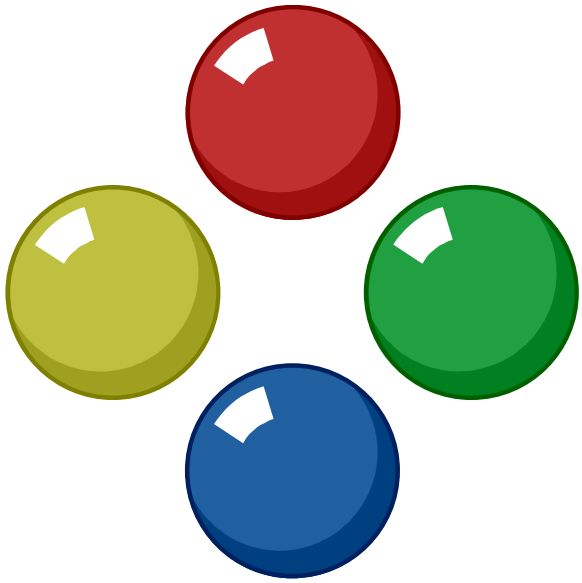
California State University, Sacramento
College of Engineering and Computer Science

Computer Science 35: Introduction to Computer Architecture

Fall 2020 – Lab 4 – *20 Points to Ravenclaw*

Overview



Student wizards and witches are taught at the famous ***Hogwarts School of Witchcraft and Wizardry***. This noble school has produced some of the finest magical students the World has ever seen.

During the school year – full of potions, spells, Quidditch, and various students being maimed – each House is awarded points by the faculty. Every time an instructor yells *"10 points to Gryffindor!"*, for example, a number of bright color balls fall into the bottom of a magical hourglass. At the end of the term, the House with the most points wins the prestigious House Cup!

These points can be earned by academics, winning contests, bravery, etc... The points can also be lost by breaking rules, poor academics, etc…

You are going to use the odd Muggle technology called "computers" to create a simple program to keep track of **two** houses. You will input the number of students that have earned/loss points in different categories. At the end, display how many points were earned. And, between the two Houses, you declare the winner!

Sample Run

The following is a sample run of the program. The user's input is printed in **blue**. The data outputted from your calculations is printed in **red**.

```
Getting an A        : 27 points
Being late to class : -11 points

How many Ravenclaws got A's?
18
How many Ravenclaws were late to class?
9
How many Hufflepuffs got A's?
12
How many Hufflepuffs were late to class?
3

Ravenclaw gained 387 points
Hufflepuff gained 291 points

Winner is Ravenclaw!
```

Information text

Prompt and input scores

Calculated output

Display the winner. There may be a tie.

Hints

- Start off by getting the first multiplication to work and print the correct value.
- Now work on each of the requirements below one at a time. You will turn in the final program, but incremental design is best for labs.

Requirements

You must think of a solution on your own. **You can come up with your own theme and categories.** You don't have to use mine. The requirements are as follows:

1. Display a table to the screen. You can create your own. *Please see above.*
2. Display a prompt, to the user, for each student count.
3. Input the number of students for each category. You need to ask the same two questions of both houses.
4. Calculate the total number of points. Tip: use a register to create a running total.
5. Output the total number of points with some helpful text.
6. Display the winner. Remember: you can have a tie.

Submitting Your Lab

This activity may only be submitted in Intel Format.
Using AT&T format will result in a zero. Any work from a prior semester will receive a zero.

Afterwards, run Alpine by typing the following and, then, enter your username and password.

```
alpine
```

Please send an e-mail to yourself (on your Outlook, Google account) to check if Alpine is working. To submit your lab, send the assembly file (not **`a.out`** or the object file) to:

```
dcook@csus.edu
```

UNIX Commands

Editing

Action	Command	Notes
Edit File	**nano** *filename*	"Nano" is an easy to use text editor.
E-Mail	**alpine**	"Alpine" is text-based e-mail application. You will e-mail your assignments it.
Assemble File	**as -o** *object source*	Don't mix up the *object* and *source* fields. It will destroy your program!
Link File	**ld -o** *exe object(s)*	Link and create an executable file from one (or more) object files

Folder Navigation

Action	Command	Description
Change current folder	**cd** *foldername*	"Changes Directory"
Go to parent folder	**cd ..**	Think of it as the "back button".
Show current folder	**pwd**	Gives the current a file path
List files	**ls**	Lists the files in current directory.

File Organization

Action	Command	Description
Create folder	**mkdir** *foldername*	Folders are called directories in UNIX.
Copy file	**cp** *oldfile newfile*	Make a copy of an existing file
Move file	**mv** *filename foldername*	Moves a file to a destination folder
Rename file	**mv** *oldname newname*	Note: same command as "move".
Delete file	**rm** *filename*	Remove (delete) a file. There is **no** undo.