CS 2340 Computer Architecture

Homework 1: MIPS Programming Basics

Objective: Get started with MIPS programming in a MARS environment. Practice load/store instructions, add and sub instructions, as well as MIPS syscalls.

Instructions

Create a MIPS program that fulfills the following specifications:

- in the .data section:
  - 3 variables to hold input values: a, b, c
  - 3 variables to hold output values (name them whatever you like)
  - a variable to hold the user's name
  - 3 variables for messages:
    - A prompt for name
    - A prompt for integers
    - A message for results (similar to the sample run below)
- in the .text section write instructions to:
  - prompt the user for their name and save it in memory
  - 3 times:
    - prompt user for an integer between 1-100
    - read and store the integers in a, b, and c
    - no input checking required
  - calculate ans1 = 2a - c + 4 (use a+a for 2a) and store the result
  - calculate ans2 = b - c + (a - 2) and store the result
  - calculate ans3 = (a + 3) - (b - 1) + (c + 3) and store the result
  - display the user's name and message for results
  - display the 3 results but print a space character in between for readability
- at the bottom of your program, add **comments** with test values for a, b, c and the results you expect from the program for ans1, ans2, ans3 (see sample run below). Show at least 2 sample runs.

What to turn in:

- after you test your program, upload the .asm file to eLearning

Grading Rubric:

Points	**Element**
10	Data section set up; data values and results stored
5	Get name from user
15	Get 3 integers from user
30	Calculate 3 expressions
10	Display user's name, message and results
10	Display space between results
10	Program test-data comments
10	Program contains meaningful comments and good use of whitespace

Sample Run: (you should use different and more interesting test data)
What is your name? Karen
Please enter an integer between 1-100: 10
Please enter an integer between 1-100: 20
Please enter an integer between 1-100: 30
Karen
your answers are: -6 -2 27
-- program is finished running --

Note:

- Notice that printing the name string causes a newline, but don't worry about it
- Note also that you will need to print space ' ' between numbers so that they don't run together
- If the user types anything other than an integer, you will get an exception. We haven't learned how to deal with that, so assume that your user can follow instructions.