CSI 201 — Computer Science I

Homework #05 - due February 22th, 2006

Write the following program on sampson. Hand it in by printing the source in GLDS 202 using `enscript hw5.cpp`. To receive a grade, your program MUST compile and execute on sampson in the 201 directory under the filename hw5.cpp

Remember to always cd 201 and save your work. Your program output should exactly match the sample execution shown below for full credit. Executing ./p201 will test your program against some other sample executions. To receive a grade higher than C, you should have all PASS when executing ./p201. This is a necessary but not sufficient quality to receive a grade greater than C (as you must still follow the instructions of the assignment).

In this homework, you will extend homework four but use functions. Create a function to compute the distance based on acceleration, velocity and time. Also write an output function which takes acceleration, velocity and time as inputs, but only performs the last output line in the sample executions. This output function should call the function you've written to compute distance. Inside this file, you will write a paragraph inside a multiline comment. This paragraph will describe how your program demonstrates the idea of procedural abstraction.

Homework four's description follows:

Write a program to compute the distance s an object moves according to acceleration a, velocity v and time t input by the user. The formula to compute distance from these inputs is: $s = vt + \frac{1}{2}at^2$. Negative accelerations and negative velocities are okay, but negative time does not make sense. A time of 0 seconds is acceptable. Your program should declare the appropriate variables, prompt the user for input, check for correct values in the input, compute the calculations and then output the result. Include a loop which prompts the user for an appropriate time value if a negative input is given. Sample program execution is attached.

Sample Executions:

```
Input floating point velocity in ft/s: 3.5
Input floating point acceleration in ft/(s*s): 1.3
Input floating point time in s: 4
The distance is 24.4 feet.

Input floating point velocity in ft/s: 10
Input floating point acceleration in ft/(s*s): -4
Input floating point time in s: -1
ERROR time is negative
Input positive time in s: -3.3
ERROR time is negative
Input positive time in s: 3.3
The distance is 11.22 feet.

Input floating point velocity in ft/s: 30
Input floating point acceleration in ft/(s*s): -15
Input floating point time in s: 2
The distance is 30 feet.

Input floating point velocity in ft/s: -15
Input floating point acceleration in ft/(s*s): 15
Input floating point time in s: 2
The distance is 0 feet.
```