Computer Science 121

Lab 9
100 Points

In this lab, you have four questions. You will be practicing methods in this lab.
You need to finish question1, 2 and 3 in the lab and demonstrate to your instructor. You need to submit question 4 in e-campus before the next lab and demonstrate during the next lab.

Also, include the following comment at the beginning of your code:

```
/*
* Your Name
* Submission Date
* CS 121 Lab
* Name of Program
*
* Describe what the program does in one to two sentences.
*/
```

Also, explain what is going on at different parts of your code. For example, state where you're declaring variable, where you're getting user input, where you're performing computations, where you're outputting results, etc. **If you do not include comments throughout your code, you will lose points!!!**

1 Method Exercise (20 Points)

Create a new Java program, and name the class `Method`.

You're given the following program segment:

```
public class Method
{
    public static void main(String[] args)
    {
        int num;
        double dec;
        ...
    }

    public static int one(int x, int y)
    {
        ...
    }

    public static double two(int x, double a)
    {
```

```
        int first;
        double z;
        ...
    }
}
```

a. Write the definition of method `one` so that it returns the sum of `x` and `y` if `x` is greater than `y`. Otherwise, it should return `x` minus 2 times `y`.

b. Write the definition of method `two` as follows:

   i. Read a number and store it in `z`.
   ii. Update the value of `z` by adding the value of `a` to its previous value.
   iii. Assign the variable `first` the value returned by method `one` with arguments 6 and 8.
   iv. Update the value of `first` by adding the value of `x` to its previous value.
   v. If the value of `z` is more than twice the value of `first`, return `z`. Otherwise, return 2 times `first` minus `z`.

c. `main` should ask the user to enter values that are stored in `num` and `dec`. `main` should then call `two` with arguments `num` and `dec`. Finally, have `main` output the result of method `two`.

2 Vowel Counting (30 Points)

Create a new Java program, and name the class `Vowel`.

Write a program that prompts the user to input a String and outputs the number of vowels. To do this, we will write a value-returning method (called `isVowel`) that returns `true` if a given character is a vowel (assume y is not a vowel). If the character is not a vowel, the method should return `false`.

For example, if the String is `Have a good spring break.`, the program should say there are 8 vowels.

Main method that invokes the isVowel is given as below. You still need to create a class, place in the main method and implement isVowel method.

```
public static void main(String[] args)
{
   String inputLine;
   Scanner console = new Scanner(System.in);

   int count = 0;

   System.out.print("Enter a String: ");
   inputLine = console.nextLine();
   System.out.println();

   for (int i = 0; i < inputLine.length(); i++)
```

```
            if (isVowel(inputLine.charAt(i)))
               count++;

        System.out.println();
        System.out.println("There are " + count + " vowel in the input
line");
    }
```

3 Overloading average (20 Points)

Create a new Java program and name the class AverageNums.
Write three methods all named `average` where the first method inputs two int values and returns the int average of them, the second method inputs two double values and returns the double average of them, and the third method inputs three double values and returns the double average of them. (Remember all three methods must be named the same name *average*, however the parameter types and return types are different, this is known as overloading of methods). In your main method, use all the methods with the required number of parameters for each methods (you can give inputs manually) and print out the results.

4 Reverse Digit (30 Points)

Create a new Java program, and name the class `Digit`.

Write a method, `reverseDigit` that takes an integer as a parameter and returns the number with its digits reverse. For example, the value of `reverseDigit(12345)` is `54321`. Let `main` ask the user to enter an integer, that will be passed as an argument to the method. Note that the method should allow negative numbers.

Hint: The number 10 will be extremely useful.

Let's say you have number 543, here is a way to inverse it:

543%10 = 3,
3*10 + (543/10)%10 = 34
34*10 + ((543/10)/10) %10 = 345

Submission

When you are done, you will need to submit your code for both questions on eCampus. When logging onto eCampus, be sure to select your corresponding lab session instead of T01. When you submit your code, be sure to submit the .java file, not the .java˜ or .class files.