10/09/15 APCS – Monta Vista High School Second Test Name_____

1. (**15 points**) Show the exact output produced by the following code fragment below. It is a good idea to show how the variables are changing (with some boxes) as you move through the loop.

```
int x = 2;
int number = 3;
int value = 10;

while (value < 18)
{
        if (value % number  == 0)
        {
                value++;
                number++;
        }
        else if (value % x == 0)
        {
                value++;
                x++;
        }
        else
        {
                number--;
                x--;
        }
        value++;
        System.out.println(x + "   " +
                number + "   " + value);
}
```

x	number	value

Output:

2. (**10 points**) Evaluate the following expressions. If there is a syntax error then write **ERROR**.

(a) **(int) (Math.PI * 100) == 314** _____

(b) **89 / 23 + 67 - 16 % 17 * 4** _____

(c) **(! 65 > 4 * 15 || false)** _____

(d) **! (8 / 7.8 > 1 && true) || (int) 7.4 < 7.4** _____

(e) **65 * 10.0 > 1324 / 4 && 30 / 9 < 3.0 || ! true** _____

3. (**15 points**) Determine what will be printed from the following code fragment.

```
for (int row = 1; row <= 5; row++)
{
        int col = 1;
        while (col <= 5)
        {
                if (row == col || row + col == 5)
                {
                        System.out.print(row);
                }
                else
                {
                        System.out.print("*");
                        //  an asterisk
                }
                col++;
        }
        System.out.println();        //  a new line
}
```

row	col

Output:

4. (**14 points**) True/False

_____ (a) Fields are declared inside the constructor.

_____ (b) All local variables should be declared private.

_____ (c) The scope of a field extends throughout the class.

_____ (d) The cast operator can be applied for any primitive data type to any other primitive data type.

_____ (e) The name of a method should sound like an adjective (for example: slow, active).

_____ (f) Naming your class "**string**" is bad because the bytecode file **string.class** conflicts with Java's **String.class**

_____ (g) The Java compiler recognizes nested blocks through indentation.

_____ (h) Text within double quotes cannot be split between two lines.

_____ (i) **static**, **void**, **true**, and **main** are all reserved words.

_____ (j) Method names should always begin with a lowercase letter.

_____ (k) The scope of a local variable extends from the time it is declared to the end of the method.

_____ (l) Using the same name for a field and a local variable causes a compiler error.

_____ (m) In the absence of parentheses, binary operators of the same rank are performed left to right.

_____ (n) A short-circuit evaluation refers to a binary logical operation in which the first operand is sometimes sufficient to determine the result.

5. (**12 points**) Write the method **alternate** that receives two **Strings** in its parameter list, **one** and **two**. The method **alternate** should create a new **String** made of the first char of **one**, the first char of **two**, the second char of **one**, the second char of **two**, and so on. Any leftover chars should be placed at the end of this new **String**, and this **String** should be returned.

alternate("pot","cab") returns **"pcoatb"**
alternate("excellent","work") returns **"ewxocrekllent"**
alternate("mixed","up") returns **"muipxed"**
alternate("long","weekend") returns **"lwoenegkend"**

public String alternate (String one, String two)

6. (**12 points**) Write a method **positionOfClosest** that returns the **index** of the element in the array closest to the given value (both passed in the parameter list). You may assume that the length of the array is at least one.

For example, if we have the array **int [] array = {10, 20, 30, 40, 50, 60};**, then the method call **positionOfClosest(array, 47);** will return **4**, since the element at index 4 (50) is the closest to 47.
For the same array, the method call **positionOfClosest(array, 23);** will return **1**, since the element at index 1 (20) is the closest to 23.

public int positionOfClosest (int [] a, int value)

7. (**12 points**) Write a method **countVowels** that takes a String **str** in its parameter list. The method **countVowels** should create an array of integers with a length of 5. The method **countVowels** should count the number of vowels (a, e, i, o, u) occurring in the String **str**, and record them in the array of integers (length 5). This array should be returned by the method **countVowels**. See the examples below. For full credit, make use of a **switch** control structure.

countVowels("aardvark") returns **{3, 0, 0, 0, 0}**
countVowels("EXCELLENT!") returns **{0, 3, 0, 0, 0}**
countVowels("programming") returns **{1, 0, 1, 1, 0}**
countVowels("COOL METHOD") returns **{0, 1, 0, 3, 0}**

public int [] countVowels(String str)

8. (**12 points**) Create a flowchart for the following algorithm. Find and print all factors for a given positive integer. You may assume that the input is a positive integer. See the examples below.
For the integer **5**, the numbers **1 5** would be printed
For the integer **12**, the numbers **1 2 3 4 6 12** would be printed
For the integer **25**, the numbers **1 5 25** would be printed