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

Midterm Exam
SOLUTIONS
Instructions:

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

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

3. ***Write your name on every page***.

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

5. You must place your answer to a question on the **specified** page in the **ANSWER PACKET**. If you place your answer anywhere else, **you will not receive credit for it!**

6. For problems that ask you to write a **method**, you must use the given method header **exactly** as shown, and you do not need to write the main( ) method.

7. You may use "kb", "kbd", "keyboard" or "input" to get input from the keyboard without defining them. Assume the following import statement and keyboard declarations have already been written for you (do not write these statements in your answers):

```
                import java.util.Scanner;

                Scanner keyboard = new Scanner (System.in);
        or      Scanner kb = 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.

Northeastern Illinois University **Name:________________________**

CS 200: Programming I

Professor Yehuda Gutstein **NEIU-ID:_______________________**

Question 1 (20 pts) Tracing.

What is the output of the following Java program?

```
public class Tracing {
   public static void main(String[] args) {
      int arm = 8, leg = 14, hand = 10, foot;
      while (hand < leg) {
          if(hand % 2 == 0)
            System.out.println(arm);
          else
            System.out.print(hand);
          arm--;
          hand++;
      }
      System.out.println();
      System.out.println(hand + " " + arm);

      for (foot = 1; foot < 5; foot++) {
          if (foot % 2 != 0)
            System.out.println("foot: " + foot);
          else
            System.out.println("hand & foot: "
                                 + (hand + foot));
      }
      printPart(foot, leg);

      for (arm = 3; arm > 1; arm--) {
          System.out.println("begin: " + arm);
          System.out.print(hand++);
          System.out.println();
          System.out.println("end: " + arm);
      }

      printPart(hand, arm);
   }

   public static void printPart(int x, int y) {
      if(x > y) {
          System.out.println("arm: " + y);
          System.out.println("hand: " + x);
      }
      else {
          System.out.println("foot: " + x);
          System.out.println("leg: " + y);
      }
   }
} // end of class
```

Output

```
8
116
13
14 4
foot: 1
hand & foot: 16
foot: 3
hand & foot: 18
foot: 5
leg: 14
begin: 3
14
end: 3
begin: 2
15
end: 2
arm: 1
hand: 16
```

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

Question 2 (20 pts) Coding.

- Write the main method of a program that has the user enter an integer value for n.
- A number that divides evenly into another number n is called a factor of n.
- Assume n is positive.
- The program adds the factors of n and displays them from largest to smallest as shown below.
- A number that is only divisible by 1 and n is prime.
- The output must be formatted exactly as the samples below.
- Sample Output:

```
Enter n:  20

10+5+4+2+1 is 22
```

```
Enter n:  13

1 is 1
```

```java
public static void main(String[] args){
   Scanner kbd = new Scanner(System.in);
   System.out.print("Enter n: ");
   int n = kbd.nextInt();
   int sum = 0, factor = n/2;

   while(factor >= 1){
      if(n % factor == 0){
         sum += factor;
         System.out.print(factor);

         if(factor > 1)
             System.out.print("+");
      }

      factor--;
   }
   System.out.println();
   System.out.print(" is " + sum);
}
```

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

Question 3 (20 pts) Coding.

- Write the main method of a class that asks the user to enter numbers until the sum of all of the numbers that are entered is exactly 10 .
- The program then displays how many numbers were entered.
- The output must be formatted exactly as the samples below.
- Sample Output:

```
Enter a number: 1.5
Enter a number: 3.5
Enter a number: 5

3
```

```
Enter a number: 20
Enter a number: -12.25
Enter a number: 0
Enter a number: 2.25

4
```

```
Enter a number: 9.1
Enter a number: -1.5
Enter a number: 1
Enter a number: -10
Enter a number: 0
Enter a number: 10
Enter a number: 1.4

7
```

```java
public static void main(String[] args){
   Scanner kbd = new Scanner(System.in);
   double sum = 0;
   int count = 0;

   while(sum != 10){
      System.out.print("Enter a number: ");
      double num = kbd.nextDouble();

      sum += num;
      count++;
   }

   System.out.println();
   System.out.print(count);
}
```

Question 4 (20 pts) Coding.

- Write the method *madeProgress* that accepts two integer parameters, *start* and *end*, as well as a boolean parameter named *penalty*.
- Given a starting position expressed as an integer value and an ending position also expressed as an integer, the progress that has been made is the difference between them—unless there is a penalty, in which case the progress is reduced by 10.
- The method returns **true** if the progress made is positive or **false** otherwise.
- Sample Runs:

madeProgress(10, 15, false) would return **true**

madeProgress(10, 20, true) would return **false**

madeProgress(100, 95, true) would return **false**

madeProgress(5, 75, false) would return **true**

madeProgress(50, 30, false) would return **false**

```
public static boolean madeProgress(int start, int end, boolean penalty){
    int result = end - start;

    if(penalty)
        result -= 10;

    return result > 0;
}
```

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

<u>Question 5 (20 pts) Coding.</u>

- Write the main method of a class that asks the user to enter the number of hours they exercised every day for two consecutive weeks.
- The program should then list the total hours exercised each week and print whether the user's exercise hours increased, decreased, or stayed the same.
- In addition, the program should output the number of days that there was a decrease in exercise hours when compared to the same day the previous week.
- The output must be formatted exactly as the samples below.
- Sample Output:

```
Week 1: 2 5 3 2 8 0 0
Week 2: 0 5 5 2 6 2 2

Week 1: 20 hours
Week 2: 22 hours

Total hours increased.
Days decreased: 2
```

```
Week 1: 1 5 3 2 8 2 0
Week 2: 1 5 3 2 8 2 0

Week 1: 21 hours
Week 2: 21 hours

Total hours the same.
Days decreased: 0
```

```
Week 1: 0 1 3 4 2 2 0
Week 2: 0 1 1 2 1 0 0

Week 1: 12 hours
Week 2: 5 hours

Total hours decreased.
Days decreased: 4
```

```java
public static void main(String[] args){
     Scanner kbd = new Scanner(System.in);
     int[] wk1 = new int[7], wk2 = new int[7];
     int wk1Hrs = 0, wk2Hrs = 0, days = 0;
     System.out.print("Week 1: ");
     for(int i = 0; i < wk1.length; i++){
         wk1[i] = kbd.nextInt();
         wk1Hrs += wk1[i];
     }
     System.out.print("Week 2: ");
     for(int i = 0; i < wk2.length; i++){
         wk2[i] = kbd.nextInt();
         wk2Hrs += wk2[i];
         if(wk1[i] > wk2[i])
             days++;
     }
     System.out.println();
     System.out.println("Week 1: " + wk1Hrs + " hours");
     System.out.println("Week 2: " + wk2Hrs + " hours");
     System.out.println();
     if(wk1Hrs > wk2Hrs)
         System.out.println("Total hours increased.");
     else if(wk1Hrs < wk2Hrs)
         System.out.println("Total hours decreased.");
     else
         System.out.println("Total hours the same.");

     System.out.println("Days decreased: " + days);
}
```