IB Computer Science

Student_____

Java Programming

Date: _____

Directions

1. Take a look at program Trace03.java on the last page.
2. Read the Java Notes below, and refer to the Trace03.java program. Notice that many of the statements are very similar to C.
3. Trace the program and determine its output.
4. Copy the source code for Trace03.java into your Cloud9 environment in the lessons directory to see the output, and whether it match was you would expect.
5. Copy the program and make the following changes:
    a. Give it a new name, replacing the name Trace03
    b. save the file with the reflecting your new name (.java)
    c. Rename method03, give it a better name, one that represents what it does.
    d. Rewrite the `System.out.println` statement because it is, well, awful.
    e. Compile and test/debug your new program. Be sure to change the top line.

Java Notes

General things to notice:

1. In addition to have main in class Trace03, there is also a method named method03 defined inside of class Trace03.
2. Method method03 is called from main.
3. Method03 is called from main, passing in a string.
4. These Java concepts are not too far from the C concepts used last year.

Dissecting some of the individual Java statements:

```
public static int[] method03(String s) {
```

This is the declaration for method03. The (String s) clause means that method03 will accept a **String** as a parameter. The **`int[]`** means that **`method03`** will return an array of integers.

```
int[] ret = new int[5];
```

```
int[] ret;
```

Declares that variable ret is an array of integers – but like C, it is really just a pointer (address) of where the array begins in memory. This declaration does not say how many elements are in the array.

```
int[] ret = new int[5];
```

Memory for `ret` is allocated usubg the `"new"` statement. This declaration says to allocate enough space to contain 5 integers, indexed from `ret[0]` to `ret[4].`

```
String temp="";
```

Declares that temp is a variable of type String.
This means that there many methods available to be used with temp because they are part of the String class.

Student_____

Java Programming

Date: _____

```
temp = s.toUpperCase();
```

Variable s is also a String. One of the methods available to a String is the toUpperCase method, which as you can guess, will return a string that is equivalent to the upper-case translation of variable s.

This statement will not alter variable s, the upper-case version of s will be stored in temp.

```
for (i = 0; i < s.length(); i++) {
```

```
s.length() returns the length of String s, it is another one of
the methods available for a String variable.
```

```
int   A = (int)'A';
```

Just like in C, (int) will cast the character 'A' into an integer – which will give the integer variable A the ASCII value of 'A'. Normally I wouldn't use an upper-case letter as a variable name, but what better variable could I have used? mrBrennansVariableNameForCapitalA ?

Student_____

Java Programming

Date: _____

```
public class Trace03 {

   public static int[] method03(String s) {
        int[] ret = new int[5];
        int   i=0;
        int   c=' ';
        String temp="";

        int   A = (int)'A';
        int   E = (int)'E';
        int   I = (int)'I';
        int   O = (int)'O';
        int   U = (int)'U';

        temp = s.toUpperCase();

        for (i=0; i<5; i++)
            ret[i] = 0;

        for (i = 0; i < s.length(); i++) {
             c = (int)temp.charAt(i);
             if (c == A) ret[0]++;
             else if (c == E) ret[1]++;
             else if (c == I) ret[2]++;
             else if (c == O) ret[3]++;
             else if (c == U) ret[4]++;
            } // for loop

        return ret;
   } // end method method03



    public static void main(String[] args) {
         int i = 0;      // will be used as a loop variable

         int[] rc = new int[5]; // declares an array of integers, they are
                                // indexed from 0 to 4.

         rc = method03("I think, therefore I am");
             // pass a string to a method named method03
             // that is defined above main

         for (i=0; i<5; i++) {
              System.out.println("rc[" + i + "] = " + rc[i]);
              } // for loop

    } // end main

} // end class Trace03
```