

COS 126 Written Exam 1 Fall 2019

There are eight questions on this exam. There is one question per lecture, numbered corresponding to the lectures, *not in order of difficulty*. If a question seems difficult to you, skip it and come back to it. You will have 50 minutes to complete the exam.

This exam is preprocessed by computer. If you use pencil (and eraser), write darkly. Write all answers inside the designated rectangles. When asked to fill in circles, do so completely with a dark pencil. If you change your mind, you must erase completely and fill in another circle!

Do this ● not this ✓ or this ✗ or this ✗.

Resources. You may reference your optional one-sided 8.5-by-11 handwritten "cheat sheet" during this exam. You may not use the textbook, your notes, or any electronic devices. You may not communicate with anyone except the course staff during this exam.

Discussing this exam. Due to travel for extracurriculars and sports, some of your peers will take this exam later. Do not discuss its contents with anyone who has not taken it.

This page. Do not remove this exam from the exam room. Fill in this page now, but do not start the exam until you are told.

Name

NetID

Precept

Exam Room

"I pledge my honor that I have not violated the Honor Code during this examination."

[copy the pledge here]

[signature]

Q1a. Types and Casts (1 point). In each row, fill in the circle corresponding to the value of the given expression. You must fill in exactly one circle in each row.

| | 0 | 1 | 0.0 | 1.0 | NaN | <i>won't compile</i> |
|--------------------------|----------------------------------|-----------------------|----------------------------------|----------------------------------|-----------------------|----------------------------------|
| $3 - (\text{int}) "2.0"$ | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
| $4 / 6 * 1.5$ | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| $1 / 1 - 1 / 1$ | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| $1.5 * 4 / 6$ | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Q1b. Terminology (1.5 points). Put the letter to left of each definition that identifies the term defined. No letter may be used more than once.

| | | | |
|--|---|----------|--------------------------|
| <input checked="" type="checkbox"/> D | Rules that determine in which order to apply operations | A | array |
| <input checked="" type="checkbox"/> F | Step-by-step description of the operation of a program | B | data type |
| <input checked="" type="checkbox"/> E | A finite sequence of characters | C | literal |
| <input checked="" type="checkbox"/> A | Data structure that holds a sequence of values of the same type | D | precedence |
| <input checked="" type="checkbox"/> C | Source-code representation of a data-type value | E | string |
| <input checked="" type="checkbox"/> B | A set of values and operations on those values | F | trace |
| | | G | variable |
| | | H | <i>none of the above</i> |

Q2. Loops and conditionals (2 points). Suppose that this code is in the file `Q2.java`

```
public class Q2
{
    public static void main(String[] args)
    {
        int N = Integer.parseInt(args[0]);
        for (int j = -N; j <= N; j++)
        {
            for (int i = -N; i <= N; i++)
            {
                if (i == j) System.out.print("A");
                else if (i == -j) System.out.print("B");
                else System.out.print("C");
            }
        }
        System.out.println();
    }
}
```

and that you compile it using the command `javac-introcs Q2`. Fill in the circle corresponding to the specified character. You must fill in exactly one circle in each row.

| | A | B | C |
|---|----------------------------------|-----------------------|-----------------------|
| first character printed by <code>java-introcs Q2 0</code> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> |

N = 0, so the first time through the loops, i and j both equal 0. According to the if/else statements that means we print "A".

| | | | |
|--|-----------------------|-----------------------|----------------------------------|
| second character printed by <code>java-introcs Q2 1</code> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |
|--|-----------------------|-----------------------|----------------------------------|

When N = 1, at first j = -1, i = -1. So the first character printed is "A". Then i increments to 0, and j remains at -1. So the second character printed is "C".

| | | | |
|---|-----------------------|----------------------------------|-----------------------|
| ninth character printed by <code>java-introcs Q2 2</code> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> |
|---|-----------------------|----------------------------------|-----------------------|

Let's trace these loops. The values of (j, i) change in the following order: (-2, -2), (-2, -1), (-2, 0), (-2, 1), (-2, 2), (-1, -2), (-1, -1), (-1, 0), (-1, 1), (-1, 2), and so on. The ninth pair in this sequence is (j, i) = (-1, 1), so the ninth character printed is "B".

| | | | |
|--|----------------------------------|-----------------------|-----------------------|
| twenty-fifth character printed by <code>java-introcs Q2 3</code> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> |
|--|----------------------------------|-----------------------|-----------------------|

Like in the previous problem, let's trace how the values of (j, i) change. You could list the first 25 values of (j, i), but if you're short on time there's a shortcut. Note that in every pass through the outer for loop, 7 new characters are printed. Because for each value of j, the inner for loop executes 7 times, as i increments from -3 to 3. So after 3 passes through the outer for loop, 21 characters have been printed. At this point, j has gone through the values -3, then -2, then -1, and now j increments to 0 to begin the fourth pass through the outer for loop. The 22nd value of (j, i) = (0, -3). The 23rd (j, i) = (0, -2). The 24th (j, i) = (0, -1). And the 25th (j, i) = (0, 0). So the 25th character is "A".

Q3a. Arrays (1 point). Consider the following Java program

```
public class Q3a
{
    public static void main(String[] args)
    {
        int[] a = new int[6];
        int[] b = new int[a.length];
        b = a;
        for (int i = 1; i < b.length; i++)
            b[i] = i;
        for (int i = 0; i < a.length; i++)
            System.out.print(a[i] + " ");
        System.out.println();
    }
}
```

Fill in the *one* circle at right that corresponds to the output of this program

- 0 0 0 0 0 0
- 0 1 2 3 4 5
- cannot be determined from the information given*

The key to this question is the line "b = a;". At first, a and b refer to two different arrays. That's to say a and b are references that point to two arrays at different memory locations. But "b = a;" makes the reference b point to the same array that a does. Now a and b are aliases of the same array. Next, when we set the values of b's array to {0, 1, 2, 3, 4, 5}, we're changing the array that a and b both point to. So printing the entries of a gives "0 1 2 3 4 5".

Q3b. Two-dimensional arrays (1 point). Put the letter to left of each definition that describes the given code. No letter may be used more than once.

Think of array a as a 1D array where every element of array a is a 1D array. The number of rows in the 2D array is the number of elements in array a (a.length). For more information, see the "Arrays" lecture slides, slide 36.

- | | | | |
|----------|---|----------|--------------------------|
| B | Refers to the number of rows | A | double[][] a; |
| | <i>Think of the 2D array as a 1D array, where each entry is a row. Then a[i] gives a particular row in the array.</i> | B | a.length |
| F | Refers to a specified row | C | a.row(i) |
| | | D | a[i][j] |
| A | Declares a two-dimensional array | E | a = new double[2][2]; |
| E | Creates a two-dimensional array | F | a[i] |
| | | H | <i>none of the above</i> |

Q4. I/O (2 points). Carefully study the following Java program. It is simple, but there is a likely bug that makes it tricky.

```
public class Q4
{
    public static void main(String[] args)
    {
        int x = StdIn.readInt();
        int y = Integer.parseInt(args[0]);
        StdOut.print(x + y + " ");
        StdOut.print(x + " ");
        StdOut.println(y);
    }
}
```

Suppose that the file `input.txt` contains the string "6 5 4 3". In the box to the left of each command, write the letter corresponding to the output it produces. A letter may be used once, more than once, or not at all.

F

`java-introcs Q4 3`

A 6 2 6

D

`java-introcs Q4 3 < input.txt`

B 11 9 2

F

`java-introcs Q4 3 > input.txt`

C 6 5 4 3

B

`java-introcs Q4 3 < input.txt | java-introcs Q4 2`

D 9 6 3

E 6 3 6 3

F *none of the above*

Q5. Functions (2 points). Consider the following code.

```
public class Cubes
{
    public static int square(int i)
    { return i * i * i; }

    public static void main(String[] args)
    {
        for (int i = 1; i <= 1000; i++)
            StdOut.println(square(i));
    }
}
```

Fill in the circle to the right of each statement to indicate whether it is true or false.

- | | <i>true</i> | <i>false</i> |
|--|----------------------------------|----------------------------------|
| Will not compile because braces in square() are not on separate lines. | <input type="radio"/> | <input checked="" type="radio"/> |
| Will not compile because braces are missing in the for loop. | <input type="radio"/> | <input checked="" type="radio"/> |
| Will not compile because i is not declared in square(). | <input type="radio"/> | <input checked="" type="radio"/> |
| Prints the squares of the integers from 1 to 1000. | <input type="radio"/> | <input checked="" type="radio"/> |
| Prints the cubes of the integers from 1 to 1000. | <input checked="" type="radio"/> | <input type="radio"/> |
| Goes into an infinite loop. | <input type="radio"/> | <input checked="" type="radio"/> |
| Prints only a few lines because square() rapidly increases the i used by main(). | <input type="radio"/> | <input checked="" type="radio"/> |
| Causes a runtime error because of a type mismatch | <input type="radio"/> | <input checked="" type="radio"/> |

Q6. Recursion (2.5 points). Suppose that this code is in the file Q6.java

```
public class Q6
{
    public static int f(int x)
    {
        if (x < 2) return x;
        if (x % 2 == 1) // x is odd
            return f((x - 1) / 2) - 1;
        else // x is even
            return f(x / 2) + 1;
    }

    public static void main(String[] args)
    {
        int n = Integer.parseInt(args[0]);
        StdOut.println(f(n));
    }
}
```

Now suppose that you compile this program using the command `javac-introcs Q6` and then run it using the command `java-introcs Q6 x` for the various values of `x` given at left below. Fill in the circle corresponding to the value printed for each given value of `x`. You must fill in exactly one circle in each row.

| | -4 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----|----------------------------------|-----------------------|-----------------------|----------------------------------|----------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|
| 0 | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 3 | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 5 | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 63 | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 64 | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> |

TOY REFERENCE CARD

INSTRUCTION FORMATS

| | | | | | |
|------------|---------|-----------|-----------|-----------|------------|
| Format RR: | | . . d . . | . . s . . | . . t . . | (0-6, A-B) |
| Format A: | opcode | d | addr | t | (7-9, C-F) |

ARITHMETIC and LOGICAL operations

| | |
|----------------|------------------------------------|
| 1: add | $R[d] \leftarrow R[s] + R[t]$ |
| 2: subtract | $R[d] \leftarrow R[s] - R[t]$ |
| 3: and | $R[d] \leftarrow R[s] \& R[t]$ |
| 4: xor | $R[d] \leftarrow R[s] \wedge R[t]$ |
| 5: shift left | $R[d] \leftarrow R[s] \ll R[t]$ |
| 6: shift right | $R[d] \leftarrow R[s] \gg R[t]$ |

TRANSFER between registers and memory

| | |
|-------------------|----------------------------------|
| 7: load address | $R[d] \leftarrow \text{addr}$ |
| 8: load | $R[d] \leftarrow M[\text{addr}]$ |
| 9: store | $M[\text{addr}] \leftarrow R[d]$ |
| A: load indirect | $R[d] \leftarrow M[R[t]]$ |
| B: store indirect | $M[R[t]] \leftarrow R[d]$ |

CONTROL

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

Q7. TOY (2 points). In the box to the left of each description, write the letter corresponding to the four-digit hex number that best fits the description. A letter may be used once, more than once, or not at all.

A

Represents the decimal integer 20

J

Represents the decimal integer -20

**A/
B**

Halt instruction

E

Doubles the value in a register

H

Writes to standard output

K

Illegal TOY instruction

E

Represents the decimal integer 4369

**C/
D/
I**

No-op (has no effect)

A 0014

B 0020

C 1020

D 1110

E 1111

F 1212

G 4369

H 9FFF

I D099

J FFEC

K *none of the above*

Q8. TOY Programming (2.5 points). An integer array is stored starting at memory location 51. The length of the array is stored in memory location 50. The following (partial) TOY program reverses the array by swapping the first element and the last element, then swapping the second element and the next-to-last element, and so on until the *lo* and *hi* pointers cross.

```

10: 7101  R[1] <- 01           R[1] gets constant 1
11: _____ R[2] <- 51       addr of first element (lo)
12: _____ R[3] <- M[50]
13: 1323  R[3] <- R[2] + R[3]
14: 2331  R[3] <- R[3] - R[1]   addr of last element (hi)
15: _____
16: D41E  if (R[4] > 0) pc <- 1E  15-16 exit loop if hi < lo
17: _____ R[5] <- M[R[2]]
18: A603  R[6] <- M[R[3]]
19: B503  M[R[3]] <- R[5]
1A: B602  M[R[2]] <- R[6]
1B: 1221  R[2] <- R[2] + R[1]   increment lo pointer
1C: 2331  R[3] <- R[3] - R[1]   decrement hi pointer
1D: _____
1E: 0000  halt

```

For each of the missing instructions, write the letter corresponding to the correct instruction in the box to the left of the given address.

- | | | | |
|-----|----------|----------|------|
| | | A | 2423 |
| 11: | C | B | 2432 |
| 12: | G | C | 7251 |
| 15: | A | D | 7350 |
| 17: | H | E | 8250 |
| 1D: | J | F | 8251 |
| | | G | 8350 |
| | | H | A502 |
| | | I | B502 |
| | | J | C015 |