Menu Option Descriptions for Project 1 Part 2

A. Add a Person
Authors: David Crippes, Nick Morrill, Tandra Felly, Kevin Drake

The addAPerson function is going to create a new person and add a pointer to that person to a vector full of person pointers for the university. The function takes an int for ID and a string for name, and then a string for person type. The function does not return anything (void). After the input is taken we will determine the person type with if's or a switch. Once the person type is determined the appropriate constructor will be called and then a pointer to that person will be added to a vector of SSUPersons.

Option B:
Authors: Matt Hardwick, Tim Jianming, Matt Johnson, Eric Wooley

* Return a boolean, which is true if the person was deleted
* Prints a report if Courses which are now Instructorless, or staff members without supervisors
* Delete any references of the person
* Delete the object
* Delete from Vector

Option C: Print a team list
Authors: Brad Tayler, Steve Klamm, Adam Sownie, Jolie Nazor

Prompt user to enter sport print students based on type "Athlete" specific sport entered by user

d. Delete a Course
Authors: Kristi Yost, Reid Sobotka, Maddy Heit, Yun Jie Xu

The function will take in a Course*
We need to include iostream and vectors at the top and make sure that a deconstructor is created inside of the class Course.
FIRST:
to delete this course from the vector of courses the faculty teaches and the students that are enrolled in the course; we will do the following:
We will iterate through every student's and faculty's vector of courses to check to see if the course is in that vector
If the course pointer that we are looking at in the vector is equal to the course that is passed to the function we will use the vector function erase to delete that course from the vector.
Delete the course by calling the Course deconstructor.

MAIN IDEA:
Get the course
go through all of the students
remove from students, remove from faculty, and then delete the course last.

**Option e: Add a student to a course:**
Authors: Lauryn Loudermilk, Jake Neugass, Brandon Mondo, Chris Gonzales

- ask the user for a student id and course id.
- check if the course is already in the student's Course vector. If it is, do nothing. If it isn't, do the following.
  - push_back the course into the student's Course vector
  - add the courses unit count to the student's current enrollment
  - if this pushes the student over 7 units, change their tuition to reflect this
  - increment the enrollment of the class by one

**f. Drop a student from a course**
Timothy Doughty Gonzalo Parajon Ryan Marnell Scott Arnold

- Remove the COURSE object from the STUDENT object
- Go to STUDENT object
- Delete COURSE object from vector of COURSE objects via the ID attribute
- Decrement the enrollment attribute in the COURSE object
- No STUDENT objects saved in the COURSE object, so nothing to do with the COURSE object

**Option g:**
Authors: Will Raus, Seth Roelke, Kyle Tripp, Chris Duran

When selected, the option will ask the user for the ID number of the student.
It will then find the student with the matching ID, and retrieve them from the vector of students.
After that, a menu will pop up, allowing the user to input a new value for any of the values in the student class: name, ID, e-mail, major, completed units, GPA, Tuition, and Balance.
It will also have an option to add/remove a course for the student.

**H. - Print a Roster for a Course**
Authors: David Crippes, Nick Morrill, Tandra Felly, Kevin Drake

The printARosterForACourse function will take a course as input and print out an alphabetical list of names and corresponding ID's for each student in the course.
A for loop will go through the vector of students in the course and output the information to a text file.

**Option I:**
Authors: Matt Hardwick, Tim Jianming, Matt Johnson, Eric Wooley

* Print a report sorted by department, with a header for each
* Sorted by id
* Includes id, name, title, person type, email, salary
* Total Salaries for Department

**Option J: Add Course**
Authors: Brad Tayler, Steve Klamm, Adam Sownie, Jolie Nazor
Prompt the user for info
have a check to department - list departments
list instructors in that department
have user input instructor ID
check that the course isn't being taught at this time
proper connect the course + instructor so that they both have the most
up to date information
return course object

k. Print a List of Students in a GPA Range
Authors: Kristi Yost, Reid Sobotka, Maddy Heit, Yun Jie Xu

Using the vector of students (previously created) we will look at every student's GPA.
If the GPA is in the range we will add that student to a new vector.
Then we will sort that vector that we created, by GPA in decending order (best to worst students).
And then finally we will print the vector.

Option l: Print a student's schedule:
Authors: Lauryn Loudermilk, Jake Neugass, Brandon Mondo, Chris Gonzales

- take as input a vector of student pointers and vector of class pointers
- ask the user for a student id
- iterate through the student vector in search for the student,
- construct an stream object containing the course info
- print out the stream object

m. Cancel low enrolled courses
Timothy Doughty Gonzalo Parajon Ryan Marnell Scott Arnold

- DETERMINE LOW ENROLLMENT LEVEL (ask Dr. Watts if a limit exists or find avg of all
courses, Dr. Watts' answer: fewer than 5 students.)
- Traverse all STUDENT objects and call studentDrop (or whatever the function is called) to remove
that COURSE object from the STUDENT object.
- Delete the COURSE from the FACULTY object as well
- Delete the COURSE object itself

Option n:
Authors: Will Raus, Seth Roelke, Kyle Tripp, Chris Duran

The function will iterate over the vector of students, checking their GPA.
If a student has a GPA below 2, it will add them to a vector of students on probation, then remove
them from all of their classes.
Finally, it will print the new list of students on probation.