csTask12 Assignment

The activities in this assignment require you to create and modify Java programs using BlueJ. Create a new BlueJ project named csTask12.

All programs (class files) must be in your new project, and the project must be copied to your shared Dropbox folder before the end of class today. Recall that you can compress (zip) your project folder and then upload it to Dropbox as a single file.

When questions are posed, write your responses as full sentences in the Readme.txt file in your project.

There are four parts to this assignment:

I. Names and Places
II. A Table of Student Grades
III. Two Meanings of +
IV. Reading Information from the Keyboard

I. Names and Places

The goal in this exercise is to develop a program that will print out a list of student names together with other information for each. The tab character (an escape sequence) is helpful in getting the list to line up nicely.

You are to copy and paste the following Java program into a new class in your project. Use Edit / Auto-layout to format the program and then fill in the Javadoc comment space at the top of the class.

```
// ***************************************************************
// Prints a list of student names with their hometowns
// and intended major
// ***************************************************************
public class Names
{
// ------------------------
// main prints the list
// ------------------------
public static void main (String[] args)
{
System.out.println ();
System.out.println ("\tName\t\tHometown");
System.out.println ("\t====\t\t========");
System.out.println ("\tSally\t\tRoanoke");
System.out.println ("\tAlexander\tWashington");
System.out.println ();
}
}
```

1. Compile the program and run it to see how it works.
2. Modify the program so that your name and hometown and the name and hometown of at least two classmates sitting near you in class also are printed. Save, compile and run the program. Make sure the columns line up.
3. Modify the program to add a third column with the intended major of each person (assume Sally's major is Computer Science and Alexander's major is Math). Be sure to add a label at the top of the third column and be sure everything is lined up (use tab characters!).

II. A Table of Student Grades

Write a Java program that prints a table with a list of at least 5 students together with their grades earned (lab points, bonus points, and the total) in the format below.

```
///////////////////\\\\\\\\\\\\\\\\\\\
        == Student Points ==
\\\\\\\\\\\\\\\\\\\///////////////////

Name        Lab         Bonus       Total
----        ---         -----       -----
Joe         43          7           50
William     50          8           58
Mary Sue    39          10          49
```

The requirements for the program are as follows:
1. Print the border on the top as illustrated (using the slash and backslash characters).
2. Use tab characters to get your columns aligned and you must use the + operator both for addition and string concatenation.
3. Make up your own student names and points—the ones shown are just for illustration purposes. You need 5 names.

Note: A table showing all of the Java escape sequences for convenient formatting is on page 60 of Chapter 2 your text. The pdf of the text is in the shared Google Drive folder.

III. Two Meanings of +

In Java, the symbol + can be used to add numbers or to concatenate strings. This exercise illustrates both uses. When using a string literal (a sequence of characters enclosed in double quotation marks) in Java the complete string must fit on one line. The following is NOT legal (it would result in a compile-time error).

```
System.out.println ("It is NOT okay to go to the next line
in a LONG string!!!");
```

The solution is to break the long string up into two shorter strings that are joined using the concatenation operator (which is the + symbol). So the following would be legal.

```
System.out.println ("It is OKAY to break a long string into " +
"parts and join them with a + symbol.");
```

So, when working with strings the + symbol means to concatenate the strings (join them). BUT, when working with numbers the + means what it has always meant—add!

1. Observing the Behavior of + To see the behavior of + in different settings do the following:

a. You are to copy and paste the following Java program into a new class in your project. Use Edit / Auto-layout to format the program and then fill in the Javadoc comment space at the top of the class.

```
// *******************************************************************
// Demonstrate the different behaviors of the + operator
// *******************************************************************
public class PlusTest
{
// -------------------------------------------------
// main prints some expressions using the + operator
// -------------------------------------------------
public static void main (String[] args)
{
System.out.println ("This is a long string that is the " +
"concatenation of two shorter strings.");
System.out.println ("The first computer was invented about" + 55 +
"years ago.");
System.out.println ("8 plus 5 is " + 8 + 5);
System.out.println ("8 plus 5 is " + (8 + 5));
System.out.println (8 + 5 + " equals 8 plus 5.");
}
}
```

b. Compile and run the program. For each of the last three output statements (the ones dealing with 8 plus 5) write down what was printed. Now for each explain why the computer printed what it did given that the following rules are used for +. Write out complete explanations.
_ If both operands are numbers + is treated as ordinary addition. (NOTE: in the expression a + b the a and b are called the operands.)
_ If at least one operand is a string the other operand is converted to a string and + is the concatenation operator.
_ If an expression contains more than one operation expressions inside parentheses are evaluated first. If there are no parentheses the expression is evaluated left to right.

c. The statement about when the computer was invented is too scrunched up. How should that be fixed?

III. Two Meanings of + (continued)

2. Writing Your Own Program With +

Now write a complete Java program that prints out the following sentence:

```
Ten robins plus 13 canaries is 23 birds.
```

Your program must use only one statement that invokes the println method. It must use the + operator both to do arithmetic and string concatenation.

IV. Reading Information from the Keyboard

The following program reads three integers and prints the average using the `Scanner` class. The `Scanner` class has been written by someone else (Thank you!) and exists in the `java.util` package. The `import` declaration at the beginning (after the comment but before the class header) says that the `java.util.Scanner` class may be used in the program. The declaration

```
Scanner scanIn = new Scanner(System.in);
```

creates a variable called `scanIn` that represents a `Scanner` object that reads input from the keyboard.

Once the Scanner variable has been declared, you can use the following statement to cause the program to accept the next `int` value from the keyboard and store it in the `int` variable `x` :

```
x = scanIn.nextInt();
```

Similarly, the following statement causes the program to accept the next `double` value from the keyboard and store it in the `double` variable `y` :

```
y = scanIn.nextDouble();
```

You are to copy and paste the following Java program into a new class in your project. Use Edit / Auto-layout to format the program and then fill in the Javadoc comment space at the top of the class. Then fill in the blank lines so that it will work correctly.

```
// *******************************************************************
// Read three integers from the user and print their average
// *******************************************************************
import java.util.Scanner;
public class Average
{
public static void main(String[] args)
{
int val1, val2, val3;
double average;
Scanner scanIn = new Scanner(System.in) ;
// get three values from user
System.out.println("Please enter three integers and " +
"I will compute their average");
____________________________________
____________________________________
____________________________________
//compute the average
____________________________________
//print the average
____________________________________
}
}
```