CIS15 Fall 2007, Assignment I, Part 2

Instructions

- This is the second part of the assignment for Unit I.
- The entire assignment is worth 5 points.
- The first part is worth 2 points and was distributed and worked on in class on Monday 27th August.
- The second part is worth 3 points and will be distributed and worked on in class on Monday September 10 and Monday September 17.
- **Both parts together are due on Monday September 24** and must be submitted by email (as below).
- **Follow these emailing instructions:**
    1. Create a mail message addressed to *parsons@sci.brooklyn.cuny.edu* with the subject line **CIS15 HW1**.
    2. Attach ONLY the **.cpp** source code files created in part 1 and below.
    3. Failure to follow these instructions will result in points being taken away from your grade. The number of points will be in proportion to the extent to which you did not follow instructions... (which can make it a lot harder for me to grade your work)

1. "rand" test program

Write a program that tests the validity of the `rand()` function. Generate 10,000 random numbers. For each random number, determine if it is even (divisible by 2) or odd, counting the total number of even and odd numbers. Compute the ratio of even to odd numbers. Print to the screen the number of even numbers generated, the number of odd numbers generated and the ratio of even to odd numbers.

What should the ratio be if `rand()` is functioning properly? How does your result compare?

(1 point)

2. Unix command help program

Write a program that will provide help with basic unix commands. The program should contain a loop that asks the user to input a unix command or enter *quit* to stop the program. Your program should recognize at least 6 of the basic Unix commands we discussed in class on September 6 (e.g., `ls`, `pwd`, etc.). Your program should print out the brief description of the command the user entered. If the user enters a command that your program does not recognize, then print a message to that effect.

You must use the C++ `string` data type for your input variable.

Hint: use the Unix `man` command to get the brief description for the commands you plan to put in your program. The brief description is the text that appears in the **NAME** section of the man page, after the command name followed by two hyphens (–). For example:

```
NAME
     pwd -- return working directory name
```

(2 points)