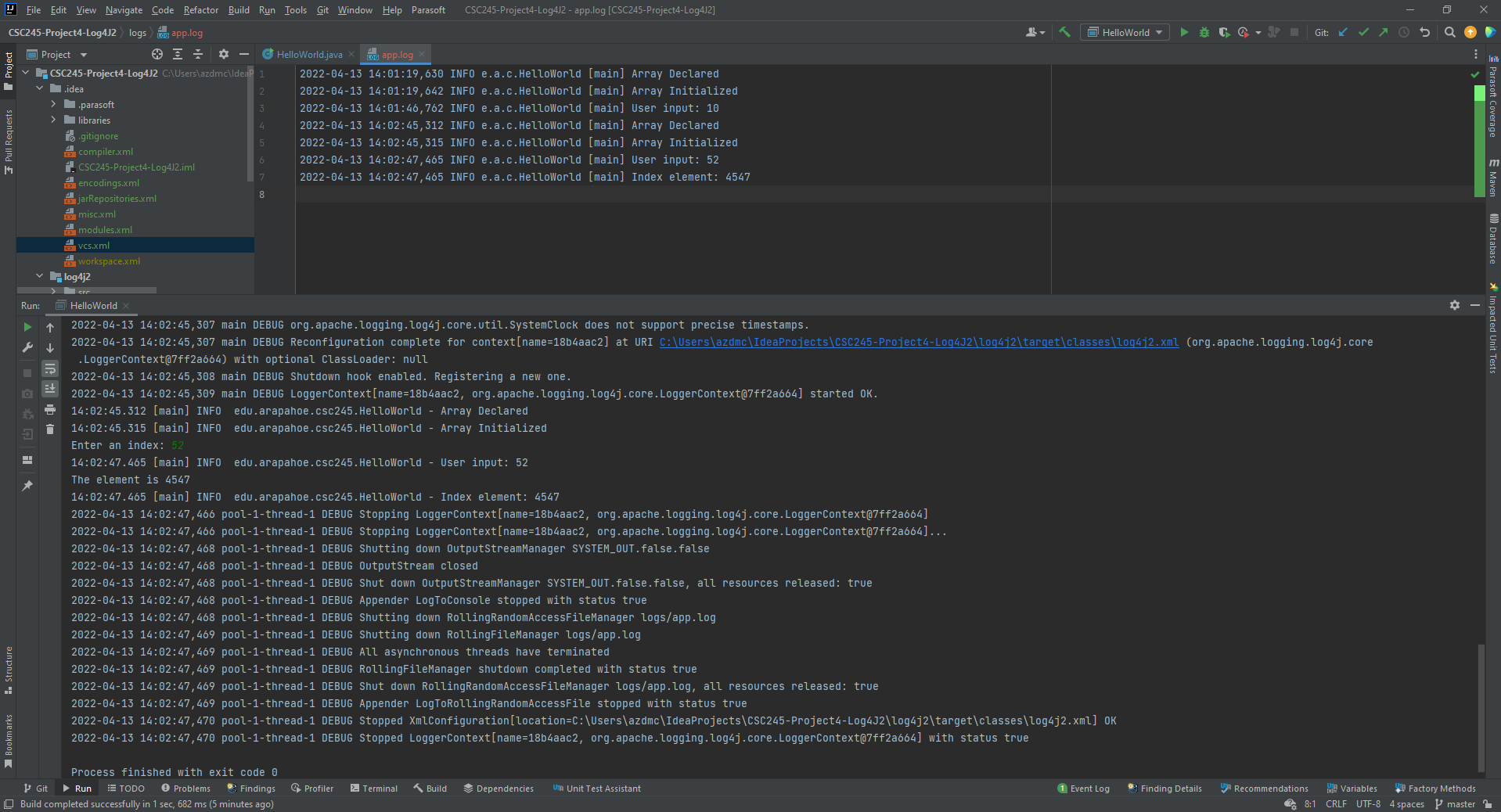
Project 4 / Task 1

The log4j functionality, more so logging as a concept, is paramount to having a fully featured program. Log4j makes the implementation of use of logging easy to use and implement by having built in features like debug, fatal, error, warm, info, trace and many more. When choosing what to log, there are creative freedoms on what to log but at the same time there needs to be a pragmatic approach to logging. If a client wants to know how many times Y feature is clicked on, you could create a log for each time that feature is accessed then use a for loop to read/count each entry when the data is requested. The presumption logging is a universal function in all production programs, it is a creative path of entry for an attack so we must make sure this vector is secure.

Page 3 – Displaying the code runs properly

Page 4 – Successful addition of code into Intellij

Page 5 & 6 – Full source code showing comments, naming convention and formatting added to code

Displaying the functionality of the program is running and logging successfully

Text

Description automatically generatedSuccessful addition of code in Intellij / Add Exercise 12.3

Commented code

// Derek McPeak  
// 03/28/2022  
// Douglas Lundin CSC245  
  
package edu.arapahoe.csc245;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
import java.util.\*;  
  
public class HelloWorld {  
  
 private static final Logger *logger* = LogManager.*getLogger*(HelloWorld.class);  
  
 public static void main(String[] args) {  
  
 //declaring the array and its size  
 int[] RandomNumberArray = new int[100];  
 // logging when the array is declared  
 *logger*.info("Array Declared");  
  
 // Initialize array  
 // Inserting a random number between 0 - 10000 into each element of the array  
 for (int incrementer = 0; incrementer < 100; incrementer++)  
 RandomNumberArray[incrementer] = (int)(Math.*random*() \* 10000);  
 // logging when the array is initialized  
 *logger*.info("Array Initialized");  
  
 try {  
 //scanner for user input  
 Scanner input = new Scanner(System.*in*);  
  
 //prompting user to enter input  
 System.*out*.print("Enter an index: ");  
  
 // variable to users input  
 int index = input.nextInt();  
 // logging the input of the user  
 *logger*.info("User input: " + index);  
  
 // printing the value of the user chosen index of the array.  
 System.*out*.println("The element is " + RandomNumberArray[index]);  
  
 // logging the value of the index  
 *logger*.info ("Index element: " + RandomNumberArray[index] );  
  
 }  
 catch (InputMismatchException ex) {  
 System.*out*.println("Input Mismatch, please enter valid input");  
 // logging if there is an input mismatch  
 *logger*.debug("Unexpected input");  
 }  
 catch (ArrayIndexOutOfBoundsException ex){  
 System.*out*.println("Index is out of bounds, please enter inbounds index");  
 // logging the input of the user if the index is out of bounds  
 *logger*.debug("Index is out of bounds");  
 }  
  
  
/\*  
  
 //Test logs  
 logger.debug("Hello from Log4j 2");  
 logger.debug("Example of a DEBUG {}", () -> 1);  
 logger.fatal("Example of a FATAL {}", () -> 2);  
 logger.error("Example of a ERROR {}", () -> 3);  
 logger.warn("Example of a WARN {}", () -> 4);  
 logger.info("Example of a INFO {}", () -> 5);  
 logger.trace("Example of a TRACE {}", () -> 6);  
  
 // while (true)//test rolling file  
 // logger.debug("hello {}", () -> getNumber());  
 \*/