Comp 110 – Summer II 2012 – Midterm Exam

UNC Honor Pledge: I certify that no unauthorized assistance has been received or given in the completion of this work.

Signature: __________________________

Read this entire first page before beginning the exam. Do not flip to the next page until authorized.

You may NOT use any course materials in completing this exam.

You will have 55 minutes to complete this exam.

This exam consists of three sections.

1. (40 points) Short Answer Questions
2. (20 points) Tracing Short Pieces of Code
3. (40 points) Writing Code

In Section 1, complete ALL questions.

In Sections 2 and 3 complete TWO of the possible three questions in each section. If you answer all three questions in a section, the two on which you receive the highest scores will be counted towards your total.

The exam is out of 100 points. It is possible to attempt for more than 100 points. If your final score is greater than 100, the extra points will be counted as extra credit.

Pace yourself! If you find that you are stuck for a long time on a question that is not worth many points, move on to another question that you know how to do. If you can’t finish a problem, you may receive partial credit for partial answers, so show incremental work.

Please print your name on every page of the exam.

If something is unclear or if you need extra scratch paper, raise your hand. Write your name on any extra scratch paper and hand it in with your exam.

Write clearly! Illegible answers will not receive credit.
1. (2 points) Which of the following are primitive types in Java (circle all that apply)?

- decimal
- boolean
- String
- long
- Student

2. (4 points) Which of the following are illegal variable names in Java (circle them)? Which of the following names are legal but do not follow Java conventions for variables (box them)?

- week-day
- dayCount
- TotalCost
- default
- 2pieces
- import
- helloWorld!
- itemPrice

3. (4 points) What are the values of the following variables?

   a. \(1.0\) double \(\text{var1} = (15 / 4) - 2.0;\)
   b. \(3\) int \(\text{var2} = (\text{int})(9.4 / 3.0);\)
   c. \(5.5\) float \(\text{var3} = (\text{float})11 / 2;\)
   d. \(1\) int \(\text{var4} = 21 \% 4;\)

4. (2 points) Declare and assign the following. Name your Java variable so that any programmer would know what value it holds. Your variable name should be at least two words. Assign the variable a reasonable initialization value or object.

   a. A variable to hold the winning score from one team in a baseball game.

      ```java
      int \text{gameScore} = 7; // There are a lot of possibilities for this answer
      ```

   b. An array that could hold a list of a gymnast’s scores (e.g. 7.2, 8.5, …) from a panel of seven judges.

      ```java
      double[] \text{gymnastScores} = \text{new double[7]};
      ```

5. (3 points) What are the three main design principles of Java?

   Encapsulation, Polymorphism, Inheritance

6. (2 points) Given these variables, what is the value of these Boolean expressions?

   a. int \(x = 5, y = 9;\)
      \((x < 7) || (y < 3)\) true

   b. int \(x = 3, y = 4;\)
      \(((x + y > 7) || (x != 6)) && (y >= 5)\) false
7. (2 points) Suppose you have the variable str of type String with the following data: “What happened to three?” Write out the data stored in str and place the index of each character above the string.

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11| 12| 13| 14| 15| 16| 17| 18| 19| 20| 21| 22|
| W | h | a | t | h | a | p | p | e | n | e | d | t | o | t | h | r | e | e | ?

8. (6 points) Give the return type and data of the following methods calls on str, as it is defined in the above problem.

   a. str.length()  int  23
   b. str.charAt(9)  char  ‘e’
   c. str.indexOf(“d”)  int  12
   d. str.equals(“What”)  boolean  false
   e. str.lastIndexOf(“t”)  int  17
   f. str.substring(10, 20)  String  “ned to thr”

9. (9 points) Consider the 1-D integer array {8, 5, 10, 3}. Show the state of this array after each iteration of the outer loop for bubble and selection sorting algorithms.

<table>
<thead>
<tr>
<th>Bubble Sort</th>
<th>Selection Sort</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. 8, 5, 10, 3</td>
<td>0. 8, 5, 10, 3</td>
</tr>
<tr>
<td>1. 5, 8, 3, 10</td>
<td>1. 3, 5, 10, 8</td>
</tr>
<tr>
<td>2. 5, 3, 8, 10</td>
<td>2. 3, 5, 10, 8</td>
</tr>
<tr>
<td>3. 3, 5, 8, 10</td>
<td>3. 3, 5, 8, 10</td>
</tr>
<tr>
<td></td>
<td>4. 3, 5, 8, 10</td>
</tr>
</tbody>
</table>

Step 4 may or may not exist, depending on the presence of the index < arr.length - 1 optimization on p. 495 (5th edition)
10. (3 points) What is wrong with this code fragment? Circle or draw an arrow indicating the source of the problem. Indicate why it is an error. Suggest a possible solution.

```java
for (int x = 0; x < 5; x++)
{
    System.out.println("x = " + x);
    int y = 1;
    while (y < 5)
    {
        System.out.println("x * y = " + (x * y));
        y++; }  
    System.out.println();
}
```

This an infinite loop because the y++ statement is not part of the while loops’s body. One possible solution would be to surround the two indented statements (System.out.println(…) and y++) inside of {}’s.

11. (3 points) In the code below, how many times is the underlined statement executed?

```java
double[] values = { 3.0, 4.0, 5.0, 9.0 };  
int i = 0;  
do 
{  
    for (int j = 6; j >= 0; j -= 2) 4 loops(j = 6, 4, 2, 0)  
    {  
        for (double val : values) 4 loops(values.length = 4)  
        {  
            System.out.println(i + ", " + j + ": " + val);  
        }  
        i++;  
    } while (i < 3) 3 loops(i = 0, 1, 2)

4 * 4 * 3 = 48 times
```
12. (10 points) What is the value of temp at the end of the loop below for the following input? 5

Input: 10 4 9 15 7 12 2 21 -1

Scanner kb = new Scanner(System.in);
int input = kb.nextInt();
int temp = 0;
while (input > 0)
{
    if (input < 12)
    {
        temp++;
        input = kb.nextInt();
    }
}

13. (10 points) What is the value of temp at the end of the loop below for the following input? 21

Input: 2 7 9 4 1 10 3 14 -1

Scanner kb = new Scanner(System.in);
int temp = 0;
for (int k = 0; k < 6; k++)
{
    int input = kb.nextInt();
    if (k % 2 == 1)
    {
        temp += input;
    }
}

14. (10 points) What is the value of temp at the end of the loop below for the following input? 96

Input: 2 5 -4 -1 3 -5 6 -2 0

Scanner kb = new Scanner(System.in);
int input = kb.nextInt();
int temp = 1;
do
{
    if (input % 2 == 0)
    {
        temp *= input;
        input = kb.nextInt();
    }
} while (input != 0);

END OF SECTION 2
15. (15 points) Write a code fragment that replaces each element in an integer array named intArr with its absolute value. The absolute value of a number is its numerical value with regard to its sign. For example, the absolute value of -3 is 3. The absolute value of 3 is also 3. Assume that intArr is already initialized and is ready for you to use. You do not need to create any new arrays.

```java
for (int i = 0; i < intArr.length; i++)
    if (intArr[i] < 0)
        intArr[i] = -intArr[i];
```

16. (20 points) Write the code for Yoda-fying a sentence by moving the middle words of a String named str to the front of that string. For example if str were “you will not go!” then the result string would be “will not you go!” Assume that str has already been declared for you and contains the string that you will manipulate. Store the result in a new String variable. Do not worry about error checking or capitalization. I will not take off points for simple syntax or off-by-one errors. The only String methods that you will need are in Question 8.

```java
int firstSpace = str.indexOf(" ");
int lastSpace = str.lastIndexOf(" ");
String result =
    str.substring(firstSpace + 1, lastSpace + 1) +
    str.substring(0, firstSpace) +
    str.substring(lastSpace);
```
17. (25 points) Write the code that computes which player is next in a partially filled in Tic-Tac-Toe board. Assume that the game board is a 3x3, 2-D array of element type char named board. The elements of board are one of ‘X’, ‘O’, or ‘ ’. In an empty game board, player X goes first. Store your result in a char variable named nextPlayer. Assume that board is already initialized, filled with data (‘X’, ‘O’, or ‘ ’), and ready for you to use. Do not worry about error checking. Assume that there is at least one square for a player to place an X or O (the game is not over). **Hint: Count.**

Examples boards and resultant values for nextPlayer:

```
|   |   |   | nextPlayer = 'X' |
|   |   |   | nextPlayer = 'O' |
|   |   |   | nextPlayer = 'X' |
```

```java
int xCount = 0, oCount = 0;
for (int row = 0; row < board.length; row++)
{
    for (int col = 0; col < board[row].length; col++)
    {
        if (board[row][col] == 'X')
            xCount++;
        else if (board[row][col] == 'O')
            oCount++;
    }
}

char nextPlayer = (xCount == oCount) ? 'X' : 'O';
```

Basically, if there are the same number of X’s and O’s, it is X’s turn. If there are more X’s, then it is O’s turn. `<=` could be used instead of `==` for the last conditional.