1. Create a class to represent each basic entity
   1. Student class
   2. Course class
   3. Enrollment class
2. Course class

Pointer to float grades to create an array that allows for reallocation if space for grades isn’t enough

Should have private variables

ID (int)

name (string or c-string)

location (also string or c-string)

meeting time (string or c-string)

instructor (string or c-string)

Need mutators and accessors for each one

Accessors to be able to print out a list of all courses

Mutators for assigning initial values

Each accessor needs to be able to be looped

Functions should probably be

Int getCourseID();

String getCourseName();

String getCourseLoc();

String getCourseTime();

String getCourseInstr();

Void setCourseID(int newID);

Void setCourseLoc(string newLoc);

Void setCourseTime(string newTime);

Void setCourseInstr(string newInstr);

Collection of courses should be dynamically sized array, use realloc everytime chunksize is exceeded

1. Student class

Should have private variables

ID (int)

name (string)

classification (string? Possibly enum for organization)

Need mutators and accessors

Each accessor loopable

Tentative function names:

Int getStudentID();

String getStudentName();

String getStudentClassif();

Void setStudentID(int newID);

Void setStudentName(string newName);

Void setStudentClassif(string newClassif);

Constructor seems unnecessary

Collection needs to be dynamically sized array, use realloc everytime chunksize is exceeded

1. Enrollment class

Needs private variables

studentID (int)

courseID (int)

enrollmentID (int)

grades (int[10])

numberGrades (int to represent how many grades in array currently)

letterGrade (char)

averageGrade (float)

needs mutators and accessors probably

tentative names:

void setEnroll(int id)

set enrollmentID = id;

void courseEnroll(int id)

set courseID = id

void studentEnroll(int id)

set studentID = id (actually could probably just use constructor for these)

int getStudent()

return id of student, can be used to find student’s name

int getCourse()

return course of enrollment, can be used to find course’s name using functions in course class

void getGrades();

prints out the values from the grades array

void averageGrades()

make temporary int

make temporary result int

loop this.grades[i] with i++, until i has reached numberGrades, and add to temp

divide temp by numberGrades, type\_cast to float at the hundredths position

set result = type cast temp to int

if result >= 90, set letterGrade = ‘A’

repeat for 80->B, 70->C

void addGrades(int newGrades)

newGrades represents number of input grades

for (int j=numberGrades+1; j < (newGrades+numberGrades); j++)

cin >> grades[j]

numberGrades++;

incrementing number of grades for next time grades are added

deconstructor is unnecessary

remember to check for if course has less than or equal to 48 students or student has less than or equal to 5 classes

1. Collection class for students

Needs private variables:

Int studentCount;

Int studentCap;

Student \*studentList;

Tentative function names:

Void addStudent();

Void printStudents();

Void saveStudents();

Void loadStudents();

Constructor needs to set studentCount = 0, studentCap = CHUNKSIZE, and studentList = new student[CHUNKSIZE]

Deconstructor needs to delete [] studentList

1. Collection class for classes

Should mirror collection class for students by rename variables to refer to classes instead

1. Collection class for enrollments

Should mirror other collection classes but use variables that refer to enrollments instead

1. Main function

Should contain checking for if course being enrolled has less than 48 students or if student has less than 5 courses

Should also print out a menu function and use numbers and switch cases to determine which functions to use