Northeastern Illinois University **Name:________________________**
CS 200: Programming I
Professor Yehuda Gutstein **NEIU-ID:_______________________**

Sample Midterm Exam
Instructions:

1. **Do not turn this page until told to do so.**

2. This quiz is ***closed book*** and ***closed notes***.

3. ***Write your STUDENT ID Number on every page***. If you do not write your ID Number on a certain page, you ***will not receive credit for the question on that page!***

4. ***Do not write your name on any page (except this one)***. If you write your name on a certain page, ***you will not receive credit for the question on that page!***

5. There are **five** problems on the exam, one per page. Each problem will be graded as "pass" or "fail".

6. You must give your answer to a question on the ***same sheet of paper that the problem appears on***. If you run out of space on the front of the page, you may continue to the back of that page only. If you put your answer on a different page, ***you will not receive credit for that problem!***

7. For problems that ask you to write code, you should only write the method indicated in the problem. You can assume the following import statement and keyboard declarations:

```
            import java.util.Scanner;

            Scanner keyboard = new Scanner (System.in);
      or    Scanner kbd = new Scanner (System.in);
      or    Scanner input = new Scanner (System.in);
```

8. You may use "SOP" as an abbreviation for "System.out.print" and "SOPln" for "System.out.println".

9. You do not need to do any error checking of input values, ***<u>unless the problem specifically asks you to do so!</u>***

10. If you are caught looking at other papers or communicating with other students in any way, you will receive an **F** for this exam.

Northeasten Illinois University
CS 200: Programming I
Professor Yehuda Gutstein

Name:________________________

NEIU-ID:_______________________

Question 1 Tracing.

What is the **exact** output of the following Java program?

```
public class Tracing1
{
  public static void main(String[] args)
  {
       int tree = 5;
       int bird = 12;
       int lake = 7;
       boolean sun = false;
       boolean moon = true;
       if (!sun)
         System.out.println("eclipse");
       if (tree * tree <= bird)
          System.out.println("tree " + lake);
       else
          System.out.println(bird + "house");
       if ((lake + tree) == bird)
          System.out.print("summer");
       if(moon)
          System.out.println("time");
       if ((tree + lake / 2) == (bird / 2))
       {
          System.out.print("snow");
          System.out.print("flake");
          System.out.println();
       }
       if (tree != tree)
          System.out.println("morning");
       else if (tree < bird)
          System.out.println("afternoon");
       else if(tree == tree)
          System.out.println("evening");
       else
          System.out.println("night");
  }
}
```

Output

```
eclipse
12house
summertime
afternoon
```

Memory

Northeastern Illinois University **Name:__________________________**
CS 200: Programming I
Professor Yehuda Gutstein **NEIU-ID:_________________________**

Question 2 Coding.

Write a JAVA program that has the user enter 3 integers: *a*, *b*, and *c*. If they are all odd, it outputs "ALL ARE ODD". If none are even, it outputs "NONE ARE ODD" If there are some odds and some evens, the program outputs which variables are odd. Here are four ***sample*** program runs:

Sample Output:

```
Enter a: 42
Enter b: 59
Enter c: 112

b is odd
```

```
Enter a: 71
Enter b: 39
Enter c: 217

ALL ARE ODD
```

```
Enter a: 15
Enter b: 44
Enter c: 33

a is odd
c is odd
```

```
Enter a: 18
Enter b: 64
Enter c: 28

NONE ARE ODD
```

```
public static void main(String[] args){
	 Scanner kbd = new Scanner(System.in);
	int a, b, c;

	System.out.print("Enter a: ");
	a = kbd.nextInt();

	System.out.print("Enter b: ");
	b = kbd.nextInt();

	System.out.print("Enter c: ");
	c = kbd.nextInt();

	System.out.println();
	 boolean x = 4%2==0;

	 System.out.println(x);
	if(a % 2 == 1 && b % 2 == 1 && c % 2 == 1)
		System.out.println("ALL ARE ODD");
	else if(a % 2 == 0 && b % 2 == 0 && c % 2 == 0)
		System.out.println("NONE ARE ODD");
	else{
		if(a % 2 == 1)
			System.out.println("a is odd");
		if(b % 2 == 1)
			System.out.println("b is odd");
		if(c % 2 == 1)
			System.out.println("c is odd");
	 }
}
```

Northeastern Illinois University **Name:________________________**
CS 200: Programming I
Professor Yehuda Gutstein **NEIU-ID:_______________________**

Question 3 Coding.

Write a Java program that asks a user to enter two integers, **a** and **b**. You may assume that **a** will be larger than **b**. Your program must compute the product of the integers in the range from **a** through **b**. Here are two ***sample*** runs:

Sample Output:

```
Enter two integers: 3 6

3 x 4 x 5 x 6 = 360
```

```
Enter two integers: 8 12

8 x 9 x 10 x 11 x 12 = 95040
```

```
public static void main(String[] args){
     Scanner kbd = new Scanner(System.in);

     System.out.print("Enter two integers: ");
     int num1 = kbd.nextInt();
     int num2 = kbd.nextInt();
     int product = 1;

     for(int i = num1; i <= num2; i++){
          product *= i;
          System.out.print(i);
          if(i < num2)
               System.out.print(" x ");
     }

     System.out.println(" = " + product);








}
```

Northeastern Illinois University **Name:________________________**
CS 200: Programming I
Professor Yehuda Gutstein **NEIU-ID:_______________________**

Question 4 Coding.
Write a program that has the user enter how many values they will enter. Then, prompt the user to enter these values. The user may enter the same value more than once. Print to the console the number of times the largest integer was entered. Here are three ***sample*** runs:
Sample Output:

```
How many values? 3
Enter integer 1: 32
Enter integer 2: 48
Enter integer 3: 32

The largest integer 48 was
entered 1 time
```

```
How many values? 5
Enter integer 1: 982
Enter integer 2: 341
Enter integer 3: 627
Enter integer 4: 982
Enter integer 5: 35

The largest integer 982 was
entered 2 times
```

```
How many values? 4
Enter integer 1: 60
Enter integer 2: 60
Enter integer 3: 60
Enter integer 4: 60

The largest integer 60 was
entered 4 times
```

```
public static void main(String[] args){
      Scanner kbd = new Scanner(System.in);

      System.out.print("How many values: ");
      int values = kbd.nextInt();
      int[] nums = new int[values];
      int count = 0, max;

      for(int i = 0; i < nums.length; i++){
           System.out.print("Enter integer " + (i + 1)  + ": ");
           nums[i] = kbd.nextInt();
      }

      max = nums[0];
      count = 1;

      for(int i = 1; i < nums.length; i++){
           if(nums[i] > max){
                max = nums[i];
                count = 1;
           }
           else if(nums[i] == max)
                count++;
      }

      System.out.print("The largest integer " + max + " was entered " +
                       count + " time");
      if(count > 1)
           System.out.println("s");
}
```

Northeastern Illinois University **Name:________________________**
CS 200: Programming I
Professor Yehuda Gutstein **NEIU-ID:_______________________**

Question 5 Coding.
Write the method below that determines if its three integer arguments *a*, *b*, and *c* can be the lengths of the sides of a *right* triangle. Three integers can be the lengths of the sides of a right triangle if and only if all three are positive and the square of the largest integer is equal to the sum of the squares of the other two integers. Example: if $a = 4$, $b = 5$, and $c = 3$, then the method would return true, since $b^2 = 5^2 = 25 = 9 + 16 = 3^2 + 4^2 = c^2 + a^2$. Other examples:

The following calls should return *true*:	**The following calls should return *false*:**
`canFormRightTriangle(3, 4, 5)`	`canFormRightTriangle(3, -4, 5)`
`canFormRightTriangle(5, 3, 4)`	`canFormRightTriangle(-5, -4, -3)`
`canFormRightTriangle(6, 10, 8)`	`canFormRightTriangle(0, 0, 0)`
`canFormRightTriangle(10, 8, 6)`	`canFormRightTriangle(5, 0, 5)`
`canFormRightTriangle(12, 5, 13)`	`canFormRightTriangle(1, 2, 3)`

```
public static boolean canFormRightTriangle(int a, int b, int c){
     if(a > 0 && b > 0 && c > 0){
          int max1 = Math.max(a, b);
          int max = Math.max(max1, c);
          int min1 = Math.min(a, b);
          int min = Math.min(min1, c);
          int middle;

          if(a != max & a != min)
               middle = a;
          else if(b != max && b != min)
               middle = b;
          else
               middle = c;

          return Math.pow(min,2) + Math.pow(middle, 2) == Math.pow(max,2);
          }
     else{
          return false;
     }



}
```