CS 312 – Exam 2 – Fall 2022 SOLUTIONS

Code Tracing - 4 points each

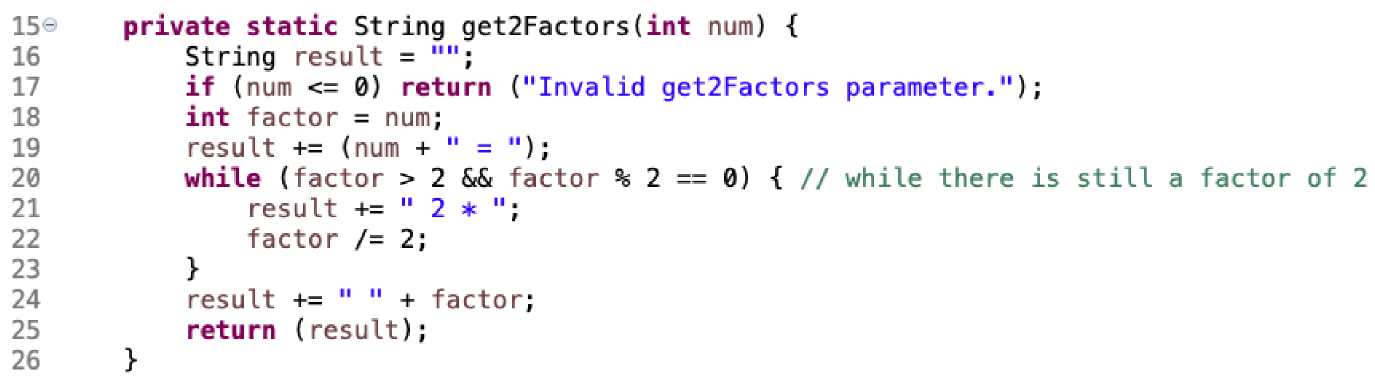
1. `true`
2. `5`
3. `a`
4. `2`
5. `infinite loop`
6. `runtime error`
7. `10.0 10.1`
8. `5.7113.9CSUT`
9. `runtime error`
10. `[15, 2, -1, 11, 2]`
11. `[7, 9, 8, 0]`
12. `[1, 0] [6, 4, 4]`
13. `[1, 1, 6, 6, 9]`
14. `[4, 5, 6, 6][5, 6, 7, 7][6, 7, 8, 8]`

Assertions - 14 Points

2 points for free, then +1 for each correct answer

	n > b	**a > 1**	**b > a**
Point A	S	A	N
Point B	A	S	S
Point C	S	A	S
Point D	N	S	S

Program - get2Factors() - 32 points



```
15      private static String get2Factors(int num) {
16          String result = "";
17          if (num <= 0) return ("Invalid get2Factors parameter.");
18          int factor = num;
19          result += (num + " = ");
20          while (factor > 2 && factor % 2 == 0) { // while there is still a factor of 2
21              result += " 2 * ";
22              factor /= 2;
23          }
24          result += " " + factor;
25          return (result);
26      }
```

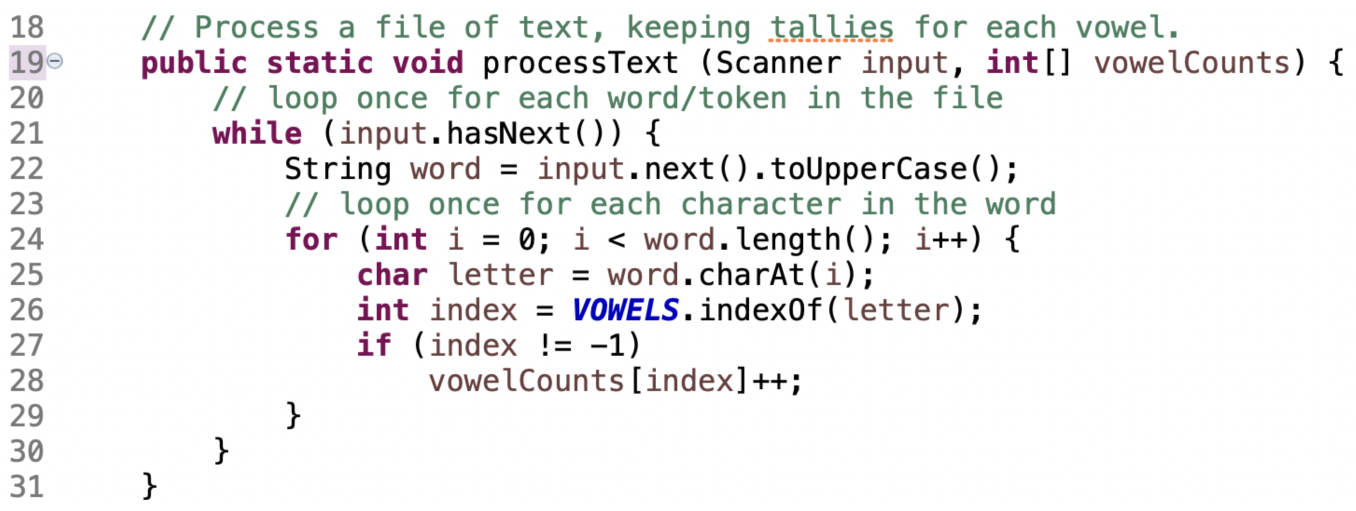
Item	Line #	Item	Points
A	15	Method declaration	+2 String return type +2 one int parameter
B	17	Check for bad parameter	+2 returns if num <=0 (-1 if close: n<0 or n==0) +2 the return is before the while loop +2 return with correct message
C	19 - 25	Result variable	+1 initialized before loop +4 correct accumulation in loop - math +4 correct accumulation in loop and after string (handles even and odd numbers) +1 correct return
D	20	While loop	+2 stops when no more factors of 2 to find +2 stops before adding the number 1 to the output
E	N/A	GENERAL	+2 No unnecessary code +2 No redundant or inefficient code +2 No Java syntax not allowed +2 No seriously incorrect style guide issue (if line count == 0, 0 max points, else if line count < 6, max 4 points, else max 8 points)

Program - containsBackwards() - 32 points

```java
13      private static boolean containsBackwards (String str1, String str2) {
14
15          // if first string is shorter, return false
16          if (str1.length() < str2.length()) return false;
17
18          // loop once for each possible position of the smaller string
19          for (int i = str1.length()-1; i>= str2.length()-1; i--) {
20              boolean match = true; // assume this position is a match
21              // loop once for each character in the smaller string
22              for (int j = 0; j<str2.length(); j++) {
23                  if (str1.toLowerCase().charAt(i-j) != str2.toLowerCase().charAt(j)) match = false;
24              }
25              // if match is still true, the smaller string was found
26              if (match) return true;
27          }
28          // if you get out of the outside for loop without returning, the smaller string wasn't found
29          return false;
30      } // end method
```

Item	Line #	Item	Points
A	13	Method declaration	+1 boolean return type +1 two String parameters
B	16	Invalid parameter	+2 if second string if larger than first, return false
C	19	For loop, once for each possible position of match	(Main idea: Traverse through the first string, looping one time for each possible match. For example, can also traverse forward.) +2 counter starts at last index in string +2 loop continues while i >= smaller string length-1 +1 counter decremented by 1
D	20, 23, 26, 29	Correct boolean return	+2 boolean value initialized correctly +2 boolean value updated correctly +1 boolean value returned correctly
E	22	Second for loop, once for each character in the smaller string	(Main idea: Determine if there is a match at a given position. For example, can traverse backwards or use indexOf with a string parameter.) +2 counter starts with 0 and incremented by 1 +2 loop continues while counter less than length of smaller string
F	23	Character comparison	+2 not case sensitive +2 correct index for both strings +2 correct update to boolean value and timing of return
G	N/A	GENERAL	+2 No unnecessary code +2 No redundant or inefficient code +2 No Java syntax not allowed: contains() and substring() +2 No seriously incorrect style guide issue (if line count == 0, 0 max points, else if line count < 6, max 4 points, else max 8 points)

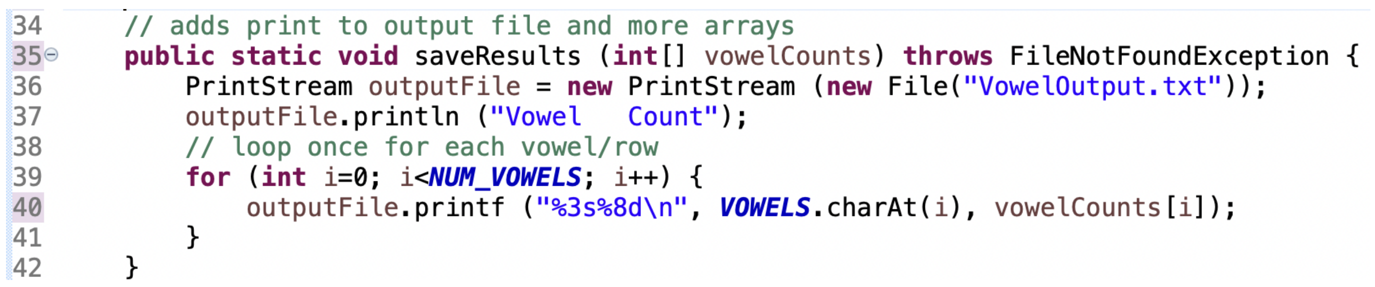
Program - Vowel Text Analysis A - processText() - 32 points



```
18      // Process a file of text, keeping tallies for each vowel.
19      public static void processText (Scanner input, int[] vowelCounts) {
20          // loop once for each word/token in the file
21          while (input.hasNext()) {
22              String word = input.next().toUpperCase();
23              // loop once for each character in the word
24              for (int i = 0; i < word.length(); i++) {
25                  char letter = word.charAt(i);
26                  int index = VOWELS.indexOf(letter);
27                  if (index != -1)
28                      vowelCounts[index]++;
29              }
30          }
31      }
```

Item	Line #	Item	Points
A	20	Method declaration	+2 void return type +2 Scanner parameter +2 int[] parameter
B	22	Loop for each token in file	+2 loop while there is another token in the file (don't consume yet) +2 read the next token inside loop +2 toUppercase
C	25	Loop for each character in line	+2 correct string traversal array (initialize, test and increment) +2 inside loop, use String for vowels (preferably the constant)
D	28	Tally increments	+3 correct letter +3 correct index +2 correct increment
E	N/A	GENERAL	+2 No unnecessary code +2 No redundant or inefficient code +2 No Java syntax not allowed +2 No seriously incorrect style guide issue (if line count == 0, 0 max points, else if line count < 6, max 4 points, else max 8 points)

Program - Vowel Text Analysis B - saveResults() - 32 points



```
34      // adds print to output file and more arrays
35      public static void saveResults (int[] vowelCounts) throws FileNotFoundException {
36          PrintStream outputFile = new PrintStream (new File("VowelOutput.txt"));
37          outputFile.println ("Vowel   Count");
38          // loop once for each vowel/row
39          for (int i=0; i<NUM_VOWELS; i++) {
40              outputFile.printf ("%3s%8d\n", VOWELS.charAt(i), vowelCounts[i]);
41          }
42      }
```

Item	Line #	Item	Points
A	34	Method declaration	+2 void return type +2 int[] parameter +2 throws FileNotFoundException or IOException
B	35	Create PrintSream	+2 create a new file object for correct filename +2 create new PrintStream object with file object as parameter
C	39	For loop for each vowel	+2 correct initialization, test and increment +2 uses array length or NUM_VOWELS
D	37-41	Correct Output	+2 correct format for headings +4 correct output for vowels and counts +4 correct format for vowels and counts
E	N/A	GENERAL	+2 No unnecessary code +2 No redundant or inefficient code +2 No Java syntax not allowed +2 No seriously incorrect style guide issue (if line count == 0, 0 max points, else if line count < 2, max 4 points, else max 8 points)

Program - AlbumTime - main() - 32 Points

```java
7     public static void main(String[] args)
8         throws FileNotFoundException {
9         Scanner input = new Scanner(new File("songTimes.txt"));
10        // loop once per album
11        while (input.hasNextInt()) {
12            input.nextLine(); // consume album number line
13            String albumName = input.nextLine();
14            int totalSeconds = 0;
15            // loop once per line/song
16            while (input.hasNextLine() && !input.hasNextInt()) {
17                totalSeconds += getSongSeconds(input.nextLine());
18            }
19            displayResults (albumName, totalSeconds);
20        }
21        input.close();
22    } // end main
```

Item	Line #	Item	Points
A	9	Create Scanner	+2 create a new file object for correct filename +3 create new Scanner object with file object as parameter
B	11-14	while loop for each line in file	+3 correct while loop test, testing for existence of another album (hasNext() or hasNextLine() or hasNextInt()) +2 consumes album number +2 reads album name +2 initializes total seconds
C	16-18	while loop for each song in album	+3 correct test for nextline and not an album number +3 correct increment of total second for the album with call to getSongSeconds()
D	19	Wrap up	+2 correct call to displayResults() +2 close Scanner
E	N/A	GENERAL	+2 No unnecessary code +2 No redundant or inefficient code +2 No Java syntax not allowed +2 No seriously incorrect style guide issue (if line count == 0, 0 max points, else if line count < 6, max 4 points, else max 8 points)