COS 126 **General Computer Science** **Spring 2017**

Written Exam 2

This exam has 9 questions (including question 0) worth a total of 70 points. You have 50 minutes. Write all answers inside the designated spaces.

Policies. The exam is closed book, except that you are allowed to use a one page cheatsheet (8.5-by-11 paper, two sides, in your own handwriting). No electronic devices are permitted.

Discussing this exam. Discussing the contents of this exam before solutions have been posted is a violation of the Honor Code.

This exam. Do not remove this exam from this room. Write your name, NetID, and the room in which you are taking the exam in the space below. Mark your precept number. Also, write and sign the Honor Code pledge. You may fill in this information now.

Name:

NetID:

Exam room:

Precept:

P01	P01A	P01B	P02	P02A	P03	P03A	P05	P06	P07	
◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	
P08	**P08A**	**P09**	**P11**	**P11A**	**P12**	**P13**	**P13A**	**P14**	**P14A**	**P15**
◯	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯

"I pledge my honor that I will not violate the Honor Code during this examination."

Signature

0	1	2	3	4	5	6	7	8	total

0. **Miscellaneous. (1 points)**

   (a) Write your name, NetID, and the room in which you are taking the exam in the space provided on the front of this exam.

   (b) Mark your precept number on the front of this exam.

   (c) Write and sign the Honor Code pledge on the front of this exam.

1. **Object-oriented programming. (8 points)**

   For each description on the left, choose the best-matching Java operator, method, or keyword on the right. Use each letter at most once.

_____ Checks whether two strings refer to the same memory address.	A. `class`
_____ Checks whether two strings refer to the same sequence of characters.	B. `final`
_____ Signifies a reference to the invoking object within an instance method.	C. `new`
_____ Signifies a reference to no object.	D. `null`
_____ Signifies a method that is not an instance method.	E. `private`
_____ Used to call an instance method of an object.	F. `public`
_____ Used to invoke a constructor.	G. `static`
_____ Helps enforce encapsulation	H. `this`
	I. `void`
	J. `.` operator
	K. `=` operator
	L. `==` operator
	M. `equals()` method

2. **Debugging. (10 points)**

Consider the following bug-infested implementation of a data type for computing a frequency table of integer values between `min` (inclusive) and `max` (exclusive).

`public class IntegerFrequencyTable`	
`public IntegerFrequencyTable(int min, int max)`	*frequency table of values in range* `[min, max)`
`public void increment(int val)`	*increment count corresponding to* `val`
`public int frequencyOf(int val)`	*return count corresponding to* `val`

Fix the program so that it works as intended. There is a twist—*you may only delete characters*; you may not add or rearrange code. Mark each line containing characters to delete.

bug

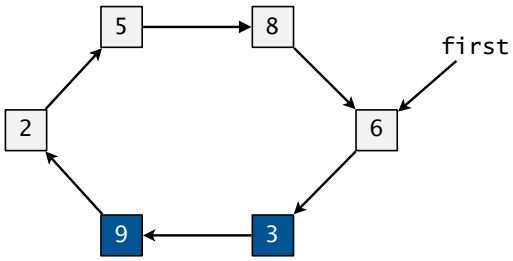
```
☑   public ~~salute the~~ class ~~of 2017~~ IntegerFrequencyTable {
☐       private int min;
☐       private int max;
☐       private int[] freq = new int[n];

☐       public void IntegerFrequencyTable(int min, int max) {
☐           int n = this.max - this.min;
☐           int[] freq = new int[n];
☐           this.min = min;
☐           this.max = max;
        }

☐       private static void validate(int val) {
☐           if (val <= min || val >= max)
☐               throw new IllegalArgumentException();
        }

☐       public void increment(int val) {
☐           validate(val);
☐           return freq[val - min]++;
        }

☐       public int frequencyOf(int val) {
☐           validate(int val);
☐           return freq[val - min];
        }
    }
```

3. **Linked structures. (7 points)**

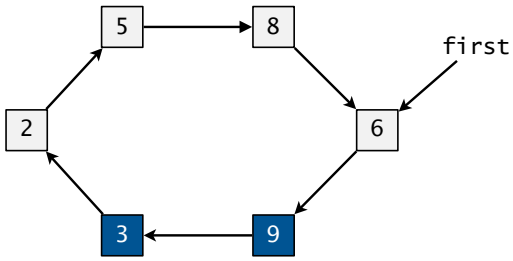
Suppose that the `Node` data type is defined as

```
private class Node {
   private int item;
   private Node next;
}
```

and that `first` is a variable of type `Node` that refers to the "first" node in a circularly linked list containing $n \geq 3$ nodes, as in the diagram at left:



before



after

Your goal is to exchange the order of the second and third nodes in the linked list.

To do so, complete the following implementation by filling in the letter of one of the expressions below in each provided space. You may use each letter any number of times. No other code is allowed.

A. `first`	C. `first.next`	E. `first.next.next`
B. `x`	D. `x.next`	F. `x.next.next`

```
Node x  =  _____ ;

_____   =  _____ ;

_____   =  _____ ;

_____   =  _____ ;
```

4. **Properties of sorting algorithms. (10 points)**

Suppose that you are sorting an array of n distinct items using the versions of insertion sort and mergesort from this course. For each statement on the left, determine whether it is a property of insertion sort, mergesort, neither, or both, by choosing the best-matching letter on the right.

_____ The number of compares is linearithmic in the worst case.

_____ If the input array is in ascending order, then the total number of compares is linear

_____ It uses only a constant amount of extra memory (besides the input array).

_____ Any pair of items is compared at most once.

_____ The number of compares depends only on the number of items in the array (and not on their values).

A. insertion sort only

B. mergesort only

C. both

D. neither

5. **TOY. (9 points)**

(a) What is the decimal representation of the 16-bit two's complement integer FFD8? Write your answer in the box.

(b) Suppose that `R[1]` contains `1234`. Which of the following TOY instructions will put `1234` into `R[2]`? Circle all that apply.

`1201` `2210` `3210` `3211` `4201`

(c) Suppose that `R[1]` contains `0001` and `M[01]` contains `FFFF`. Which of the following TOY instructions will put `FFFF` into `R[2]`? Circle all that apply.

`2201` `72FF` `7201` `8201` `A201`

TOY REFERENCE CARD

INSTRUCTION FORMATS

```
              | . . . . | . . . . | . . . . | . . . .|
  Format RR:  | opcode  |    d    |    s    |    t   |  (1-6, A-B)
  Format A:   | opcode  |    d    |       addr       |  (7-9, C-F)
```

ARITHMETIC and LOGICAL operations

```
    1: add              R[d] <- R[s] +  R[t]
    2: subtract         R[d] <- R[s] -  R[t]
    3: and              R[d] <- R[s] &  R[t]
    4: xor              R[d] <- R[s] ^  R[t]
    5: shift left       R[d] <- R[s] << R[t]
    6: shift right      R[d] <- R[s] >> R[t]
```

TRANSFER between registers and memory

```
    7: load address     R[d] <- addr
    8: load             R[d] <- M[addr]
    9: store            M[addr] <- R[d]
    A: load indirect    R[d] <- M[R[t]]
    B: store indirect   M[R[t]] <- R[d]
```

CONTROL

```
    0: halt             halt
    C: branch zero      if (R[d] == 0) PC <- addr
    D: branch positive  if (R[d] >  0) PC <- addr
    E: jump register    PC <- R[d]
    F: jump and link    R[d] <- PC; PC <- addr
```

Register 0 always reads 0.
Loads from M[FF] come from stdin.
Stores to M[FF] go to stdout.

16-bit registers (using two's complement arithmetic)
16-bit memory locations
 8-bit program counter

6. **Theory of computing. (8 points)**

You are in the final round of a job interview at a post-factual political-tech startup. Your interviewer asks you to classify various claims made by different companies as

- *old news* (known to be mathematically true)
- *alternative fact* (known to be mathematically false)
- *truthiness* (implies that $\mathcal{P} = \mathcal{NP}$)
- *fake news* (implies that the Church–Turing thesis is false)

For each claim, choose the best-matching letter on the right.

B ____ Adobe publishes $2 + 2 = 5$.

____ Twitter tweetstorms a poly-time algorithm for TSP.

____ Facebook posts that SAT is $\mathcal{NP}$-complete.

____ Tumblr blogs an exponential-time algorithm for SAT.

____ YouTube broadcasts that SORTING poly-time reduces to SAT.

____ Google advertises that SAT poly-time reduces to SORTING.

____ WikiLeaks reveals a formal language that can be described by some regular expression but cannot be recognized by any DFA.

____ Apple releases a poly-time algorithm that solves the halting problem. This revolutionary algorithm runs only on OS X.

____ SpaceX touts a physically realizable computing device that harnesses the power of black holes to solve the halting problem.

A. True (*old news*)

B. False (*alternative fact*)

C. Implies that $\mathcal{P} = \mathcal{NP}$ (*truthiness*)

D. Implies that the Church-Turing thesis is false (*fake news*)

7. **Powers of 2. (7 points)**

For each description on the left, choose the best-matching power of 2 on the right. You may use each letter any number of times.

_____ Number of 1s in the binary representation of $2^{32} - 1$.

_____ Number of distinct negative values representable in a TOY register.

_____ Multiplicative factor by which the running time increases when you double the size of n in the following code fragment:

```
int count = 0;
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        for (int k = 0; k < n; k++)
            count++;
```

_____ Minimum height of a binary search tree with 256 nodes. (Recall, the *height* of a tree is the maximum number of links on any path from the root node to a leaf node.)

_____ Number of strings in the language described by the regular expression

```
(A|C|G|T)(ACGT)(A|C)(G|T)
```

_____ Number of multiway AND gates in the sum-of-products representation of the 32-bit odd-parity function. (Recall, the odd-parity function function is 1 if and only if an odd number of its 32 inputs are 1.)

_____ Length of the string `s` after executing the following code fragment:

```
String s = "A";
for (int i = 0; i < 32; i++)
    s = s + s;
```

A. 2^0

B. 2^1

C. 2^2

D. 2^3

E. 2^4

F. 2^5

G. 2^6

H. 2^8

I. 2^{10}

J. 2^{12}

K. 2^{15}

L. 2^{16}

M. 2^{31}

N. 2^{32}

O. 2^{64}

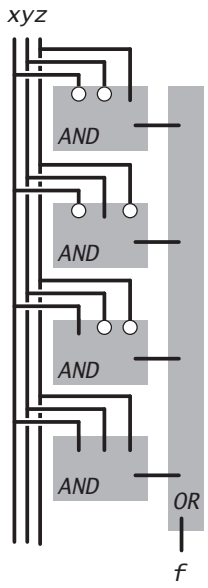
8. **Circuits. (10 points)**

Consider the boolean function of 3 variables $f(x, y, z) = x \wedge y \wedge z$. (Recall, $\wedge$ denotes the *xor* operator.) Which of the following represents the function f? Mark all that apply.

yes ● *no* ○

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

yes ○ *no* ○



yes ○ *no* ○

yes ○ *no* ○

$f = x \wedge ((x \wedge y) \wedge (x \wedge z))$

yes ○ *no* ○

$f = xyz + xy'z' + x'yz' + x'y'z$

yes ○ *no* ○

```
public static boolean f(boolean x, boolean y, boolean z) {
    if (x && y) return z;
    if (x || y) return !z;
    return !z;
}
```

This page is provided as scratch paper. If you tear it out, please write your name, NetID, and precept number in the space provided and return it inside your exam.

Name: _____ **NetID:** _____ **Precept:** _____