COS 126	Princeton University	Fall 2025

Written Exam 2

This exam has 8 questions worth a total of 72 points. You have 80 minutes.

Instructions. This exam is preprocessed by computer. Write neatly, legibly, and darkly. Put all answers (and nothing else) inside the designated spaces. *Fill in* bubbles completely, like this: ● . To change an answer, erase it completely and redo. Do not remove or unstaple any pages. You may use the last page for scratch work, as well as margins, provided it does not interfere with boxes and bubbles for answers. Additional scratch paper is available upon request.

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

Honor Code. This exam is governed by Princeton's Honor Code. Discussing the contents of this exam before solutions are posted is a violation of the Honor Code.

Please complete the following information now.

Name:

NetID:

Exam room: ◯ McCosh 46 ◯ McCosh 50 ◯ Other

Precept:

P01	P01A	P02	P03	P03A	P04	P04A	P05	P05A
◯	◯	◯	◯	◯	◯	◯	◯	◯

P06	P10	P10A	P10B	P11	P11A	P12
◯	◯	◯	◯	◯	◯	◯

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

Signature

1. **Initialization (1 point)**

   In the designated spaces on the front of this exam,

   - *Print* your name.
   - *Print* your Princeton NetID (6–8 characters, like "`cs0126`").
   - *Fill in* the bubble corresponding to the room in which you are taking this exam.
   - *Fill in* the bubble corresponding to your precept.
   - *Write* and *sign* the Honor Code pledge.

2. **Object-Oriented Programming (10 points)**

For each description below, write the **letter** corresponding to the best matching Java term. Each letter may be used at most once; not all answers will be used.

Answer	Description
☐	A method that belongs to the class rather than to any instance of that class.
☐	A method invoked on a specific object that can read or update instance variables of that object.
☐	A special method used to initialize a newly created object; it has no return type and shares the class name.
☐	A design principle in which an object hides its internal representation and exposes only a public API.
☐	A data type whose externally visible value never changes once the object is constructed.
☐	The keyword used inside an instance method or constructor to refer to the current object.
☐	A variable whose lifetime lasts only during a single method call.
☐	A practice in Java which allows methods to have the same name as long as they have different parameter lists.
☐	A variable that cannot be reassigned after initialization.
☐	A method defined in object classes that converts an object into a human-readable string.

Possible Answers:

A. method overloading
B. static method
C. primitive type
D. `toString` method
E. reference type
F. encapsulation
G. final variable
H. abstract data type
I. immutable data type
J. `print` method
K. constructor
L. instance method
M. local variable
N. `this`
O. static variable
P. instance variable
Q. `self`

3. **Data types (12 points)**

Consider the following Java class:

```
public class DNA {
    private final String name;
    private final String seq;

    public DNA(String name, String seq) {
        this.name = name;
        this.seq = seq;
    }

    public String sequence() {
        return seq;
    }

    public int count(String pat) {
        int count = 0;
        for (int i = 0; i + pat.length() <= seq.length(); i++) {
            if (seq.substring(i, i + pat.length()).equals(pat))
                count++;
        }
        return count;
    }

    public DNA reverseComplement() {
        char[] rc = new char[seq.length()];
        for (int i = 0; i < seq.length(); i++) {
            switch (seq.charAt(seq.length() - 1 - i)) {
                case 'A': rc[i] = 'T'; break;
                case 'T': rc[i] = 'A'; break;
                case 'C': rc[i] = 'G'; break;
                case 'G': rc[i] = 'C'; break;
            }
        }
        return new DNA(name + "_rc", new String(rc));
    }

    public boolean find(DNA pattern) {
        return this.seq.contains(pattern.seq)
            || this.reverseComplement().seq.contains(pattern.seq);
    }

    public boolean find(String pattern) {
        return this.seq.contains(pattern);
    }

    public boolean equals(DNA other) {
        return this.name.equals(other.name) && this.seq.equals(other.seq);
    }
}
```

The program compiles, but there is at least one poor design or implementation choice in this class, so be careful when reasoning about the behavior of the methods. (See next page.)

Assume the following variables have already been declared and initialized:

```
DNA a = new DNA("geneA", "AATG");
DNA b = new DNA("geneB", "CAAATT");
DNA c = new DNA("geneC", "ATGCTAACGGTATTCGACCTTAGGCACTTG");
DNA d = new DNA("geneD", "TG");
```

For each Java expression below, write the **letter** corresponding to its value from the list of possible answers. Each letter may be used once, more than once or not at all.

Answer	**Expression**
☐	`a.count("A")`
☐	`b.count("AA")`
☐	`b.count("TG")`
☐	`b.find("TG")`
☐	`b.find(d)`
☐	`a.reverseComplement().sequence()`
☐	`b.reverseComplement().count("AT")`
☐	`a.equals(b)`
☐	`b.count('AA')`
☐	`b.find(reverseComplement(d))`
☐	`c.count("T") == c.reverseComplement().count("A")`
☐	`c.equals(c.reverseComplement().reverseComplement())`

Possible Answers:

A. `1`

B. `true`

C. `0`

D. `GTAA`

E. `2`

F. `AATG`

G. `TTAC`

H. `false`

I. `3`

J. `CATT`

K. *Error*

L. `4`

TOY reference

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`

Registers and memory

- 16 16-bit registers: `R[0]` to `R[F]`
- 256 16-bit memory locations: `M[00]` to `M[FF]`
- 1 8-bit program counter: `PC`
- `R[0]` always reads as `0000`.
- Loads from `M[FF]` come from `stdin`.
- Stores to `M[FF]` go to `stdout`.

4. **TOY machine (10 points)**

Set the program counter to 10 and run the following TOY program. Then answer the questions below. Fill in the bubble of the correct answer. Choose one answer (unless specified otherwise).

```
10:  7101
11:  8213
12:  831A
13:  7400
14:  1442
15:  2331
16:  94FF
17:  D314
18:  941A
19:  0000
1A:  0003
```

(a) After executing the instruction at address `13`, what is the value in register `R[3]`?

◯ `0000` ◯ `0001` ◯ `0003` ◯ `0013`
◯ `7400` ◯ `7401` ◯ `1442` ◯ `FFFF`

(b) How many 16-bit words are written to stdout by this program before it halts?

◯ 0 ◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ 6 ◯ 7

(c) What is the value in register `R[3]` immediately after executing the instruction at address `18`?

◯ `0000` ◯ `0001` ◯ `0003` ◯ `0004`
◯ `73FF` ◯ `7400` ◯ `5C00` ◯ `FFFF`

(d) What is the value stored in memory location `M[1A]` after the program has finished executing?

◯ `0000` ◯ `7400` ◯ `5C00` ◯ `5C03`
◯ `0003` ◯ `7403` ◯ `1A00` ◯ `15C0`

(e) Suppose TOY was revised so that the 0 opcode (assuming format A for the instruction) was defined as:

```
0:  newhalt  if (R[d] == 0) halt else PC <- addr
```

Which of the following instructions could replace the instruction at address `19` without changing the behavior of the program?
Choose **all answers that apply**:

◯ `0000` ◯ `00FF` ◯ `0116` ◯ `0110`

5. **Algorithmic performance (9 points)**

For the following algorithms, specify the order-of-growth of running time relative to n. Write in the corresponding box to the left the **letter** of the best-matching term from the right column. Each letter may be used once, more than once, or not at all.

Answer	Description
☐	Worst-case running time of **binary search** on a sorted array of size n.
☐	Worst-case running time of **sequential search** on an array of size n.
☐	Worst-case running time of **insertion sort** on an array of size n.
☐	Best-case running time of **insertion sort** on an array of size n.
☐	Worst-case running time of `pop()` operation for **a stack** `Stack<Item>` of size n.
☐	Worst-case running time of `contains()` operation for **a symbol table** `ST<Key, Value>` of size n.
☐	Recursive solution to **Towers of Hanoi** of size n.
☐	Worst-case running time of **merge sort** on an array of size n.
☐	Running time of **a DFA** with 126 states (implemented by a Java program) on an input string of size n.

A. $\Theta(1)$ *constant*

B. $\Theta(\log n)$ *logarithmic*

C. $\Theta(n)$ *linear*

D. $\Theta(n \log n)$ *linearithmic*

E. $\Theta(n^2)$ *quadratic*

F. $\Theta(n^2 \log n)$ *quadrithmic*

G. $\Theta(n^3)$ *cubic*

H. $\Theta(2^n)$ *exponential*

6. **Number systems (10 points)**

(a) Convert the decimal number 93_{10} to binary. Choose one answer:

◯ 1101110 ◯ 1001011 ◯ 1110110 ◯ 1011101 ◯ 1011011 ◯ 111011

(b) Convert the hexadecimal number $\texttt{3AF}_{16}$ to decimal. Choose one answer:

◯ 799 ◯ 912 ◯ 943 ◯ 1023 ◯ 878 ◯ 1007

(c) Convert the decimal number 255_{10} to hexadecimal. Choose one answer:

◯ `A0` ◯ `FE` ◯ `FF` ◯ `F0` ◯ `EF` ◯ `1FF`

(d) Compute in binary: $101101_2 + 11011_2$. Choose one answer:

◯ 111000 ◯ 1001000 ◯ 1100110 ◯ 1010101 ◯ 1001100 ◯ 1000111

(e) In 8-bit two's complement, what is the decimal value of 11110110_2? Choose one answer:

◯ −11 ◯ −10 ◯ −9 ◯ −6 ◯ 118 ◯ 246 ◯ 247

(f) Convert base-5 number 324_5 to decimal. Choose one answer:

◯ 69 ◯ 84 ◯ 89 ◯ 94 ◯ 99 ◯ 101

(g) Which binary numbers below are odd?

Choose **all answers that apply**:

◯ 101010 ◯ 111011 ◯ 110110 ◯ 100000 ◯ 110000 ◯ 101100

(h) Which of the following numbers has a binary representation ending in `000`?

Choose **all answers that apply**:

◯ 16 ◯ 18 ◯ 20 ◯ 21 ◯ 22 ◯ 24 ◯ 28

(i) In Java, an integer literal that begins with `0x` is written in hexadecimal.

Let `int x = 0x25;` and `int y = 0x1C;`. What is the decimal value of `x + y`?

Choose one answer:

◯ 47 ◯ 56 ◯ 65 ◯ 75 ◯ 81 ◯ 93

(j) In Java, an integer literal that begins with `0x` is written in hexadecimal.

Let `int p = 0x3C;` and `int q = 0x55;`. What is the value of `p ^ q` in hexadecimal? Choose one answer:

◯ `0x14` ◯ `0x69` ◯ `0x6C` ◯ `0x78` ◯ `0x7D` ◯ `0x91`

7. **Boolean logic (10 points)**

(a) For each of the following statements from Boolean algebra using Boolean variables x and y, fill in the bubble corresponding to *true* or *false*.

true	*false*	
◯	◯	$x + x'y$ is equivalent to $x + y$
◯	◯	$xy + x$ is equivalent to x
◯	◯	$(x + y)'$ is equivalent to $x'y'$
◯	◯	$xy + x'y'$ is equivalent to $xy' + x'y$

(b) Consider the Boolean function: $f(x, y) = xy' + x'y$. Which of the following expressions are equivalent to $f(x, y)$? Choose **all answers that apply**.

- ◯ $x'y' + xy$
- ◯ $(x + y)(x' + y')$
- ◯ $x + y$
- ◯ xy
- ◯ $x'y'$

(c) Let w, x, y, z be Boolean variables, and define the majority function $\text{MAJ}_4(w, x, y, z) = 1$ if and only if **at least three** of the four inputs are 1.

Which of the following Boolean expressions correctly computes $\text{MAJ}_4(w, x, y, z)$? Choose **all answers that apply**.

- ◯ $(w + x)(y + z)$
- ◯ $wx + wy + wz + xy + xz + yz$
- ◯ $wxyz' + wxy'z + wx'yz + w'xyz + wxyz$
- ◯ $wxy + wxz + wyz + xyz$
- ◯ $wxy + wxz + wyz$

(d) Let $f(x, y, z)$ be defined by the following truth table:

x	y	z	$f(x, y, z)$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Which of the following Boolean expressions is a correct representation of $f(x, y, z)$? Choose **all answers that apply**.

- ◯ $x'y'z + x'yz' + xy'z'$
- ◯ $x'y'z' + x'yz' + xy'z$
- ◯ $xyz + xy'z + x'yz + x'yz'$
- ◯ $x'y'z' + x'yz' + xy'z' + xy'z$
- ◯ $x'y'z' + xy'z' + xyz'$
- ◯ $x'y'z' + x'yz' + xy'z' + xy'z + xyz'$

8. **Miscellaneous (10 points)**

(a) For each of the following statements, fill in the bubble corresponding to *true* or *false*:

true	*false*	
◯	◯	A host name always corresponds to a single IP (Internet Protocol) address.
◯	◯	A deterministic finite-state automaton can simulate any Turing machine.
◯	◯	An image classifier program can, in principle, be implemented using a Turing machine.
◯	◯	A generative AI system built using a large language model can determine, when provided a Java program as input, whether this program will halt.
◯	◯	A deterministic finite-state automaton can implement a search for the subsequence `ATG` inside a gene sequence.
◯	◯	The Internet provides best-effort packet delivery, meaning packets may be lost, corrupted, or delivered out of order.
◯	◯	Sequence alignment algorithms are used to measure similarity between DNA sequences.
◯	◯	Image multiclass classification using perceptrons is an example of an unsupervised machine learning problem.

(b) In a correct implementation of the Chat126 programming assignment, suppose the Markov model has order $k = 0$. Which statements are true? Choose **all answers that apply**.

◯ Chat126 generates random characters using any characters from the ASCII character set.

◯ Chat126 may generate any character that appears at least once in the original text.

◯ Chat126 throws an exception when asked for a k-gram of length 0.

◯ The order-of-growth of running time for Chat126 will be greater than for $k > 0$.

This page is intentionally left blank. You may use it for scratch work.