Instructions. This exam has six (6) questions worth a total of one hundred (100) points. You have eighty (80) minutes.

This exam is preprocessed by computer. Write neatly, legibly and darkly. If you use a pencil, write darkly. Put all answers (and nothing else) inside the designated boxes. Fill in bubbles and checkboxes completely: ⚫ and ⬛ (not ✔ or ✘). To change an answer, erase it completely and redo.

Resources. The exam is closed book, except that you are allowed to use a single one-sided reference sheet (8.5-by-11 paper, one-sided, 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 have been posted is a violation of the Honor Code.

NAME:

NETID

PRECEPT

P01	P02	P02A	P03	P04	P05	P06	P07
◯	◯	◯	◯	◯	◯	◯	◯
P08	**P08A**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**
◯	◯	◯	◯	◯	◯	◯	◯

EXAM ROOM McCosh 50 ◯ McCosh 10 ◯ McCosh 62 ◯ OTHER _____

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

_____

Signature

Question 1 **Java Data Values and Types** **24** points

Give the value and type of each of the following expressions. To express a value, write a Java literal of the appropriate type, such as **`0`** (for an **`int`**), **`3.14`** (for a **`double`**), **`false`** (for a **`boolean`**), **`"tiger"`** (for a **`String`**), **`'a'`** (for a **`char`**). If the expression does not compile or causes a runtime exception, put an **`X`** in both boxes.

Expression	Value	Type
`8 / 10 * 1.5`		
`12 % 7`		
`2 + 3 * 4`		
`2 + 3.0 * 4`		
`8 - (int) "2.0"`		
`1 / 1 / 0`		
`1.0 / 1 / 0`		
`true || false && true`		
`true && !(0 < -5)`		
`(!!false || !!true)`		
`1 < 2 < 3`		
`Math.max(1.0, 2.0, 3.0)`		

Question 2	**Debugging**	**15** points

```
 1 public class BuggyCode {
 2     // Prints the absolute value of the sum of all even elements
 3     public static void main(String[] args) {
 4         int N = args[0];
 5         int sum;
 6         for (i = 0; i < N; i++) {
 7             int val = StdIn.readInt();
 8             if (val % 2)
 9                 sum += val;
10         }
11         if (N < 0)
12             StdOut.println(-sum);
13         else
14             StdOut.println(sum);
15     }
16 }
```

The compiler reports an error on lines 4, 6, and 8 for the **BuggyCode** class. Fix them by writing each line corrected in the natural way. Write your answer in the boxes below:

Line 4:

Line 6:

Line 8:

After fixing these errors, the compiler reports a ***variable might not have been initialized*** error on lines 9, 12, and 14. Fill in the bubbles corresponding to the line(s) would you change to fix this error?

1	2	3	4	5	6	7	8
◯	◯	◯	◯	◯	◯	◯	◯

9	10	11	12	13	14	15	16
◯	◯	◯	◯	◯	◯	◯	◯

Question 3	**Tracing**	**18** points

```
public class Scatmain {
    public static void main(String[] args) {
        String[] x = { "Ski-", "Bwi-", "Ba-", "Bop-", "Dop-", "Bop\n", "Yeah!\n", "Bada-" };
        int[] y = { 0, 2, 3, 2, 4, 5, -1, x.length - 2 };
        String z = x[2] + x[2] + x[x.length - 1] + "Dop";
        for (int i = 0; i < x.length; i++) {
            if (y[i] < 0) {
                StdOut.print(x[7] + x[y[1] - 1]);
                for (int j = 0; j < 3; j++)
                    StdOut.println(z);
            }
            else
                StdOut.print(x[y[i]]);
        }
    }
}
```

This program produces between 1 and 7 lines of output. What does it print? Write the letter corresponding to each line's text in the box. Use only one letter per box. Every box needs a letter. You may use each letter once, multiple times, or not at all.

Line 1: ☐

Line 2: ☐

Line 3: ☐

Line 4: ☐

Line 5: ☐

Line 6: ☐

Line 7: ☐

A	<blank> or not a line the program prints.
B	**Bwi-Bwi-Bada-Dop**
C	**Ba-Ba-Bada-Dop**
D	**Yeah!**
E	**Ba-Ba-Yeah!**
F	**Dop**
G	**Bwi-Bwi-Yeah!**
H	**Ski-Ba-Bop-Ba-Dop-Bop**
I	**Bop**
J	**Bada-Bwi**
K	**Bada-Bwi-Ba-Ba-Bada-Dop**
L	**Ski-Bwi-Ba-Bwi-Bop-Dop**
M	**I'm the Scatman!**

Question 4	**Java Variable Scope**	**16** points

Consider the following code fragment. The labeled dotted boxes represent regions where additional code exists.

```
1.   public static void func(int i) {
        A
2.      int j = i;
        B
     }
     public static void main(String[] args) {
        C
3.      for (int i = 0; i < n; i++) {
           D
4.         for (int j = 0; i < n; i++) {
              E
              func(j);
              F
           }
           G
        }
        H
     }
```

What is the valid scope for variable **i** in line number 1? Select the regions that apply.

A	B	C	D	E	F	G	H
◯	◯	◯	◯	◯	◯	◯	◯

What is the valid scope for variable **j** in line number 2? Select the regions that apply.

A	B	C	D	E	F	G	H
◯	◯	◯	◯	◯	◯	◯	◯

What is the valid scope for variable **i** in line number 3? Select the regions that apply.

A	B	C	D	E	F	G	H
◯	◯	◯	◯	◯	◯	◯	◯

What is the valid scope for variable **j** in line number 4? Select the regions that apply.

A	B	C	D	E	F	G	H
◯	◯	◯	◯	◯	◯	◯	◯

Question 5	**Number Systems**	**15** points

1. Express the decimal number **6** as 8-bit two's complement:

2. Express the decimal number **-9** as 8-bit two’s complement:

3. Convert **FACE** from hexadecimal to binary:

4. Convert **10110000** from 8-bit two's complement to decimal:

5. Convert **294** from decimal to hexadecimal:

Question 6	**Recursion**	**12** points

In this question, if you are not sure of an answer, select "Not Sure" to receive 1 point. Consider the following recursive function:

```
public static void mystery(int n, double x, double y, double size) {
    if (n <= 0) return;
    StdDraw.filledSquare(x, y, size / 6);
    double newSize = size / 3;
    mystery(n - 1, x - newSize, y, newSize);
    mystery(n - 1, x + newSize, y, newSize);
    mystery(n - 2, x, y + newSize, newSize);
}
```

What results from calling **`mystery(0, 0.5, 0.5, 1)`**? Select letter of image below, blank image, or other.

A	B	C	D	E	F	Blank	Other	Not Sure
◯	◯	◯	◯	◯	◯	◯	◯	◯

What results from calling **`mystery(3, 0.5, 0.5, 1)`**? Select letter of image below, blank image, or other.

A	B	C	D	E	F	Blank	Other	Not Sure
◯	◯	◯	◯	◯	◯	◯	◯	◯

What results from calling **`mystery(4, 0.5, 0.5, 1)`**? Select letter of image below, blank image, or other.

A	B	C	D	E	F	Blank	Other	Not Sure
◯	◯	◯	◯	◯	◯	◯	◯	◯