IS115 Introduction to Programming Fall 2016

Assignment #7 - Total Points = 100 points

Date Assigned: **10/3**

Date Due: **10/10**

This assignment is based on the material covered in **chapter 6** of your textbook.

We will use the Raptor flowcharting tool for this assignment.

Helper file: See the helper files for each of the chapters/concepts located under the assignment link where you picked up this assignment file for a sample question and answer.

Deliverable: **One Raptor program file (.rap file).** A good file naming convention always includes your name. For example, I might name my files something like: **hw7_jim_smith_Q1.rap.**

I am also asking for code documentation. Your program must contain several comment lines at key steps of the program with the following information:

Your name:

Course number and course section:

Date of completion:

The time it took you to complete this exercise: Log your time from the time you start this exercise till completion. For example, you could write 1 hour of 30 minutes, 90 minutes, etc.

Brief explanation of the program:

In **Raptor**, you right click any component such as Start, Input, etc. and select Comment. All comment lines will be color coded.

Make sure your ALL output is descriptive. Don't just display a value.

1. **Based on chapter 6 (More loops & decisions):** Use Raptor for the following problem.

a) Ask the user for the number of students who will be using this program. For each student, ask for two numbers (whole numbers such as 10 or 5, etc.). **The first number must be less than the second number.** This constitutes valid data. Don't worry about checking for decimal values. The user can enter any number.

b) If the data is not valid, issue an error message and ask the user for two new numbers.

c) Display all **odd numbers** between the first number and the second number.

d) Display all **even numbers** between the first number and the second number.
e) Calculate and display the **total** of all **odd numbers** between the first and second number. The total value must be displayed only once.
f) Calculate and display the **total** of all **even numbers** between the first and second number. The total value must be displayed only once.
g) Calculate and display the **total** of all **numbers** between the first and second number. The total value must be displayed only once.

As an example, if the user enters 3 for the first number and 7 for the second number, the program will display 3 5 7 for the **odd** numbers, 4 6 for the **even** numbers, total of all odd numbers will be 15 (3+5+7 = 15), and the total of all even numbers will be 10 (4+6). The total of all numbers will be 25. You should include simple accumulators to calculate the totals. Of course, this just a sample run. Your code should work for any number of input values. As another example, if the user enters 7 for the first number and 3 for the second number, your code issues an error message and allows the user to enter another set of numbers.