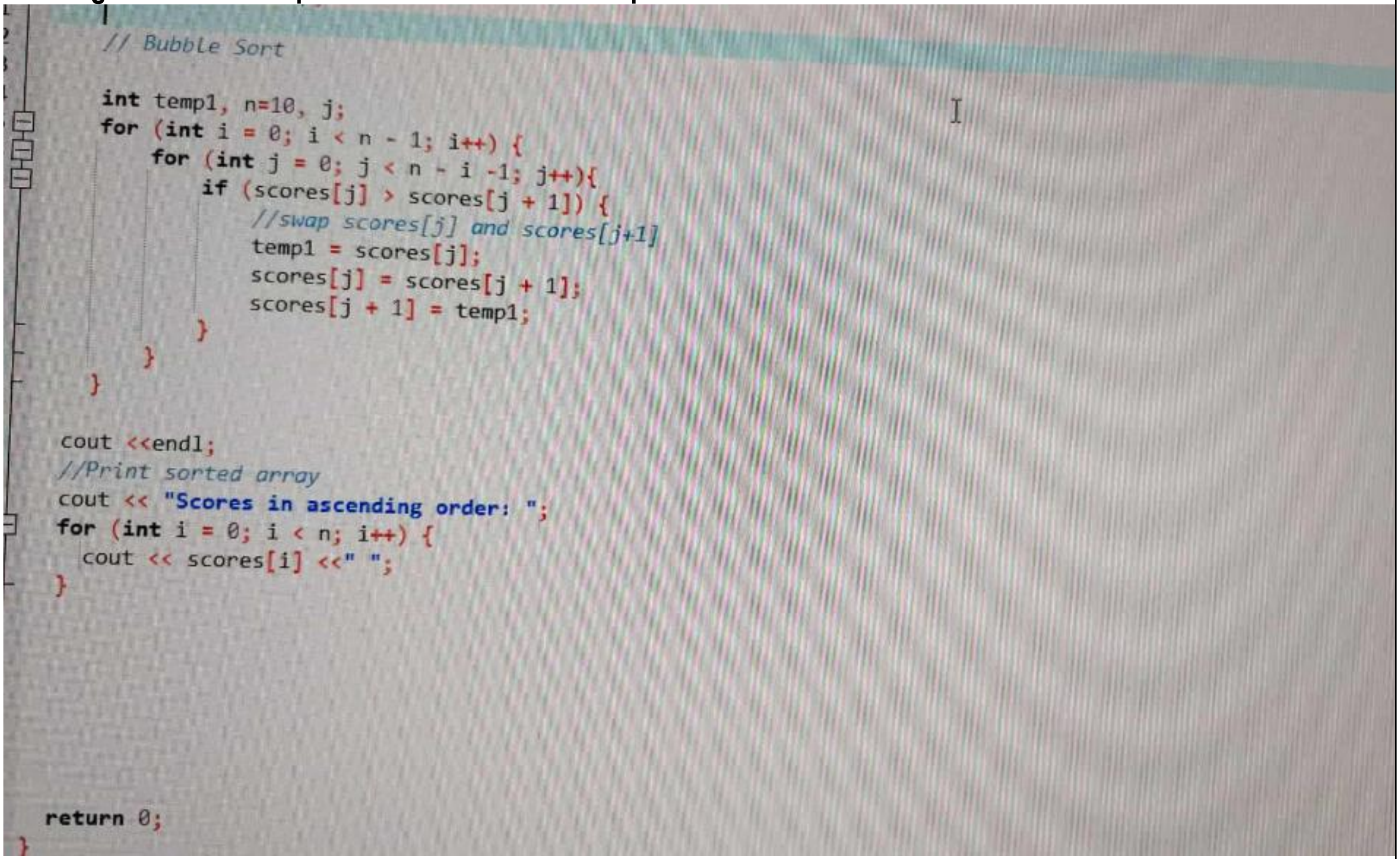
Activity No. <4.2>

<Bubble Sort >

Course Code: CPE007	**Program:** Computer Engineering
Course Title: Bubble Sort	**Date Performed:9/10/25**
Section:CPE11S1	**Date Submitted:9/10/25**
Name(s):Will Stuart D. Ponce Jr.	**Instructor: Engr. Jimlord Quejado**

6. Output
Working Bubble Sort Implementation Code and Output:



```
    // Bubble Sort

    int temp1, n=10, j;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i -1; j++){
            if (scores[j] > scores[j + 1]) {
                //swap scores[j] and scores[j+1]
                temp1 = scores[j];
                scores[j] = scores[j + 1];
                scores[j + 1] = temp1;
            }
        }
    }

  cout <<endl;
  //Print sorted array
  cout << "Scores in ascending order: ";
  for (int i = 0; i < n; i++) {
    cout << scores[i] <<" ";
  }

 return 0;
}
```

A comprehensive discussion of how bubble sort works
Bubble sort is an algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. This is done repeatedly until no more swaps are required, and then it is sorted. The highest (or lowest) elements "bubble" up (or down) to their correct position on each pass. It is intuitable but slow on big data sets.

7. Supplementary Activity

8. Conclusion: I learn its hard to do this bubble sort if you didn't know the easier to make one of that and I understand it very complicated to learn it if you didn't understand the words.

9. Assessment Rubric