

**CIS 110 — Introduction to Computer Programming
Fall 2018 — Midterm**

Name: _____

Recitation #: _____

Pennkey (e.g., paulmcb): _____

My signature below certifies that I have complied with the University of Pennsylvania's Code of Academic Integrity in completing this examination.

Signature

Date

Instructions:

- **Do not open this exam until told by the proctor.** You will have exactly 120 minutes to finish it.
- **Make sure your phone is turned OFF (not to vibrate!) before the exam starts.**
- Food, gum, and drink are strictly forbidden.
- **You may not use your phone or open your bag for any reason**, including to retrieve or put away pens or pencils, until you have left the exam room.
- This exam is *closed-book, closed-notes, and closed-computational devices*.
- If you get stuck on a problem, it may be to your benefit to move on to another question and come back later.
- All code must be written out in proper java format, including all curly braces and semicolons.
- **Do not separate the pages.** You may tear off the one scratch page at the end of the exam. This scratch paper must be turned in or you lose 3 points.
- Turn in all scratch paper to your exam. Do not take any sheets of paper with you.
- If you require extra paper, please use the backs of the exam pages or the extra pages provided at the end of the exam. **Only answers on the FRONT of pages will be grading. The back is for scratch work only.**
- Use a pencil, or blue or black pen to complete the exam.
- If you have any questions, raise your hand and a proctor will come to answer them.
- When you turn in your exam, you may be required to show ID. **If you forgot to bring your ID, talk to an exam proctor immediately.**
- We wish you the best of luck.

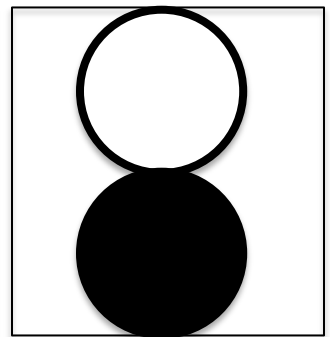
Scores: [For instructor use only]

Question 1		7 pts
Question 2		14 pts
Question 3a		5 pts
Question 3b		8 pts
Question 4		12 pts
Question 5a		8 pts
Question 5b		4 pts
Question 6		12 pts
Total:		70 pts

1.) Miscellaneous (6 points)**1.1) (1 point)** The Easy One:

- Check that your exam has all 10 pages (excluding the cover sheet, which is not numbered).
- Write your name, recitation number, and PennKey (username) on the front of the exam.
- Sign the certification that you comply with the Penn Academic Integrity Code.

1.2) (2 points) In the space provided, write code to draw the figure pictured in the window to the right (The circles are centered vertically). You do not have to draw the window, only the shapes. You do not need to write the class or main declaration.

**1.3) (1 point)** How many times would the following print "Hello World"?

```
int i = 22;
while (i > 0) {
    i /= 3;
    if (i % 2 == 0) {
        System.out.println("Hello World");
    }
}
```



(3 points) Rewrite the above code using a for loop so it prints "Hello World" the same number of times. You are welcome to change the starting value of i, the exit condition, and/or the modification step.



2.) Operators and Expressions (14 points) – 1 point for each blank

For each code fragment, (a) fill in the most appropriate word in the 1st column (if there are multiple blanks, all blanks are the same word) and (b) give the value that z contains after the code has been executed in the 2nd column.

If the code would result in an error, write “ERROR” in the 1st column and give the reason for the error in the 2nd column (you do not need to write the exact error message, just a general explanation). The first two problems have been completed for you.

DO NOT WRITE WHAT PRINTS. ONLY THE TYPE IN THE BLANK AND THE VALUE OF Z

<code>???? z = 1; z = z + 2;</code>	int	3
<code>int x = "Hello World"; ???? z = x.charAt(0);</code>	ERROR	Cannot set an int variable to a String
<code>String str = "hello world"; ???? z = str.length + "14";</code>		
<code>int[] x = {1, 2, 3, 4, 5}; ???? z = x.length + "14";</code>		
<code>int[] x = {1, 2, 3, 4, 5}; String s = "taco salad"; ???? z = x[2] + s.charAt(2) - s.charAt(6); //line 4 is a continuation of line 3</code>		
<code>int y = 5; ???? z = y + "" + 4.0;</code>		
<code>int[] fst = {2, 9, 1, 4, 6}; int[] snd = fst; ???? z = snd[1] % fst[4];</code>		
<code>???? x = "CIS110".charAt(0) == 'C'; ???? y = 3 != 5; ???? z = (!x y) && (y (!y && x));</code>		
<code>???? z = 3 * (3 / 2) + 1.5;</code>		

3a) Conditionals Fill-In (5 points total)

Fill in the blanks for the following method. This method takes in 3 numbers and returns true if they are all different. If any two match, or all 3 match, then it returns false.

```
public static boolean allDiff(double a, double b, double c) {

    if ( _____ ) { //2 points

        if ( _____ ) { //2 points

            return a != c;

        }

    }

    return _____; //1 point

}
```

3b) (8 points) Loops Fill-In

Fill in the blanks for the following method. This function counts the number of odd numbers in an array.

```
public static int oddsInArray(int[] arr) {

    _____ count = _____; // 2 points

    for ( _____ ){ // 3 points

        if (arr[_____] % 2 == _____) { // 2 points
            count++;
        }

    }

    return _____; // 1 point

}
```

4) Tracing (12 points –0.4 points for each blank)

In the following code, print the contents of the x and y arrays at each checkpoint. It is highly recommended you tear off and use the scratch page at the end of the exam.

```
int[] x = {1, 5, 2, 11, 3};
int[] y = {2, 3, 3, 9, 4};

for (int i = 1; i < 5; i += i) {
    int k = 4 - i;
    while (x[i] > y[k]) {
        x[i] -= y[k];
    }
}
//Checkpoint A

for (int i = 0; i < 5; i++) {
    int j = x[i]*2;
    int k = j % 5;
    if (y[k] > x[i]) {
        y[k] += x[i];
        x[i] = y[k];
    }
}
//Checkpoint B

for (int i = 0; i < 5; i+=2) {
    if (x[i] >= y[i]) {
        y[i] = y[i] - x[i];
        x[i] = y[i] - x[i];
    } else {
        i--;
    }
}
//checkpoint C
```

	x	y
Checkpoint A	[_, _, _, _, _]	[_, _, _, _, _]
Checkpoint B	[_, _, _, _, _]	[_, _, _, _, _]
Checkpoint C	[_, _, _, _, _]	[_, _, _, _, _]

5a) Method Mixup (8 points)

The code below is supposed to take in an integer array and return a new array that contains only the even numbered elements. However, Dr. McBurney, who is notoriously disorganized, got all the lines of code out of order. Re-order these lines of code to complete the method. **You may not write your own code. You must use the code as written.**

```
}  
}  
}  
}  
}  
count++;  
for (int i = 0; i < arr.length; i++) {  
for (int i = 0; i < arr.length; i++) {  
if (arr[i] % 2 == 0) {  
if (arr[i] % 2 == 0) {  
insertAt++;  
int count = 0;  
int insertAt = 0;  
int[] out = new int[count];  
out[insertAt] = arr[i];  
return out;
```

```
public class Exam {  
    public static int[] getEvenArray(int[] arr) {  
        //insert lines of code above to complete the method
```

5b) Method Mix-up Testing (4 points)

The method on the previous page takes in an array and returns an array that contains only the even numbers in the same order. Given this description, complete the JUnit test below. You can assume that the method `getEvenArray` is in a file called `Exam.java` in the same folder as this program. (Hint: The method `assertArrayEquals(int[] a, int[] b)` returns true if the **contents** of the arrays `a` and `b` are equal. `assertEquals()` without the word `Array` does NOT work on arrays.) Write the test as we have shown in class. A test input has been generated for you.

```
import static org.junit.Assert.*;
import org.junit.*;

public class ExamTest {
    @Test
    public void getEvenArrayTest() {

        int[] inputArray = {5, 6, 2, 1, 4, 3};
```

6) Method Writing (12 points)

MC BananaPeelZ, a SoundCloud star, has recently been contacted by DMX to produce a cool hip and trendy rap song. While MC BananaPeelZ was very excited to work with a legend, he discovered something tragic: DMX does not write his own ad-libs. In fact, MC Banana Peelz found out that the positioning of the various “Come on!”s, “Yeah”s and “Uh”s throughout DMX songs are actually computer generated!

MC BananaPeelZ discovered that the algorithm works as follows

- 1 *YEAH* on every beat that is a power of 2 (the beat = 2^n for some $n \geq 0$).
- 1 *uh-huh* per every 2 beats.
- If both *YEAH* and *uh-huh* are printed for a given beat, *YEAH* comes **before** *uh-huh*
- Finally, the song ends with a “Grrr... WOOF WOOF.”

Here’s an example of a series of adlibs created by DMX’s adlib method, for a song with only 9 beats. We have annotated the beats with comments. The comments are NOT part of the method output, but the uh-huh's and other adlibs are. The method does not return anything, but simply prints.

uh-huh //beat 0

YEAH! //beat 1

YEAH! //beat 2

uh-huh //beat 2

YEAH! //beat 4

uh-huh //beat 4

uh-huh //beat 6

YEAH! //beat 8

uh-huh //beat 8

Grrr... WOOF WOOF //end of the song

Wanting to expose DMX as a computer generated hack, MC BananaPeelZ has tasked you with writing the code that performs this behavior on the next page. Write a function that produces DMX's "ad-libbing" on the next page.


```
public static void dmxAAdLibPrinter(int totalBeats) {
```

Extra Space for Answers

DO NOT RIP THIS PAGE OFF. ANY WRITING ON THIS PAGE CAN BE GRADED

Scratch Paper

**YOU MAY USE THIS PAGE FOR SCRATCH WORK, BUT NOTHING ON IT WILL
BE GRADED**