CSE2 Final Exam Fall 2013 Name_____

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

Put your name on each page, or risk having an incompletely graded exam. Raise your hand to notify a proctor if you have a question. **No talking, no tapping or other noise. No cell phones, calculators, or other electronics.**

Read the entire exam and answer the questions that seem easiest first. Write answers in the box when provided. T**here are 250 points on this exam.**

Note: Usage of a method from one question as part of another question does not require the first question to be correctly answered. We will evaluate the second as if the method was correctly implemented.

Special circumstances only:
- ☐ Challenge exam
- ☐ Anticipatory exam

Grader Use Only

Question 1:	/20	Question 9:	/20
Question 2:	/25	Question 10:	/20
Question 3:	/20	Question 11:	/20
Question 4:	/20	Question 12:	/20
Question 5:	/25		
Question 6:	/20		
Question 7:	/20		
Question 8:	/20		

Exam Total: ____

CSE2 Final Exam Fall 2013 Name_____

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

1) Write a sequence of four Unix commands to perform the following:

- Change the working directory to your home directory.
- Create a sub-directory of your home directory called “final_exam”.
- Copy a file in your home directory called “studyGuide” into the final_exam sub-directory that you just created.
- Display the contents of the final_exam sub-directory.

(5 points each, 20 points total)

Your answer:

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

1) What is the final value of x after each snippet of code? If x is undefined, write "undefined". (5 points each, 20 points total)

```
///snippet 1
int x = 2;
int y = 7 * x;
x *= y;
y = x;
```

Your answer:

```
///snippet 2
double w = 1.5, x = 3.0, y = 4.8;
w = x + y;
x = y / z;
y = z * w;
z = w - x;
x = z;
```

Your answer:

```
//snippet 3
int x = 2;
double y = 3.4;
y *= x;
x += (int) y;
```

Your answer:

```
//snippet 4
String w = "the", x = "quick", y = "white", z = "rabbit";
y = y + " " + z;
w = w + " " + y;
x = w + " " + x;
```

Your answer:

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

2) What is printed on the screen for each snippet of code? If this code triggers a compile-time error, write CE. (5 points each, 25 points total)

```
///snippet 1
int x = 5;
int z = 2;
System.out.println((--x)/z);
```

Your answer:

```
///snippet 2
double x = 6.4;
System.out.println((int) x/2);
```

Your answer:

```
//snippet 3
double y = 5.5F;
y -= 4;

System.out.println(y*2);
```

Your answer:

```
//snippet 4
int x = 3, y = 4, z = 1.5 ;
x += y * z;
System.out.println(x);
```

Your answer:

```
// snippet 5
final int x = 6;
int y = 2;
x += y;
System.out.println(x);
```

Your answer:

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

3) What is printed by the following lines of code? (20 points)

Your answer:

```
int myValue = 3;
switch( myValue ){
     case 1:
          myValue += 3;
          break;
     case 2:
          myValue /= 1;
          System.out.println("Good Morning " + myValue);
     case 3:
          myValue *= 2;
          System.out.println("Good Afternoon " + myValue);
     case 4:
          myValue -= 2;
          System.out.println("Good Evening " + myValue);
     case 5:
     case 6:
          myValue -= 2;
          System.out.println("Good Night " + myValue);
          break;
     default:
          System.out.println("default: " + myValue);
          break;
}
```

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

4) What is printed by the following lines of code? (20 points)

```
int myValue = 5;
boolean x = (myValue < 5);
boolean y = (myValue > 5);

if (!x&&y)
   System.out.println("C");
else
   System.out.println("B");

System.out.println(myValue == 4.0 ? "C" : "D");
```

Your answer:

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

5) Write a method that accepts no input and generates no output, except that it prints the output written below. When using loops, use only while loops. (25 points)

```
55555
55551
55511
55111
51111
4444
4442
4422
4222
333
333
333
22
24
1
```

Your code:

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

6) Write a method that takes an arbitrary array of integers as input, and creates a new array with the same values in reverse order. Call this method reverse(). (20 points)

Your answer:

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

7) Suppose you have a unsorted array of integers A and an integer k that is between 0 and A.length-1 (inclusive). Write a method called swapMax takes A and k as input, finds the largest value of A after k, and swaps that value for that of A[k]. For example, if k is 3, then swapMax will look for the largest value between array entries A[3] and A[A.length-1]. (20 points)

Your answer:

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

8) Given an array of ints A, use the methods swapMax() and reverse(), which you wrote above, to create a new method, sort(), that generates a sorted copy of A, via selection sort. The resulting array should be sorted from smallest to largest. You must use both swapMax() and reverse() for full credit. (20 points)

Your answer:

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

9) Given a ragged two-dimensional array of ints, A, write a method, sortArrays(), that uses the methods you've written above to create a new copy of A where each member array is sorted from largest to smallest, and order of the member arrays is sorted from largest to smallest. (20 points)

Your answer:

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

10) Convert all loops in the following code into **while** loops. Code outside the loops may stay the same, but your code should produce the same output. As long as you produce the same output, your while loop may have any structure. (10 points each, 20 points total)

```
// Question 10a
int x = -1;
do {
     x += 4;
     System.out.println( "val: " + x );
} while (x < 10);
```

Your answer:

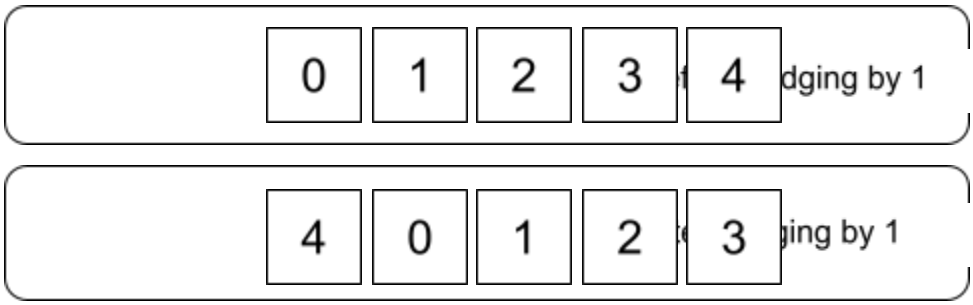
```
// Question 10b
for(int i = 0; i<4; i++) {
     for( int j = i; j<4; j++){
          System.out.print(i+j + " ");

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

Your answer:

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

11) Imagine that an integer array was circular. If you could nudge the values of the array over by k places, the value in A[0] would then be found in A[k], the value originally in A[1] would be found in A[k+1], etc. Values pushed beyond the length of the array would not be lost, they would wrap around to the beginning of the array: If nudge factor k was 1, then A[A.length-1] could then be found in A[0].
The figure below illustrates this point:



Implement circular nudging on int arrays by modifying the input (20 points)

Lehigh Username (use upper case letters): ☐☐☐☐☐☐@lehigh.edu

total score

12) On a 2D rectangular array A, circle nudging can work two ways: The first way is to nudge all the members of a given subarray. That is possible using the method circleNudge() that you wrote above. A second way is also possible, where, for a value b less than the length of any subarray, you nudge the b^{th} member of each subarray. That is, if you are nudging by k places, the value in A[0][b] can be found in A[k][b], the value in A[1][b] can be found in A[k+1][b], etc. These values would also wrap around to the beginning of the array: If k = 1, then A[A.length-1][b] would be found in A[0][b]. Implement this second kind of circle nudging in a method called twoDimensionalCircleNudge(). (20 points)