CSC 171 Practice Problems on Loops and Arrays

Name: ____

Answer ALL questions. Show ALL work. CLEARLY indicate your final answer.
Each question has equal weight.

1. What output is produced by the following code fragment?

```
int  num = 0, max = 20;
while (num < max)
{
   System.out.println (num);
   num += 5;
}
```

2. Transform the while loop from the previous question into an equivalent for loop. (Make sure it produces the same output.)

3. Write a method named *isValid* which receives one parameter of type *double* representing a number of hours, and returns true if the number of hours is valid (between 0 and 168 inclusive), false otherwise.

4. Write a do loop that reads in a number of hours from the user (between 0 and 168); if the hours entered is invalid, print an error message and read the input again. (For full credit, call/invoke the method in the previous question appropriately.)

Continued →

5.

```
int x = 0;
while (x < 7) {
      x += 2;
 }
```

// What is the value of x after the loop is finished executing?

a) 2
b) 6
c) 7
d) 8
e) None of the above; this is an infinite loop.

6. Show the output, including proper spacing.

```
for (int j = 1; j <= 4; j++)
{
      for (int k = 1; k <= 4; k++)
       {
           if (j <= k)
               System.out.print(‘*‘);
           else
               System.out.print(‘ ’);
       }

      System.out.println();
}
```

7. Write a method to compute and return the value of the *smallest* element of an int array. The method is passed the int array and an int value that denotes the number of elements in the array. If the array has no elements, return –1.

Continued →

8. What is the exact output produced by running the method test?

```
/**
 * TestSwap.java
 * Demonstrates parameter passing involving arrays and integers,
 * scope of variables, flow of control, and overloaded methods.
 */
public class TestSwap
{
    public void swap (int x, int y)
    {
        int temp;
        temp = x;
        x = y;
        y = temp;
        System.out.println("Inside swap version 1:");
        System.out.println("x = " + x);
        System.out.println("y = " + y);
    }

    public void swap (int[] a, int i, int j)
    {
        int temp;
        temp = a[i];
        a[i] = a[j];
        a[j] = temp;
        System.out.println("Inside swap version 2:");
        System.out.println("a[" + i + "] = " + a[i]);
        System.out.println("a[" + j + "] = "+ a[j]);
    }

    public void printArray (int[] a)
    {
        System.out.print("Array elements: ");

        for (int i = 0; i < a.length; i++)
            System.out.print(a[i] + " ");

        System.out.println();
    }

    public static void test()
    {
        TestSwap t = new TestSwap();
        final int ARRAY_SIZE = 3;
        int[] arr = new int[ARRAY_SIZE];
        arr[0] = 3;
        arr[1] = 8;
        arr[2] = 6;
        int x = 1, y = 2;

        t.printArray (arr);
        t.swap (x, y);
        System.out.println ("Inside test:");
        System.out.println ("x = " + x);
        System.out.println ("y = " + y);
        t.printArray (arr);
        t.swap (arr[x], arr[y]);
        t.printArray (arr);
        t.swap (arr, x, y);
        t.printArray (arr);
    }
}
```

Continued →

9. Assume that the methods below (*reverse* and *swap*) are defined in the same class. Explain what is wrong and show all corrections that are needed.

```
/**
 * reverse: reverses the elements of an array of integers.
 * For example, if the array elements are 85, 99, 78, 100, then the array should be changed so
 * that its elements are 100, 78, 99, 85.
 *
 * @param   a       an array of integers
 * @param   len     the number of elements in the array
 */

public void reverse(int[] a, int len)
{
        for (int i = 0; i < len; i++)
            this.swap(a[i], a[len – i]);
}


/**
 * swap: swaps (exchanges) two values.
 *
 * @param  x  the first value
 * @param  y  the second value
 */

public void swap(int x, int y)
{
        int temp;
        temp = x;
        x = y;
        y = temp;
}
```

Name: ____

Extra scrap paper (if needed).

Name: ____

Extra scrap paper (if needed).