

ANSWER KEY

1.	(2 pts) E	13.	(1 pt) E	25.	(1 pt) D	37.	(2 pts) A
2.	(2 pts) A	14.	(1 pt) AC	26.	(1 pt) A	38.	(2 pts) AE
3.	(2 pts) C	15.	(1 pt) C	27.	(1 pt) B	39.	(2 pts) BE
4.	(2 pts) D	16.	(1 pt) BD	28.	(1 pt) D	40.	(2 pts) AB
5.	(1 pt) AD	17.	(1 pt) A	29.	(1 pt) A	41.	(2 pts) A
6.	(1 pt) D	18.	(1 pt) A	30.	(1 pt) B	42.	(2 pts) C
7.	(1 pt) CE	19.	(1 pt) B	31.	(1 pt) B	43.	(2 pts) A
8.	(1 pt) B	20.	(1 pt) B	32.	(1 pt) B	44.	(2 pts) D
9.	(1 pt) A	21.	(1 pt) A	33.	(1 pt) A	45.	(2 pts) B
10.	(1 pt) A	22.	(2 pts) B	34.	(1 pt) B	46.	(2 pts) A
11.	(1 pt) C	23.	(2 pts) C	35.	(1 pt) A		
12.	(1 pt) A	24.	(1 pt) E	36.	(2 pts) AC		

47. (12 pts; 2 pts each)

Line	Error	Correction
11	Missing datatype	int numTrials = ...
18	Missing # elements	... = new int[52];
37	Division will always be 0 (logical error)	Cast numerator (or denominator) to int
40	Function is not static	public static Boolean isFlush(...
48	Wrong input type	Change “int[]” to “int”
52/53	Missing final return statement	Add “return INVALID_SUIT;” to end of function

48.

```

public static int numPerfectSquares(int[] arr) {
    int num = 0;
    for (int i = 0; i < arr.length; i++) {
        int integerSqrt = (int) Math.sqrt(arr[i]);
        if (integerSqrt == Math.sqrt(arr[i])) {
            num++;
        }
    }
    return num;
}

```

49.

Note: when this question was graded, we chose to not penalize for incorrect handling of left shifts; correct handling of right shifts was the only thing required to earn you full credit. Consequently, the solution below only handles right shifts correctly.

```

public static int[] cycle(int[] arr, int shift) {
    if (arr == null) return null;
    int[] result = new int[arr.length];
    for (int i = 0; i < arr.length; i++) {
        int position = (i + shift) % arr.length;
        result[position] = arr[i];
    }
    return result;
}

```