2019 UT REGENTS' OUTSTANDING TEACHING AWARD APPLICATION

Martine Ceberio, Ph.D.
Associate Professor
Department of Computer Science
The University of Texas at El Paso

Summer Contact Information:
602 Bluff Canyon Circle
El Paso, TX 79912
Phone number: (915) 355 8293
Email: mceberio@utep.edu

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Teaching has been a passion of mine for as long as I can remember. This dossier covers the work I have done in most recent years in teaching and conveys the passion that drives me in guiding and accompanying my students to their next educational and professional levels.

Throughout my career, I have had the opportunity to teach a varied set of courses at all levels of the computer science undergraduate and graduate curricula, of both applied and theoretical content. In this dossier, I focus on the initiatives I have conducted and led over the last 4 or so years on the early years of the computer science undergraduate curriculum at UTEP.

Healthy recruitment and retention are the golden goals of all educational programs; Computer Science is not immune to that. In addition, in Computer Science, a special focus has to be placed on women as they join our programs in lower numbers (about 18% of CS students are women) and are more at risk of leaving before graduating. Throughout my tenure at UTEP, I have been active in the local community, reaching out to schools (teachers, students, administrators) to enhance our recruitment efforts. But this is only half the work: once students join our program, their experience through the first few semesters in our program is critical to their retention. In Fall 2014, I was asked to revisit our introductory course (Introduction to Computer Science: CS1401 and later changed to CS1301/1101; also dubbed CS1) to enhance retention. Shortly after, I became chair of the introductory course sequence curriculum committee in my department. This has allowed me to effect multiple changes in CS1 (transformative at first, and then more and more incremental following our formative assessment of the success we had) and to propose curriculum additions (in the form of one-credit-hour courses), among which a problem solving courses sequence and a computer-science-owned version of discrete mathematics. This dossier provides details about the rationale for and description of these new courses and course changes.

My contributions to curriculum and my passion to enhance curriculum are fueled by my curiosity and love of problem solving. These have led me to seize many opportunities to learn more about teaching and learning (through professional development workshops and seminars, and a lot of reading and digging), which not only informed my teaching practice but also took me to the path of revisiting my teaching philosophy. In this dossier, I present how this change came about and what my philosophy now consists in. In short, it is inspired by Dr. Dweck's growth mindset research, is heavily based on asset-based teaching, and promotes empathy.

The voices of my students and peers appear in this dossier, through their letters, to complement my own description of my efforts and initiatives. In addition, students' comments and assessment of my teaching practice are provided to complete the picture, and evidence is provided of my continuing involvement with students outside the classroom, through mentoring of undergraduate students and high-school students in my research lab, to advising student organizations, to informal interactions with and advising/mentoring of many more students.

Although it focuses on my most recent years of teaching, this dossier is a testimony to my continuous teaching efforts and it provides a snapshot of who I am as an educator. I trust that my drive for continuous teaching improvement will never cease or even decrease. However, I am positive that my philosophy will keep evolving as years go by and I keep discovering more about how to best support my students.

TEACHING PORTFOLIO

1. Teaching Philosophy

I have loved teaching for as far as I can remember. Solving problems and teaching. I see teaching as a challenge of multiple facets: our students are so varied, with different backgrounds, interests, aspirations; and yet, our objective is to help them all to meet the same performance bar, and even surpass it. I am a firm believer that anyone can learn: regardless of their preparedness, we can work with students and help them grow into any topic / any skill. It may be challenging, yes, but nothing of interest and impact comes easy.

I have been a faculty in Computer Science at the University of Texas at El Paso since 2003, an associate professor since 2012. During these years, I have embraced UTEP's mission of **Access and Excellence**, pursuing a research agenda of quality, as demonstrated by my publications and funding record (see my curriculum vitae for more details), and engaging students in research and in community building to enhance recruitment and retention in our computer science program.

My work as an educator at the University of Texas at El Paso has allowed me to combine all of my passions (research, teaching, service to multiple communities), but it has primarily allowed me to grow as an educator with the added reward that is the huge potential for impact and social mobility our instruction bears on our students. There is no better feeling than going to work and knowing you can make a difference!

In El Paso and in my area of Computer Science, the differences I seek to make everyday are about:

- **Retaining students who choose computer science** but are likely to be discouraged by a variety of factors, including outside constraints (family obligations, heavy work schedule, etc.), stereotypes (about who should be a computer scientist and who should not: women are highly at risk of discarding themselves), perceived inadequate preparedness.
- **Recruiting students to computer science**, or at the very least informing them about the field so that they can make an informed decision that would otherwise be left to stereotypes and clichés.
- **Inspiring students to go above and beyond** what is required in their degree and making them believe that they can achieve great successes.

To this end, I have focused my work on activities aimed to enhance recruitment and retention of students in our computer science department, for instance by creating an ACM-W student chapter at UTEP in 2012, by leading the NCWIT AiC El Paso affiliate from 2011 to 2018, by designing computer science summer camps, and by hosting high-school students as research interns in summer since 2010. I have been involved in projects that aim at understanding the challenges faced by Latinas in engineering (NSF Research on Gender – $524,900), and enhancing the retention of our majority-minority students in computer science (NSF IUSE/PFE RED –$1,992,592). I also spend a substantial part of my time reaching out to teachers of the community to inform them about opportunities for their students, to propose ideas but also help them identify how to integrate computational thinking in their classrooms.

In my scheduled teaching, I constantly seek improvement, attending many professional development workshops and conferences (about 23 over the last 6 years), adjusting my pedagogy accordingly, and creating new courses. Over the last three years, I have taught a number of classes, mostly undergraduate-level courses because of the need to cover a list of required courses and to give way for junior colleagues to teach graduate courses. Most of my teaching has consisted of the following undergraduate course: Introduction to Computer Science (a.k.a. CS1, taught every semester since Spring 2015), but I also created two new courses on Problem-Solving (one 3-credit-hour course cross-listed for undergraduate and graduate students, one 1-credit-hour course only for undergrads). The 1-credit-hour course development is a project in collaboration with Google and CAHSI (Computing Alliance for Hispanic-Serving Institutions). The idea of creating and

offering 1-credit-hour courses is the result of working with the CS department's curriculum committee within our department's NSF IUSE/PFE RED (Revolutionizing Engineering Departments) project to offer more options to CS students as early as when they are in their first semester to enhance their motivation in the program while strengthening their essential problem-solving skills. Specifically, the two above-mentioned new problem-solving courses emerged from an extra-curricular initiative I offered to students taking our Introduction to Computer Science course (CS1401, which became CS1301/1101): in spring 2015, I started offering a 1-hour weekly problem-solving club, which proved to be very effective in developing our students' awareness of their already existing abilities and equipping them with more problem-solving techniques. Similarly, our department recently decided to pilot an initiative around Discrete Math, which our students usually take from UTEP's Math Department as a sophomore-level 3-credit-hour course but struggle with. Instead, we decided to break it into one 1-credit-hour and one 2-credit-hour course to be taken concurrently with our two introductory computer science courses (CS1301 and CS2401) as an attempt to better support our students' learning and decrease their struggle and fail rate. I put together these two courses and am monitoring the offering of the first 1-credit-hour course in fall 2018.

Mentoring is a key part of my teaching activities. In addition to the students I formally teach, each semester, I mentor at least 5 undergraduate students conducting research with me (in my lab CR2G, see: cr2g.constraintsolving.com). I informally mentor the students from the student organizations I advise: the ACM-W chapter at UTEP until summer 2018, UTEP's SIAM student chapter, Harmony Miners Association, CLIO (a CS students' organization that focuses on CS outreach to local schools). I also advise several groups of high-school students in town and I regularly give talks to the community about computer science to better inform then about the field. Finally, I regularly give talks to and interact with teachers of our community to share with them ideas about how to integrate computational thinking in their classroom and also to ease our students' transition to the University.

Overall, I see myself as a resource to our students and I catch every opportunity to mentor any student I meet: asking about the courses they take, connecting them to the help they may need, informing them about internships and other extra-curricular relevant opportunities, telling them about research and facilitating their meeting with a mentor of their choice (possibly inviting them to my research group), etc. For me, teaching and mentoring are a way of life, not something that only takes place in a classroom or in formal settings.

2. Course Material

In what follows, I share course material about three courses I have taught over the past three years:

- CS1301/1101: Introduction to Computer Science: These are two pieces of a same course (lecture and lab)
- CS4365: Problem Solving: Summer course I designed in 2016 as a result of creating a problem-solving club in spring 2015
- CS1190: Problem Solving: The course I designed as part of an effort with the Computing Alliance of Hispanic-Serving Institutions and Google

CS1: CS1301/CS1101 Introduction to Computer Science (lecture and lab)

This course (as a group: CS1301/1101) is the first required computer science course taken by all computer science majors. There are usually three sections offered each semester. I have taught one section every semester since spring 2015, two sections in fall 2018. Effectively teaching this course is crucial for our ability to retain students. I contributed to its revamping when I came back to teaching it starting in spring 2015 and I brought multiple improvements to it since

then along with my colleagues who teach other sections of it. I found inspiration for revamping this course from the literature but also from attending professional development workshops and programs. Improvements include:

- Refocusing the course emphasis on problem solving (vs. coding): This course is an introduction to computer science and students often think that they will solely learn how to code in this course. Instead, we aim to develop their algorithmic thinking / problem-solving skills, and coding is really only a mean to an end: we solve problems and we happen to use computers to solve these problems, hence the need to learn how to code.
- Engaging students beyond the classroom: as a way to increase retention. Studies show that students, who feel more engaged with the department, with the major, with the students of the major, perform better. We started including points for student engagement in the final grade, along with examples of engagement activities that students can do to fulfill the requirement. A result of this initiative, we have seen an increase in students' engagement in CS student organizations, in research, in seeking internships.
- Using an online textbook with homework tracking ability: Faced with growing enrollments and aiming to be able to provide quality feedback and accompaniment to our students, using an online textbook (zybook in our case) has allowed us instructors to be able to track the amount of reading and homework assignments completed by each of our students individually. This has proven to be an invaluable tool for us to be able to identify students who might be struggling and contact them to offer individually relevant and timely help.
- Splitting the lecture from the lab: Starting in Spring 2017, the original CS1401 was split into CS1301 and CS1101. This is the result of realizing that our students tend to struggle particularly with the lab assignments, not because they lack the skills, but often because they lack the ability to dedicate timely efforts to it. As a results, our students often were failing the lab portion of CS1401, hence the whole CS1401, while all they were failing to demonstrate was their lab skills. We felt that having them retake the 4-credit-hour course was both unnecessary and a reason why some students, after failing CS1401, were not coming back to retake it and dropped out of the CS program altogether.
- Active learning activities: We started by adopting a flipped-classroom style, and then backtracked to a mostly hybrid style with mini lectures and cooperative learning / group activities where hands-on activities are given to students. All of our courses (in lecture) now follow this style where students are put in teams, taught early on how to function in teams (with roles and expectations), and have to actively engage in their learning with, depending on the lectures, activities that they pick within a topic or activities that are assigned to them.
- Ensuring that lab assignments are engaging and relevant to our students: Showing the relevance of computer science to everyday life and more broadly to society is shown to help retain students in the major (in particular women and URM students). Making sure that our lab assignments were fun and relevant has been a focus of the redesign of CS1301/1101 (dubbed CS1).

I wrote a blog piece about my major teaching philosophy change in CS1301/1101 (see: http://martineceberio.fr/blog/cs1-philosophy-change-a-kind-approach).

More recently, based on my experience:

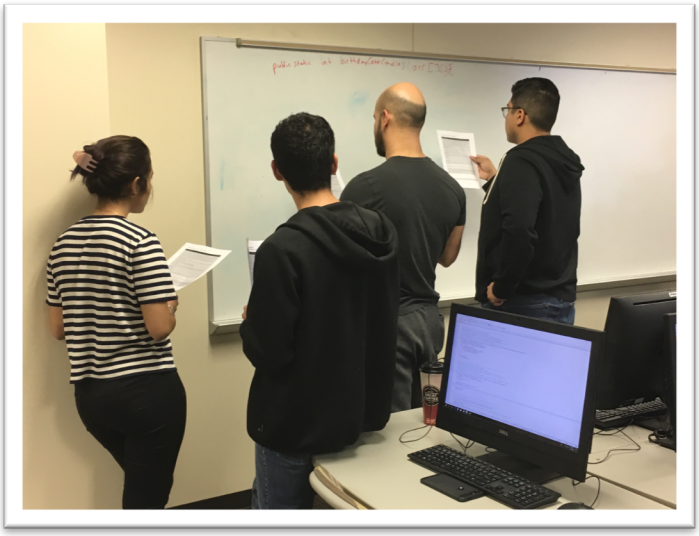
- being part of UTEP's STEM Accelerator program and professional development workshop, with Olin college (2016-2018);
- participating in the NSF workshop about Rethinking Engineering Education at HSIs, put together by faculty from UTEP, the University of Miami, and Olin College (March 2018); and
- being a Faculty in Residence at Google in June 2018;

I brought new improvements to CS1301/1101. Among these changes are:

- Creating an atmosphere conducive to team work and respect: Ice-breaking activities I participated in at Google this summer were very successful and managed to create team bonding and prevent the fear of failing. These activities were build on games in which we are bound to failing, hence de-dramatizing the fear of failing and making failing part of the norm: in computer science, building resilience to failing is essential to staying and being successful in the program. I wanted to bring this to my teaching, all the more in CS1301/1101 where the DFW rates are usually so high. I also asked students (who felt comfortable doing it) to share their life paths on drawings they then presented to the class: I did the same and presented mine to them.
  Additionally, based on my experience completing lab assignments at Google where our instructors were playing music during the sessions, I brought this idea to my CS1101 lab and we are now conducting labs with light background music. I find my students to be more relaxed and more likely to stand up and go collaborate with others.
  It is hard to quantify the effect of a change on its first attempt, and while the semester has not ended yet; however, in this semester, I have experienced a higher student engagement and a lower sense of fear in my students (to fail and to seek help) than in previous semesters.

- Weekly problem-solving team activity – Khôlles: Following up on my idea of strengthening team-bonding feeling in my course, I believed that building expectations and habits in students would help them feel connected and build identity in belonging to this course. I was inspired by my own experience as a student when I had weekly oral individual examinations, called "khôlles" (one student examined by one professor): this was scary and hard, but in my cohort of students, we all knew this was coming weekly and it became part of the culture of our degree. I wanted to try and recreate this in my course, but in a much less threatening and painful way. During my stay as a Faculty in Residence at Google in June, I elaborated a plan for a weekly team activity to be held in lab. Starting in fall 2018, all students registered in my CS1101 lab come once a week to lab and collectively work on a problem-solving activity (that involves coding on a board).
  - Group work and roles: Each group is composed of 8 students. There are 4 roles in the team (helper, problem solver, reviewer/note taker, tester). As a result, there are 2 students in each role, which allows not putting any single person "on the spot". Roles are clearly defined with specification of what each role should accomplish and guidelines for self assessment. Each pair of students on a given role rotates each week to the next role, to ensure that students will get to practice each of the roles after 4 weeks.
  - Benefits of this activity: I see multiple benefits to this activity. First, in this course, it is very common that students are intimidated. Working in pairs help them not feel pressured individually and it puts them in a position to have to collaborate (I tested this approach at Google in the context of going through several mock interviews with a fellow faculty in residence). Often too, students feel that they cannot write proper code. Being able to watch someone else develop code helps them put their own performance into perspective. Code review (which happens when you watch someone code, or read code written by someone else) is also an essential skills that is always mentioned by companies recruiting our students. Testing is another one of the essential skills our students need in industry: starting early is placing it on their radar is crucial to their success. Usually, a problem solver will test his or her approach. In Khôlles, I have deliberately split the role of problem solver into a mere problem solver and a tester to make clear that testing is important and that it is a role in itself (even if in practice often fulfilled by the same person). Finally, the expectation we build for the students who come every Friday to lab for a Khôlle is further conducive of team building and accountability: each student is part of the same team throughout the semester and cannot let the fellow team-members down. I kept the name "Khôlle" as a nod to my past as a student.

Below are pictures of students working on their weekly khôlle.



CS1301 syllabus for Fall 2018:

CS1301 Introduction to Computer Science

Course Objectives: Students will learn to be active learners, understand the motivations for computing, basic concepts of algorithms, basic computer organization, and impacts of computing.
They will develop problem-solving skills, implement solutions to computing problems in a high-level programming language, and build team skills, critical-thinking skills, and professionalism.
Prerequisite: MATH 1508 or MATH 1411 with a grade of C or better.

Knowledge and Abilities Required Before Entering the Course: Students entering the course are not required to have a background in Computer Science or programming. They should be familiar with topics from Pre-calculus, including algebraic functions, proofs, and base representations of numbers.

Topics covered this semester: The semester will be structured in four phases.

- During the first phase (first 3 to 4 weeks), we will cover algorithms, abstraction, memory and variables, including arrays. We will also go over conditionals and repetitions as they appear in our daily lives.
- In the second phase (next 3 to 4 weeks), we will introduce how conditionals and repetitions as loops can be used in algorithm design and in java, blending this with methods.
- During phase 3 (next 4.5 weeks), we will introduce recursion and will practice integrating the essential components presented in phases 1 and 2 with recursion.
- Finally, in our last phase, we will learn about user-data types and introduce the use of linked lists.

Logistics:

Lecture sessions: MW 9 a.m. -10:20 a.m. in CCSB 1.0704
Or MW 10:30 a.m. -11:50 a.m. in CCSB 1.0704
Instructor: Dr. Martine Ceberio – mceberio@utep.edu – office room: CCSB 3.0406
Office hours: MW from 1:30 p.m. to 2:50 p.m.
+ **by appointment & open-door policy**

Textbook: *Programming in Java*, by Zybooks, available at zybooks.zyante.com. To subscribe to your textbook, please enter the following code:

UTEPCS1301-01CeberioFall2018

If you are scheduled at 9 a.m.

UTEPCS1301-02CeberioFall2018

If you are scheduled at 10:30 a.m.

Notebook: All students will be required to use a paper notebook in which they are expected to take handwritten notes of the lectures. These notebooks will be collected regularly for assessment and grading.

Communication platform: This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.

Find our class page at: piazza.com/utep/fall2018/cs1301

Software: Software used in this course is available on the desktop computers in the main computer lab and in the two instructional labs on the first floor. To use the course software on your home or laptop computer, instructions will be given in the labs and available online on our piazza page.

Note: You should be enrolled in one lab section. Do not drop in on a lab or lecture section other than yours without prior approval from your instructor.

Grading

Grades are communicated to students in a timely manner. It is the students' responsibility to keep track of their grades by compiling the grades they receive. Your semester grade will be based on a combination of homework assignments, weekly quizzes, class participation, 3 mid-term exams, student engagement, and a final exam.

The approximate percentages are as follows:

- 10% Homework
- 20% Quizzes, including in-class assignments
- 65% Exams (3 mid-term exams and 1 final exam)
- 2% Student Engagement in Computer Science (to be completed before the end of week 11)
- 3% Class participation (includes on-time lecture attendance, active participation in class, completion of any quizzes for attendance and survey purposes)

The nominal percentage-score-to-letter-grade conversion for CS 1301 is as follows:

- 90% or higher is an A
- 80-89% is a B
- 70-79% is a C
- 60-69% is a D
- Below 60% is an F

Note: Regardless of your standing in the class at that time, you need to earn a 65 or better at the final exam to pass the course. Additionally, you must earn a C or better in each of these two courses, CS1301 and CS1101, to continue to the next course in this sequence, which is CS2401.

Expectations

Class Participation: Attendance at and participation in all lecture sessions are critical factors of your success in this course.

Students should be on time for all scheduled sessions and **attend the entire session**. Attendance will be taken at every session (at first you will have to sign in but as time goes the instructor will know you and mark you present without your help) and will count towards your class participation grade.

Students should notify the instructor prior to missing a session if at all possible, and certainly right after if earlier was not possible. The instructor will allow two unexcused absences per semester before having the option to deduct points from the final grade (5 points per subsequent unexcused absence).

It is the student's responsibility to obtain the content covered during missed class(es). Participation points also include completing post-lecture and post-labs online quizzes (when requested) that are administered as surveys to monitor students' overall progress and potential struggles.

Quizzes: The purpose of each **quiz** is to ensure that you are staying current with the weekly reading assignments and video lectures and to verify that you have acquired the skills developed in class. Quizzes are unannounced. They usually will usually be on-line quizzes on socrative.com. There will be **no make-up** on missed quizzes.

As part of the quiz grade, the students will have to turn in their **paper notebooks** when asked so that the instructor can assess their notes (clarity, readability, usability, correctness, and completeness). The grade assigned for notebooks will count as 30% of the quiz grade. Note: special accommodations will be considered if a student cannot take notes because of a medical condition.

Finally, there will be unannounced **in-class assignments**, to be turned in either by the end of the class or within a short period of time after the class (details will be given for each assignment). There will be **no make-up** for missed in-class assignments. Grades of such assignments will weigh equally with grades from online quizzes.

Homework: **Reading and homework assignments** will be announced in class and/or posted on piazza (under the Homework section of Resources). If you miss a lecture session, it is your responsibility to find out what you missed. You should expect to spend **at least four hours per week outside of lecture** on reading and homework. Most of your homework will be work assigned on your online zybook: all deadlines are already available on your zybook so that you can plan ahead. Completing the assigned activities on time will be crucial to your success in the class (since these activities prepare you for classwork) and to getting a good grade (since late completion will be penalized). In addition to the homework deadlines, extra deadlines will be set, which are meant for you to be able to earn extra points on homework. Similarly, if you were not able to complete a homework assignment at some point during the semester, you will have two chances to make up for some of the missed points. This is to acknowledge that things happen and that we are here to support you.

Exams: There will be **3 midterm exams** and **one final exam**. All four exams together will weigh 65% of your overall final grade for CS1301. Because the exams contribute so heavily to your total grade, it is vital that you do well on them. If you have test-taking difficulties in general, or if you have difficulties with our tests in particular, please come let me know as soon as possible and/or request appropriate accommodation from UTEP's Center for Accommodation and Students' Services.

The purpose of the **midterm exams** is to allow you to demonstrate mastery of course concepts covered thus far during the semester. Mid-term exams will take place during the regular lecture session and are tentatively scheduled to be held around week 6-7, week 11, and week 14. Make-up exams will be given only in extremely unusual circumstances. If you must miss an exam, please meet with an instructor, BEFORE the exam.

The **final exam** will be comprehensive. You must score 65% or better on the final exam to pass this course. You must take the final exam during the time shown in the schedule for the lecture section that you normally attend. Do not "drop in" to another section: there will not be a copy of the exam for you. This is University policy. If you have a scheduling conflict (e.g., if you are taking a final at EPCC) or if you are scheduled for three final exams in one day, see your instructor in advance for accommodation. The final exam schedule is available online. It is the students' responsibility to keep informed.

Student Engagement in Computer Science: During the course of the semester (before week 10 for credit), you must engage as a computer scientist in activities as shown below, in a way that you cumulate at least 2 points (towards your final grade). Possible activities (along with the number of points each yields) include (but are not limited to – check with Dr. Ceberio if you'd like to do something that is not on the list):

- **0.5 points** for each of the following:
  - Write a summary of a seminar you attended (proof of attendance needs to be provided as well);
  - Attend two review sessions provided by your undergraduate TAs or peer leaders before exams;
  - Participate in a Department's open house as a volunteer student;
- **1 points** for each of the following:
  - Design a video about a specific career in Computer Science;
  - Write a summary of a book / chapter, agreed upon with Dr. Ceberio;
  - Write an essay about a specific research area in Computer Science;
  - Be an active participant in Google IgniteCS program (or equivalent);

- Be an active undergraduate researcher in one of the Computer Science Research labs.

Note that these points should have been acquired **by the end of week 11** of the semester. No submission for credit will be accepted past this deadline.

Standing in the course

Special Assignments: will be given to students if deemed necessary, which will need to be completed to ensure that said students remain in the class and be successful. These will be designed to help students grow into the course and develop the necessary skills. It is important that students feel free to ask their instructor about any such opportunity as well so that a special plan of development into CS1301 be tailored to them.

Standing in the Course: Students will have access to their grades for all assignments so that they can self monitor their standing and progress. However, it is also completely fine for any student to come and talk to their instructor about their standing and work together to make sure the student is as successful as can be.

Resources

Special Accommodations: If you have a disability and need classroom accommodations, please contact the Center for Accommodations and Support Services (CASS) at 747-5148 or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at www.sa.utep.edu/cass. CASS' staff are the only individuals who can validate and if need be, authorize accommodations for students with disabilities.

Scholastic Dishonesty: Any student who commits an act of scholastic dishonesty is subject to discipline. Scholastic dishonesty includes, but not limited to cheating, plagiarism, collusion, and submission for credit of any work or materials that are attributable to another person.

Cheating is:
Copying from the test paper of another student
Communicating with another student during a test to be taken individually
Giving or seeking aid from another student during a test to be taken individually
Possession and/or use of unauthorized materials during tests (i.e. crib notes, class notes, books, etc.)
Substituting for another person to take a test
Falsifying research data, reports, academic work offered for credit

Plagiarism is:
Using someone's work in your assignments without the proper citations
Submitting the same paper or assignment from a different course, without direct permission of instructors
To avoid plagiarism, see: http://sa.utep.edu/osccr/wp-content/uploads/sites/8/2012/09/Avoiding-Plagiarism.pdf

Collusion is:
Unauthorized collaboration with another person in preparing academic assignments

Important! When in doubt on any of the above, please contact your instructor to check if you are following authorized procedure. Also, please check the UTEP's Handbook of Operating Procedures at:
https://admin.utep.edu/Default.aspx?tabid=73922.

Detailed Learning Outcomes

Level 1: Knowledge and Comprehension. Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. On successful completion of this course, students will be able to describe, at a high level:

1.The history of computing
2. The relation between computing and society, including social, ethical, and legal issues
3. Computing as a profession, from required knowledge and skills to major career options
4. The relation between computing and society, including main social, ethical, and legal issues
5. Computer representation of simple data types and operations, including operations with binary numbers
6. Differences among programming languages
7. Pseudocode of the use of Multi-D arrays
8. Pseudocode of the use of Linked lists

Level 2: Application and Analysis. Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details. Upon successful completion of this course, students will be able:

1.To analyze problems and express solution algorithms in pseudocode, including a correct use of:
   a. Arithmetic and logical expressions
   b. Simple I/O operations
   c. User-defined subprograms, including recursive methods
   d. User-defined types
2. To use testing and debugging strategies, including black-box and white-box testing, test drivers, stubs and test suites, to identify software faults
3. Use teamwork roles and methods in the classroom

Level 3 Outcomes: Synthesis and Evaluation. Level 3 outcomes are those in which the student can apply the material in new situations. This is the highest level of mastery. On successful completion of this course, students will be able to use the syntax and semantics of a higher-level language to express solutions to programming problems, including the pseudocode correct use of:

1. Basic variable types such as integer, real number, character, string, 1-D array
2. Assignment, arithmetic, and logical operations
3. Basic control structures: if-then, for-loop, while-loop

CS1301 Assessment Material:

In this assignment, given in lecture (with computers available), students are put in a very practical and common situation as a computer scientist: fixing code. They are given code that does not perform as expected and are required to fix it and to provide test cases to show that the fixed code now performs as expected.

Such an activity is central to the work of a software engineer and even one that is used by Google to regularly train its own software engineers (during what they call "Fix-It" weeks).

```
/*****************************************************************************
 * Welcome to Your First FIX IT Week!
 * In this file, I need help. My methods are not doing what they should
 * and I need you to help me fix them! Will you help me?
 *****************************************************************************/
/* NAME:
 * CS1301
 *****************************************************************************/

public class fixMyCode {

    /*************************************************************************
     * this method takes an array of integers and returns the average value of
     * its elements: the result is a double
     *************************************************************************/
    public static double average(int[] A) {
        int sum = 0;
        double average;

        for (int i=0; i<A.length; i++)
            sum = sum + A[i];

        average = sum / A.length;
        return average;
    }

    /*************************************************************************
     * this method takes an array A of doubles and a double d
     * and returns true if d is found within A
     *************************************************************************/
    public static boolean find(double[] A, double d) {
        if (A[0]==d) return true;
        else return false;
    }

    /*************************************************************************
     * this method takes an array A of strings
     * and returns the length of its longest string element
     *************************************************************************/
    public static int longest(String[] A) {
        int max = 0;
        for (int i=0; i<A.length-1; i++) {
            if (A[i].length() < A[i+1].length())
                max = A[i].length();
            else
                max = A[i+1].length();
        }
        return max;
    }
```

```
     /*************************************************************************
      * this method creates a new string made of every other character of the
      * input string, and returns it
      *************************************************************************/
     public static String everyOther(String str) {
         String result = "";
         for (int i=0; i<A.length; i+=2)
             result = result + str.charAt(i);
         System.out.println(result);
     }

     /*************************************************************************
      * this method takes an array A of doubles and a double d
      * and returns true if d is found within A
      *************************************************************************/
     public static boolean find(double[] A, double d) {
         for (int h = 1; h < A.length-1; h+1) {
             if (A[1]==d) return true;
             else return false;
         }
     }


     /*************************************************************************
     * Main method where the above methods will be tested
     *** In this part of the file, we test the above methods.
     *** One test per method is provided to you
     *** Please add at least three more tests per method to either show the flaw
     ***   of each method if they can run properly, or once fixed, to show that
     ***   the methods are now correct
     *****************************************************************************/
     public static void main(String[] args) {
         /***** Testing Average ***/
         int[] A = {1,2,3,4,5,6,7,8,9,10};
         System.out.println(average(A));

         /***** Testing Find ***/
         double[] D = {0.1, 0.7, 9.7, 4.5, 5.3};
         System.out.println(find(D,0.1));

         /***** Testing Longest ***/
         String[] S = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday"};
         System.out.println(longest(S));

         /***** Testing everyOther ***/
         String G = "abcdefghijklmnopqrstuvwxyz";
         System.out.println(everyOther(G));
     }
 }
```

CS1101 Introduction to Computer Science – Lab

Lab Objectives: Students will learn the foundations of algorithmic thinking and algorithm development, and learn how to implement them in a variety of languages. They will also learn to be active learners. They will develop problem-solving skills and build team skills, critical-thinking skills, and professionalism.
Prerequisite: MATH 1508 or MATH 1411 with a grade of C or better.

Knowledge and Abilities Required Before Entering the Course: Students entering the course are not required to have a background in Computer Science or programming. They should be familiar with topics from Pre-calculus, including algebraic functions, proofs, and base representations of numbers.

Logistics:

Lab sessions: MWF from 8 a.m. to 8:50 a.m. in CCSB 1.0704
Instructor: Dr. Martine Ceberio – mceberio@utep.edu – office room: CCSB 3.0406
Office hours: MW from 1:30 p.m. to 2:50 p.m.
+ **by appointment & open-door policy**

Textbook: *Programming in Java*, by Zybooks, available at zybooks.zyante.com. It is the same as for your lecture CS1301, so if you are registered in CS1301, do not purchase a book again. If you are only registered in the lab, you should subscribe to your textbook by entering the following code:

UTEPCS1301-02CeberioFall2018

Communication platform: This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.

Find our class page at: piazza.com/utep/fall2018/cs1101

Software: Software used in this course is available on the desktop computers in the main computer lab and in the two instructional labs on the first floor. To use the course software on your home or laptop computer, instructions will be given in the labs and available online on our piazza page.

Important rule about using your personal laptop computers:

It is your choice to use your personal computer or UTEP's desktop to complete the labs assigned to you. However it is essential that you be able to show your work anytime we ask you for it in lab. For instance we will not accept that "your work is on your laptop – or somewhere else – and you cannot produce it at the time we request it". To avoid such situation you could for instance use Dropbox (dropbox.com on which you get extra free space based on your utep.edu address) and hence make sure that you can access your work from anywhere. Any option you pick, you need to be able to produce your work at any time in lab for our review and grading. **There will be no exception to this rule.**

Note: You should be enrolled in one lab section. Do not drop in on a lab or lecture section other than yours without prior approval from your instructor.

Grading

Grades are communicated to students in a timely manner. It is the students' responsibility to keep track of their grades by compiling the grades they receive. Your semester grade will be based on a combination of lab assignments, homework assignments, pop quizzes and in-lab assignments, and lab participation.

The approximate percentages are as follows:

- 58% Lab assignments (about 10 per semester)
- 12% Homework: Challenge activities from the online textbook zybook
- 25% Pop quizzes and in-lab assignments
- 5% Lab participation (includes on-time attendance, participation in labs, any quizzes for attendance and survey purposes)

The nominal percentage-score-to-letter-grade conversion for CS 1301 is as follows:

- 90% or higher is an A
- 80-89% is a B
- 70-79% is a C
- 60-69% is a D
- Below 60% is an F

Note: You must earn a C or better in each of CS1301 and CS1101 to continue to the next course in this sequence, which is CS2401. In order to pass CS1101, you need to:

- Earn a C or better overall
- **AND** have submitted all 3 comprehensive labs and obtain at least a C average on them
- **AND** out of the last 5 labs, submit at least 3 and obtain at least a C in each

Expectations

Lab assignments are designed to allow you to practice the topics that constitute the outcomes of this course. Lab assignments will be either:

- Regular lab assignments meant to provide practice on a couple of very specific topics covered at that time of the semester (11 to 13 total); or
- Comprehensive lab assignments (3 total) meant to check the acquisition of a broader set of skills, already addressed earlier in the semester in regular programming assignments.

Regular and **comprehensive lab assignments** will not weigh the same. Comprehensive labs will weigh more and will usually require more time to complete. Also, please note that, to pass this class, students need to obtain a C average on the 3 comprehensive labs.

All lab assignments will include a part that has to be done without a computer: the description of the algorithms you designed to address the problems at hand. Such algorithms are not written in code as it is really important that students understand, early on, that computer science is about designing ways to solve problems and that these approaches (algorithms) most usually do not depend on any specific language.

Deadlines for lab assignments will be clearly specified in the description of each assignment. Assignments turned in up to three days late will have scores reduced by 15% for each day of lateness.

When assessing labs, TAs will spend 5 to 10 minutes with each students asking probing questions about the topics covered in the assignments: these questions will be asked regardless of whether you completed the assignment or not. This allows you flexibility, in case something happened and you were not able to complete an assignment, to make up for some points.

Homework: Homework will be assigned weekly from the online textbook of the course. They will be the challenge activities of this book (as participation activities will be assigned as homework for CS1301) and assignments from other web-based sources. Completing homework on time is essential to staying on track with the work done in lab. Homework will be assigned with plenty of time for students to complete it. Lateness to complete the assigned homework will be penalized by 5 points per day of lateness.

Quizzes and in-class assignments: The purpose of each quiz and in-class assignments is to ensure that you are staying current with the weekly assignments and to verify that you have acquired the skills developed in lab so far. Quizzes will usually be on-line quizzes on socrative.com. Individual checks on the homework where the TA asks a student to explain his or her work will also count towards the quizzes and in-class assignments grade. There will be no make-up on missed quizzes, in-class assignments, or homework checks, so attendance is crucial.

Lab Participation: Attendance at and participation in all lab sessions are critical factors of your success in this lab course. Students should be on time for all scheduled sessions and attend the entire session. Attendance will be taken at every session and will count towards your class participation grade.

Students should **notify the instructor prior to missing a session** if at all possible, and certainly right after if earlier was not possible. The instructor will allow two unexcused absences per semester before having the option to deduct points from the final grade (5 points per subsequent unexcused absence). Note that excessive absence may result in being dropped from the lab.

It is the **student's responsibility to obtain the content covered during missed labs**. Participation points also include completing post-labs online quizzes (when requested, if any) that are administered as surveys to monitor students' overall progress and potential struggles.

Resources

Special Accommodations: If you have a disability and need classroom accommodations, please contact the Center for Accommodations and Support Services (CASS) at 747-5148 or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at www.sa.utep.edu/cass. CASS' staff are the only individuals who can validate and if need be, authorize accommodations for students with disabilities.

Scholastic Dishonesty: Any student who commits an act of scholastic dishonesty is subject to discipline. Scholastic dishonesty includes, but not limited to cheating, plagiarism, collusion, and submission for credit of any work or materials that are attributable to another person.

Cheating is:
- Copying from the test paper of another student
- Communicating with another student during a test to be taken individually
- Giving or seeking aid from another student during a test to be taken individually
- Possession and/or use of unauthorized materials during tests (i.e. crib notes, class notes, books, etc.)
- Substituting for another person to take a test
- Falsifying research data, reports, academic work offered for credit

Plagiarism is:
- Using someone's work in your assignments without the proper citations
- Submitting the same paper or assignment from a different course, without direct permission of instructors

To avoid plagiarism, see: http://sa.utep.edu/osccr/wp-content/uploads/sites/8/2012/09/Avoiding-Plagiarism.pdf

Collusion is: Unauthorized collaboration with another person in preparing academic assignments

Important! When in doubt on any of the above, please contact your instructor to check if you are following authorized procedure.

Detailed Learning Outcomes

Level 1: Knowledge and Comprehension. Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. On successful completion of this course, students will be able to describe, at a high level:

1. Computer representation of simple data types and operations, including operations with binary numbers
2. Technical aspects of computing, including memory, operating systems, editors, interpreters, compilers, debuggers, and virtual machine
3. Differences among programming languages
4. The purpose and use of exceptions
5. Pseudocode and implementation in a programming language of the use of Multi-D arrays
6. Pseudocode and implementation in a programming language of the use of Linked lists

Level 2: Application and Analysis. Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details. Upon successful completion of this course, students will be able:

1.To analyze problems and express solution algorithms in pseudocode
2. To implement pseudocode algorithms in a high-level language, including the correct use of:
    a. Arithmetic and logical expressions
    b. Simple I/O operations
    c. User-defined subprograms, including recursive methods
    d. User-defined types
3. To use testing and debugging strategies, including black-box and white-box testing, test drivers, stubs and test suites, to identify software faults
4. Development of teamwork skills, including the use of teamwork roles.

Level 3 Outcomes: Synthesis and Evaluation. Level 3 outcomes are those in which the student can apply the material in new situations. This is the highest level of mastery. On successful completion of this course, students will be able to use the syntax and semantics of a higher-level language to express solutions to programming problems, including the correct use of:

1. Basic variable types such as integer, real number, character, string, 1-D array
2. Assignment, arithmetic, and logical operations
3. Basic control structures: if-then, for-loop, while-loop

CS1101: Weekly Group Work – Khôlles

Below is an example of a Khôlle assignment along with the report template that the students have to fill and the self-assessment document that the students use when they complete the report in which they have to assess their work.

Welcome back to your Friday Khôlle!

If you are reading this document, it means that you are a helper today! Great! Let's get started!

Today's Problem

Given a time in a 12-hour AM/PM format, convert it to military (24-hour) time.

Note: Midnight is 12:00:00AM on a 12-hour clock, and 00:00:00 on a 24-hour clock. Noon is 12:00:00PM on a 12-hour clock, and 12:00:00 on a 24-hour clock.

Function Description

Complete the timeConversion function in the editor below. It should return a new string representing the input time in 24-hour format. Method timeConversion has the following parameter:

s: a string representing time in 12-hour format.

Input Format

A single string *s* containing a time in 12-hour clock format (i.e.: hh:mm:ssAM or hh:mm:ssPM), where 01 ≤ hh ≤ 12 and 00 ≤ mm, ss ≤ 59.

Constraints: All input times are valid

Output Format: Convert and print the given time in 24-hour format, where 00 ≤ hh ≤ 23.

Sample Input: 07:05:45PM

Sample Output: 19:05:45

You are expected to write the body of method timeConversion:

```
static String timeConversion(String s) {
   // your code goes in this section
   //(write it on the board for your whole team to see)
}
```

From: HackerRank.com.

Do not forget to:

- Answer any question the problem solvers ask you
- Monitor the time of the exercise: the goal is to ensure that the whole team gets to participate in their own role
- Intervene when prompted for help by the problem solvers: helpers intervene, with a hint or a solution, if they deem that the problem solvers need help (when stuck or going in the wrong direction or time is running out)

Possible Hint: *If the problem solvers are stuck (or somehow stuck), you may want to point their attention onto the fact that:*
1/ their attention should primarily be on the last 2 characters: to identify whether they have to change an AM or a PM time.
2/ once they have identified whether it is AM or PM, then, the only change that could happen concerns the first 2 characters of the string (the hour digits).

MORE HELP

The problem solvers may ask what the conversion does when it is 12:XX:XXAM. You should return the question to them, and then if they cannot figure it out, give them the answer of 00:XX:XX.
Similarly, they may ask what the conversion does when it is 12:XX:XXPM. You should return the question to them, and then if they cannot figure it out, give them the answer of 12:XX:XX.
Note: if they never ask but are stuck or going in the wrong direction, intervene and give them these examples.

SOLUTION

Below, we are showing you a possible solution for your eyes only. Do not share directly with students of your team who are not helpers today. However, having access to the solution further helps you identify when the problem solvers may not be on the right track.

```
static String timeConversion(String s) {
      int hour;
      if (s.substring(s.length()-2).compareToIgnoreCase("AM")==0) {
            hour = Integer.valueOf(s.substring(0,2));
            if (hour == 12)
                  return "00" + s.substring(2,s.length()-2);
            else
                  return s.substring(0,s.length()-2);
      } else {
            hour = Integer.valueOf(s.substring(0,2));
            if (hour == 12)
                  return "" + hour + s.substring(2,s.length()-2);
            else {
                  hour += 12;
                  return "" + hour + s.substring(2,s.length()-2);
            }
      }
}
```

Input: 12:05:58AM
Output: 00:05:58

Possible test cases (to be provided by the testers) are:

- Regular input: 03:54:23AM → 03:54:23
- Regular input: 03:54:23PM → 15:54:23
- Input: 12:05:58AM → Output: 00:05:58
- Input: 12:05:58PM → Output: 12:05:58

Khôlle assignment:

Problem Solvers' Approach	Helpers' hints or more
Here transcribe the problem solvers' approach and their reasoning… It should be a mix of pseudocode (or code) and explanations. Indicate at which point (if any) the problem solvers ask for help. *Take as much space as needed.*	*Here, take note of whenever the helpers provide hint, direct the problem solvers to another type of solution, or even give a solution and report what help was provided.*

Reviewers' comments
Enter as many comments as relevant / necessary on code correctness and coding style. Be as specific as possible. Use as much space as you need.

Tests provided by testers	Justification for tests
… use as many rows as needed...	

Team Self Assessment: (circle what applies)

Problem Solvers:	NI	G	V	E
Helpers:	NI	G	V	E
Reviewers/Note-Takers:	NI	G	V	E
Testers:	NI	G	V	E

Approved by: ***(below, each member should put their names to indicate approval of the above report)***

…… ……
…… ……
…… ……
…… ……

Report template that students fill after completing each weekly Khôlle:

Our Weekly Khôlle Report

(name this file: YYYYMMDD-groupX where X is your group number or your group nickname)

Who? (below: list all present members of your original team)	
Problem Solver 1	
Problem Solver 2	
Helper 1	
Helper 2	
Reviewer / Note Taker 1	
Reviewer / Note Taker 2	
Tester 1	
Tester 2	

Who else? (any guest member?)	
Name...	Today's role...
Name...	Today's role...
Name...	Today's role...

Problem of the Day:
Write the problem in here…

Questions by problem solvers	**Answers by helpers**
… add more as you need	

Khôlles Assessment and Self Assessment Guide

Each student will be assessed as follows:
Scale: Needs Improvement / Good / Very Good / Excellent

For each problem addressed by a team, a report will be filled. This report (see report template) will account for each role contribution. Below is a description of what is expected of each role. Note that students will be assessed in their pairs, not individually.

- **Problem solvers:**
    a. Should ask questions to understand, clarify, probe special cases of the problem posed to them
    b. Should think out loud (and write their thinking down. At least it should be recorded by the note takers)
    c. Should be open to suggestions by the helpers
    d. Should be discussing alternatives approaches
    e. Should be able to discuss the performance of their approach

  NI: 0-1 skill / G: 2 skills / VG: 3-4 skills / E: 5 skills
- **Helpers:** should understand the problem, as demonstrated by their ability to:
    a. Answer questions
    b. Guide the problem solvers when needed or asked
    c. Should ask further questions or propose alternative approaches

  NI: 0-1 skill / G: 2 skills / VG: 3 skills / E: 3 skills fluently, effectively
- **Reviewers/note takers:**
    a. Should transcribe all questions asked by the problem solvers and the helpers' answers (see report template)
    b. Should transcribe the problem solvers and helpers interaction, approaches, and hints (see report template)
    c. Should comment on the correctness of the code/approach -- and these comments should be transcribed on the report
    d. Should comment on the quality of the code (or understandability, readability of the answer if no code is provided or relevant) -- and these comments should be transcribed on the report

  NI: 0-1 skill / G: 2-3 skills / VG: 4 skills / E: 4 skills fluently, effectively
- **Testers:**
    a. Should provide at least one test case for the problem that was just solved
    b. Should justify why each test case was chosen
    c. Should trace the given approach for their test case(s)
    d. Should provide at least one edge case and be able to justify it

  NI: 0-2 skill with one test case only / G: 2 skills with at least 2 test cases or 3 skills with at least 1 test case / VG: 3 skills with at least 2 test cases / E: 4 skills

Finally, each team is expected to complete the following:

- All members agree to the report (it is not allowed for any member to sign for a member who is not present).
- The team submits the report on time.

These tasks will be assessed as a team as follows:
Failed to submit -- FS (report not submitted) / Needs team coordination -- NTC (submitted but not approved by all or not submitted on time) / Satisfactory -- S (completed both tasks)

IMPORTANT NOTE:

It is expected that students will develop into their individual roles as the semester progresses. The objective is for each to reach a Very Good to Excellent level by the end of the semester, as the result of a generally upward trend throughout the semester. However, it is understood that some weeks will not be as good as others: it is expected and should not affect the overall performance if the general trend is still upward. The key is that the more you practice, the more chances you have to reach this goal.

CS4365: Problem Solving and Algorithms (cross-listed with CS5354)

In Spring 2015, I started teaching CS1 (CS1301/1101) after a long break (5 years) without teaching it. At that time, I re-discovered the nature of our entering student body. I found students with low confidence, and yet plenty of skills I could recognize. They were, in particular, very able to solve problems: they did every day in their lives, juggling so many constraints! However, when it came to the classwork, even the simplest problems left them helpless. Convinced that what I was observing was not a lack of ability, but rather anxiety and a lack of confidence, I decided to start a weekly problem-solving club, originally intended for my CS1301/1101 students (CS1401 at that time), and later open to all CS1,2,3 students (Freshmen and Sophomore students in CS). The goals of this club were 1/ to up my students' confidence in their ability to solve problems, 2/ to re-ignite their interest in solving problems through fun activities, and 3/ to equip my students with problem-solving "tools" they could keep using in their classes (mine included). This worked well, so I decided to try and teach a similar content, but as a course: for summer 2016, I put together a 3-credit-hour course that I offered cross-listed to undergraduate and graduate students together.

This course was a mix of seemingly random problems to solve, to have my students rekindle with the idea that solving problems is fun and that they can do it, and more structured activities involving programming and discussing different problem-solving strategies.

CS 4365 / 5354 Topics in Soft Computing: Problem Solving & Algorithms

Summer 2016 Syllabus

Sections: Lecture sessions MTWRF 7:00-9:10 a.m., in CCSB 1.0204, from June 6 to June 30. *Instructor Contact Information:* Martine Ceberio, mceberio@utep.edu, CCSB 3.0406

Course Objectives: This course is intended to enhance students' problem-solving abilities. Through problem solving, they will learn an array of general strategies for algorithm design, they will practice performance analysis and develop critical thinking skills. They will review and apply in context notions of data structures, algorithms, discrete math, and logical foundations of computer science.

In this class, you are going to develop / enhance your problem-solving skills and increase your knowledge of fundamentals of computer science. However, doing so might not feel very comfortable. For instance, you will be challenged in class to identify solutions to problems. You will be requested to convey your solutions to the whole class (in a manner that is clearly understandable). While doing this, you will review fundamental topics of computer science: algorithmic strategies and algorithm analysis for instance.

You are more than ever required to come to class prepared (having completed your reading assignment) to ensure that the class activities can be as rich as possible.

Textbook:
For students taking **CS4365:** Problem Solving Through Recreational Mathematics, by Averbach & Chein. Dover.
For students taking **CS5354:** Problem Solving Through Recreational Mathematics, by Averbach & Chein. Dover, and The Power of Algorithms: Inspiration and Examples in Everyday Life, by Giorgio Ausiello and Rossella Petreschi, Springer.

Grading: Grades are turned in to students in a timely manner. It is the students' responsibility to keep track of their grades by compiling the grades they receive. Your semester grade will be based on a combination of homework assignments and quizzes (unannounced), class participation, mid-term exam (1), and a final exam.

The approximate percentages for students taking CS4365 are as follows:	The approximate percentages for students taking CS5354 are as follows:	The nominal percentage-score-to-letter-grade conversion is as follows:
• 30% Class participation (includes quizzes for attendance and survey purposes) • 30% Quizzes & homework • 25% Mid-term exam • 15% Final exam	• 20% Class participation (includes quizzes for attendance and survey purposes) • 25% Quizzes & homework • 25% Project • 15% Mid-term exam • 15% Final exam	• 90% or higher is an A • 80-89% is a B • 70-79% is a C • 60-69% is a D • below 60% is an F

Attendance and active class participation: Attendance at and participation in all class sessions are critical components of this course, hence the high percentage for attendance and participation in your overall grade. Participation includes contributing to the class sessions, showing that reading has been completed (even if not fully understood – the purpose of the class is to clear doubts and to go deeper), presenting solutions to problems, discussing and contrasting different approaches.

Students should attempt to be on time for all scheduled sessions and attend the entire session. Students should notify the instructor prior to missing a session if at all possible, and certainly right after if earlier was not possible. The instructor will allow two unexcused absences per semester before having the option to deduct points from the final grade (5 points per subsequent unexcused absence). Any assignments due on the date of the unexcused absence will be considered late if not turned in as specified by the assignment guidelines, unless an exception has been previously granted by the instructor. Points lost due to an unexcused absence may not be made up. It is the student's responsibility to obtain the content covered during missed class(es). Regularly during the semester, you will also be expected to take online quizzes, whose aim is to better understand where you are at to best help you. Taking these quizzes is critical to your success and therefore will be taken into account for your grade.

Assignments: Reading and homework assignments will be handed out or announced in class, and/or posted on the class Website, which is hosted on piazza.com. Most homework will be "on the paper" but some will be programming assignments because it is important to understand how computers can affect / inflect our problem-solving approaches. Homework assignments are due by the beginning of the class on the due date, unless specified otherwise. If you miss a class, it is your responsibility to find out what you missed. You should expect to spend at least twenty hours per week outside of class on reading and homework.

Quizzes: The purpose of each quiz is to ensure that you are staying current with the weekly reading assignments and to verify that you have mastered the skills developed in class. Quizzes usually will be on-line take-home quizzes on socrative.com. There will be no make-up for missed quizzes.

Projects: Students taking CS5354 will have to pick a project and present it during the semester. Project options as well as presentation dates will be specified during the first week of classes.

Exams: The purpose of exams is to allow you to demonstrate mastery of course concepts. Because of the nature of this course, exams will be take-home exams to allow you more time to think and answer questions. There will be one mid-term exam and one final exam.

Standards of Conduct: You are expected to conduct yourself in a professional and courteous manner, as prescribed by the UTEP Standards of Conduct.

Graded work (for example, homework or exams), is to be completed independently and should be unmistakably your own work (or, in the case of pair work, your pair's work). You may not represent as your own work material that is transcribed or copied from another person, book, or any other source, such as a web page. Professors are required to—and will—report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.

Use of Unauthorized Electronic Devices during Class Sessions: Any use of unauthorized electronic devices that disrupts the learning environment (e.g., surfing the Web, listening to music, checking Facebook, Twittering, playing Angry Birds Rio, or playing online Scrabble while class is in session) will not be tolerated. Electronic devices should serve as tools for learning and are limited to course-related work only; any other use is considered inappropriate. Inappropriate use of electronic devices will be considered a disruption of the classroom and may be reported to the Dean's office. All unauthorized electronic devices should be silenced or shut off upon entering the classroom. In the event of an emergency or other urgent situation, the student should step outside of the classroom beyond hearing range or text silently. It is the student's responsibility to ensure that all electronic devices are managed within the guidelines. The instructor reserves the right to disallow use of any electronic equipment during class sessions.

Disabilities: If you feel that you may have a disability that requires accommodation, contact the Center for Accommodations and Support Services at 747-5184, go to Room 106 E. Union, or email cass@utep.edu.

Assessments in this course were in the form of:

- Quizzes: to show students at what level they are expected to perform, on almost a daily basis;
- Homework: because a lot of the work students learned in this class required them to spend time outside the classroom, so often I sent them home with work to complete. I wanted them to show me their best work, not the work they can do in limited time.
- Classwork: we solved problems together so I could model the approaches I taught them. Class activities also included discussions of multiple approaches to solving a given problem: is there a better one? If so, how so? If not, why?
- Exams: There were two exams. Both were take-home exams for the same reason as the students were given a lot of homework. I wanted students to have time to think about their approaches. Since the work I expected from them was more at the modeling and argumentation levels, I was not worried about risks of cheating: each student was expected to be able to articulate his/her approach to solving given problems. Most of the exams were not replicable. Below is the final exam given in this class.

CS4365 Final Exam

CS4365 – 5354 Problem Solving and Algorithms
Final Examination
June 30, 2016 – Take-home final exam

Available on:
Piazza.com/utep/summer2016/cs43655354/home on June 30 at 7 a.m.
Due on:
July 2nd by 11:59 p.m. as a private note on piazza attaching the docx **answer sheet** as specified in the exam's description.

Exam's Rules:
You are not allowed to look up solutions of the below questions online.
You are allowed to brainstorm on solutions with a group of your classmates. However, you should be able to fully understand the solution and clearly articulate it in your answer sheet.
If there is any doubt that the solution was yours, I reserve the right to ask you to complete your exam with an oral examination.

Total number of points:	**/ 110 + 10 points extra credit**
Question 1.	**/10**
Question 2.	**/10**
Question 3.	**/10**
Question 4.	**/10**
Question 5.	**/10**
Question 6.	**/10**
Question 7.	**/10**
Question 8.	**/10**
Question 9.	**/10**
Question 10.	**/10**
Question 11.	**/10**
Question 12.	**/10**

Question 1. Warm-up problem.
Solve the following problem. Explain the reasoning that supports your answer.

During his sabbatical, Professor Flugel visited that favorite puzzleland country in which there are only two types of inhabitants: those who always speak the truth and those who always lie. They also answer "Yes" or "No" to any question for which such an answer is meaningful. During his visit, the professor amused himself by asking a question which none of the inhabitants could answer, although the question could be answered "Yes" or "No" and did not require any factual knowledge of which the inhabitants were unaware. What was the question?

Question 2. A weight problem.
Propose a solution to the problem. Explain how you reached this solution. What was your strategy?

A chemist has a set of five weights. She knows that it includes one 1-gram weight, and also one each 2-, 3-, 4-, and 5-gram weights, but because they are unmarked, she has no way of telling them apart except by placing them on a balance. She may place any combination of weights on each of the two pans and determine if one side is heavier than the other or if they balance.
Show how in five weighings she can identify each of the weights.

Question 3. Rabbits.
Explain how you approach this problem. Describe your reasoning and show your solution in details.

A man puts a pair of rabbits in a place surrounded on all sides by a wall. The initial pair of rabbits (male and female) are newborn. All rabbit pairs are not fertile during their first month of life but give birth to one new male/female pair at the end of the second month and every month thereafter. How many pairs of rabbits will there be in a year?

Question 4. Survival problem.
Explain how you approach this problem. Describe your reasoning and show your solution.

There are 12 very smart prisoners in a jail. To get rid of them, the warden comes up with the following test. He will put a hat, either black or white, on the head of each of these prisoners. There will be at least one hat of each color, and the prisoners will be informed about this fact. They will be able to see everyone else's hat but their own; there will be no communications of any kind among the prisoners. The warden will line up the prisoners every 5 minutes starting at 12:05 pm and ending at 12:55 pm. To pass the test, all the prisoners with a black hat and only those prisoners will have to step forward during the same line up.
If they do, all the prisoners will have to step forward during the same line up. If they do, all prisoners will be freed, otherwise they will be executed.
How can the prisoners pass the test?

Question 5. Logic 1.
For each of the following sentences, write an English sentence that conveys its negation.

(1) No lecture was attended by every student.
(2) Every football team has a quarterback.
(3) No animal is both a cat and a dog.

Question 6. Logic 2.
Is the following argument valid? 1/ Translate the following reasoning in formal logic formula, and 2/ show (in details) whether the reasoning is valid.

John, a student in this class, is 16 years old. Everyone who is 16 years old can get a driver's license. Therefore, someone in this class can get a driver's license.

Question 7. Logic 3.
Is the following argument valid? 1/ Translate the following reasoning in formal logic formula, and 2/ show (in details) whether the reasoning is valid.

If it is right for me to lie and not right for you to lie, then there is a relevant difference between our cases. There is no relevant difference between our cases. It is not right for you to lie. Therefore it is not right for me to lie.

Problem 8. Posing problems properly 1.
Read the following problem and pose it to be solved (with unknowns, domains, and knowns): explain why you pose it the way you do. Then solve it.

At 8 a.m., a train leaves Topeka for Santa Fe and another train leaves Santa Fe for Topeka. The trains maintain constant speeds with no stops. The first train requires five hours to complete the trip and the second train requires seven hours.
At what time do the trains pass each other?

Problem 9. Posing problems properly 2.
Read the following problem and pose it to be solved (with unknowns, domains, and knowns): explain why you pose it the way you do. Then solve it.

One fourth of a heard of camels was seen in the forest. Twice the square root of that herd had gone to the mountain slopes. Three times five camels remained on the riverbank. How many camels were there in the herd?

Problem 10. Posing problems properly 3.
Read the following problem and pose it to be solved (with unknowns, domains, and knowns): explain why you pose it the way you do. Then solve it.

If I were to give 7 cents to each of the beggars at my door, I would have 24 cents left. I lack 32 cents of being able to give them 9 cents apiece. How many beggars are there? And how much money do I have?

Problem 11. Reading well and posing the problem.
Read the following problem and pose it: explain the strategy you use to solve it and show how you do.

Adam, Robert, Clifton, Stephen, and Brent are the five starters on the Doylestown Dribblers basketball team. Two are left-handed and three right-handed. Two are over 6 feet tall and three are under 6 feet. Adam and Clifton are of the same handedness, whereas Stephen and Brent use different hands. Robert and Brent are of the same height range, while Clifton and Stephen are in different height ranges.
The man who plays center is over 6 feet and is left handed.
Who is he?

Problem 12. Last problem ☺
Read the following problem. Clearly present the model you use to solve it and the approach to solve the given model. Then solve it and describe your solution throroughly.

Show that in a room full of people (more than one person), there are at least two people who have the same number of friends in the room. (Assume that if B is A's friend, then A is also B's friend).

This course received good evaluations and feedback and became the starting point of the creation of a series of three problem-solving courses, of which I designed the first, as described below.

CS1190: Problem Solving

This course was first taught in fall 2017. Its creation is the result of the following:

- The CS4365/5354 summer Problem-Solving and Algorithms course I taught in summer 2016;
- The NSF RED grant our department received in July 2016, which led us to revisit our pedagogy, course offerings, curriculum, professional development needs, etc.; and most importantly,
- A departmental retreat (end of summer 2016) where we discussed the idea of curriculum revamping with the creation of 1-credit-hour courses to better serve our students: As our CS department was embarking in our NSF RED project, our faculty brainstormed about how to provide an even better experience for our students. Among many potential aspects of this problem, one that I am familiar with as a CS1 instructor is attrition in early semesters and lack of confidence in problem-solving skills. Attrition in early semesters is often linked with our students' lack of sense of purpose, of understanding of CS' "big picture'". This is reinforced by the fact that in their first three semesters in CS, our students typically only have one computer science course available to take each semester (namely: CS1, CS2, and then CS3). As a department, we proposed to offer 1-credit-hour courses available for our CS students to take in parallel of their "thin" course sequence CS1-CS2-CS3. One of the courses we proposed to offer is a 1CH course on Problem-Solving.

As a result, in spring 2017, our department, led by Dr. Ann Gates, started working with Google Engineers, Lorne Needle and Mike Gainer, on the creation of problem-solving courses, in collaboration with CAHSI institutions, New Mexico State University and the University of California at Dominguez Hills. In this collaboration, we came up with the idea of three 1-credit-hour courses on problem solving, from introductory to advanced level. I was put in charge of creating the introductory course, which I then taught as a topics in computing CS1190 over 6 weeks (2.5 hours per week) in early fall 2017 and again in late fall 2017. In late summer 2017, while finishing the preparation of the pilot of this CS1190 course, Dr. Johannes Strobel (from University of Missouri) joined the team

My problem-solving course focuses on developing the students' ability to attack a problem, any problem, even outside of the students' area of expertise. I provide a problem-solving framework so that they have a structured approach and are aware of important steps in problem solving. A lot of what I do in this course consists in helping students realize how skilled they are and helping them reflect on their problem-solving approaches so that we can better understand them and refine them. When I taught it as a 6-week course in early fall 2017, with 12 students, students were exposed to a variety of problems, from short riddles to larger problems, some proposed by Google engineer Mike Gainer, and one by Craig Tweedie, an environmental scientist, professor at UTEP. The culmination of this course was a project presentation to Google engineer Lorne Needle. I taught it again at the end of the fall 2017 semester following a similar format. Feedback was very positive. I wrote a blog piece about my experience with this course. It is available at: http://martineceberio.fr/blog/problem-solving-computer-scientists.

Since then, I was involved in packaging my course material so that it can be disseminated to other institutions across the country. As of this semester Fall 2018, 13 institutions are teaching one of the 3 problem solving courses developed as collaboration between our NSF RED project, CAHSI, and Google.

CS1190 Syllabus:

CS1190: Special Topics in Computing
Problem Solving and Algorithms

Course Objectives. In this course, students will learn problem-solving approaches and hone their problem-solving skills on a variety of problems in a wide range of domains and articulating the risks and benefits of various solutions.

Logistics. This course will meet in room CCSB 1.0510 on:

- Wednesdays from 9 a.m. to 9:50 a.m.
- Fridays from 12 p.m. to 1:20 p.m.

From November 1, 2017 until December 8, 2017.
There will be no textbook.

Communication platform. This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com. Find our class page at: https://piazza.com/utep/other/cs1190/home.

Grading. Grades are turned in to students in a timely manner. It is the students' responsibility to keep track of their grades by compiling the grades they receive. Your semester grade will be based on a combination of attendance with active class participation, presentations, homework assignments, one midterm exam, and one final exam. The approximate percentages are as follows:

- 10% Attendance and Active participation in class
- 35% Homework assignments, including in-class Presentations of Homework
- 20% Final Project Presentation
- 15% Exam 1
- 20% Exam 2

The nominal percentage-score-to-letter-grade conversion for CS 1190 is as follows:

- 90% or higher is an A
- 80-89% is a B
- 70-79% is a C
- 60-69% is a D
- Below 60% is an F

Expectations:

Class Participation: Attendance at and participation in all lecture sessions are critical factors of your success in this course. Students should be **on time** for all scheduled sessions and attend the entire session. Attendance will be taken at every session and will count towards your class attendance grade. Students should **notify the instructor prior to missing a session** if at all possible, and certainly right after if earlier was not possible. The instructor will allow two unexcused absences per semester before having the option to deduct points from the final grade (5 points per subsequent unexcused absence).

Detailed Learning Outcomes:

Level 1: Knowledge and Comprehension. Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. The material has been presented only at a superficial level. Upon successful completion of this course, students will be able to:

1.1 Describe two problem-solving approaches.
1.2 Describe the difference between clarifying and probing questions.

Level 2: Application and Analysis. Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details. Upon successful completion of this course, students will be able to:

2.1 Apply the IDEAL and 7-step problem-solving approaches to familiar problems.
2.2 Evaluable information or situations.
2.3 Break down a problem into its key components.
2.4 Assess the benefits and risks of given solutions.
2.5 Contribute to brainstorming activities in which needed resources are identified to solve a given problem.
2.6 Ask clarifying and probing questions to improve understanding of a problem.
2.7 Rephrase a problem description to demonstrate understanding.
2.8 Reflect on one's own process to identify possible improvements.

Level 3: Synthesis and Evaluation. Level 3 outcomes are those in which the student can apply the material in new situations. This is the highest level of mastery. Upon successful completion of this course, students will be able to:

3.1 Refine a problem description by asking relevant clarifying and probing questions.
3.2 Identify resources (e.g., data and expertise) that are necessary to attack the problem.
3.3 Examine different perspectives to solving a problem.
3.4 Articulate and defend the solution to a problem over other options.

In summary, students who complete the course will have the follows capabilities:

[Applying] Articulate an approach for problem solving.
[Asking] Ask both clarifying and meaningful probing questions.
[Identifying] Identify the resources needed to solve given problems.
[Presenting/Modeling] Succinctly and unambiguously define the problem to be solved (e.g., by using modeling techniques, problem decomposition techniques).
[Expanding] Apply techniques or tools for extending considerations related to the problem (e.g., domains should not be assumed). They will be able to challenge assumptions about the problem at hand by examining different perspectives.

It is the student's responsibility to obtain the content covered during missed class(es) and be up to date with the homework given in the missed class(es).

Homework: **Homework assignments** will be announced in class and/or posted on piazza (under the Homework section of Resources). If you miss a lecture session, it is your responsibility to find out what you missed. You should expect to spend at least four hours per week outside of lecture on homework.

Presentations: Students will be asked to present their class work and homework in class on a regular and frequent basis. Each presentation will turn into a grade. The average of these grades will contribute to the student's overall final grade. There will be a final project presentation with Google Engineers.

Exams: There will be 1 midterm exam and 1 final exam. These 2 exams together weigh 35% of your overall final grade for CS1190. If you have test-taking difficulties in general, or if you have difficulties with our tests in particular, please come and let me know as soon as possible and/or request appropriate accommodation from UTEP's Center for Accommodation and Students' Services.

The purpose of the **midterm exam** is to allow you to demonstrate mastery of course concepts covered thus far during the semester. The **final exam** will be comprehensive. Both exams will be given during the regularly scheduled lecture sessions and you will then have time to finish them at home: on November 17 and on December 6, 2017. There will be no make-up exams.

Resources:

Special Accommodations: If you have a disability and need classroom accommodations, please contact the Center for Accommodations and Support Services (CASS) at 747-5148 or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at www.sa.utep.edu/cass. CASS' staff are the only individuals who can validate and if need be, authorize accommodations for students with disabilities.

Scholastic Dishonesty: Any student who commits an act of scholastic dishonesty is subject to discipline. Scholastic dishonesty includes, but not limited to cheating, plagiarism, collusion, submission for credit of any work or materials that are attributable to another person.

Cheating is:
- Copying from the test paper of another student
- Communicating with another student during a test to be taken individually
- Giving or seeking aid from another student during a test to be taken individually
- Possession and/or use of unauthorized materials during tests (i.e. crib notes, class notes, books, etc.)
- Substituting for another person to take a test
- Falsifying research data, reports, academic work offered for credit

Plagiarism is:
- Using someone's work in your assignments without the proper citations
- Submitting the same paper or assignment from a different course, without direct permission of instructors

To avoid plagiarism, see: http://sa.utep.edu/osccr/wp-content/uploads/sites/8/2012/09/Avoiding-Plagiarism.pdf

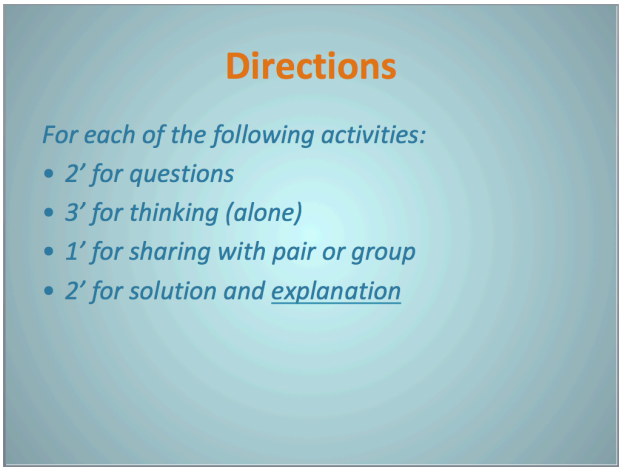
Collusion is: Unauthorized collaboration with another person in preparing academic assignments

Important! When in doubt on any of the above, please contact your instructor to check if you are following authorized procedure.

CS1190 Assessment Material

A central element of this course was the use of a problem-solving framework known as I.D.E.A.L. I provided information to the students about it and asked them to explain it in their own words or creative drawings. Then, students had to solve riddles and larger problems during the semester, until the culminating point of their final project presentations to Google Engineer Lorne Needle. To provide them with enough practice, students were asked to fill out a reporting form for each problem they solved throughout the semester. It allowed me to provide constructive and relevant feedback, better helping students acquire fluency in using the I.D.E.A.L. problem solving framework and in making arguments. Below are the corresponding assessment materials.

Students usually worked in pairs in most of the activities they completed for the course, even for their final project. However, for most of the problems they had to solve, they were asked to first spend time thinking on their own and then to share with their partners. Below is a slide I used at each class session when working on short riddles or problems, to remind my students about the need for them to first think for themselves before to share.



The I.D.E.A.L. Framework for Problem Solving

I: Identify the problem

D: define and represent the problem

E: Explore possible strategies or solutions

A: Act on selected strategy or solution

L: Look back and evaluate

Homework:

1/ How would you rephrase these steps for yourself? Explain what these mean and possibly, if needed, illustrate on one problem.

2/ How would you represent the I.D.E.A.L. framework in a nice graphics? Be as creative as you like, provided your product represents the framework.

Reading Assignment:

Read https://media.lanecc.edu/users/kime/Ch7Cproblemsolving.pdf.

Reporting document

PART I ----------

Rank the following between 1 (poor) and 100 (excellent):

- **Your confidence to be able to solve the same problem in the future:** .....
- **Your ability to solve a similar problem:** .....
- **The usefulness of this exercise in developing your problem-solving skills:** .....
- **How comfortable you felt working on this problem:** .....

PART II ----------

The Problem:

1/ What was the problem to be solved? *(description as given by instructor)*

** **2/ Rephrase the problem:**

3/ What were the important elements in the description of this problem?

4/ Were there non-important elements in the description of this problem, if so what were they?

Approach to this problem:

** **5/ How did you approach this problem?** *Note: Use as much of the vocabulary of the problem-solving framework as possible. You can illustrate with a graphics.*

6/ Were you able to reach a solution?

6.a. 1/ If yes, what is it? Describe it and explain how and why you were able to reach a solution?

2/ What problem-solving skills or strategies did you use to solve this problem?

3/ How would you rate your problem-solving approach on this particular problem and why?

6.b. 1/ If no, why?

2/ What problem-solving skills or strategies did you use to address this problem?

3/ What problem-solving skills or strategies did you lack to solve this problem?

The solution(s):

** **7/ Did this problem have multiple solutions? Explain why or why not.**

8/ Which solution did you aim for? Why? (Discuss its pros and cons)

9/ If this problem had multiple solutions, discuss the pros and cons of at least one other solution.

10/ Note to self for future use and final thoughts:

E.g., What would you do differently next time? What skill(s) did you feel most comfortable / uncomfortable using?

** **Fundamental skills.**

3. Evidence of Continuous Quality Improvement

Student success is a passion! Unfortunately, it is a goal whose contributing factors are always in flux: students change, their situation does too. As educators, we constantly need to redefine what it means to teach well, to teach right. Being true to my goals of recruitment and retention, I need to keep myself up to date with the content and possibly new pedagogy, new technologies, results of studies that inform us how to better serve our students, etc. The changes I bring to my courses, courses' material, pedagogy, and philosophy, are all intended to help me better reach my goal of student success.

Over my 15 years at UTEP, in an effort to always better serve our students, I have consistently and frequently sought professional development. In the last six years only, I have participated in 23 professional development workshops and programs, among which 13 were specifically dedicated or contributed to the improvement of my teaching.

Concrete results of these professional development efforts include the design of new courses, the constant improvement of course pedagogy and material, the development of extra-curricular activities. Below, I go over examples of changes I have brought to my courses and to my philosophy.

Growth Mindset, Asset-Based Approach to Teaching, Competency-Based Assessment

A few years ago, at the 2014 NCWIT Summit, I was fortunate enough to attend the presentation of Dr. Carol Dweck, from Stanford (Psychology). She shared her approach about ***growth mindset*** and presented the compelling results of her studies about the effect of having a growth mindset vs. not. What she shared enlightened my way of teaching. I refined my approach to interacting with students: in the vocabulary I use to provide feedback that encourages growth and celebrates effort, but also I began to explicitly share with my students my new philosophy, being really clear about the fact that their growth matters to me more than whether they come to the start of semester with great computer skills already, hence comforting them about the fact that "they can make it!" This experience was eye opening for me and allowed me to really align my narrative to the students with my true belief that we can achieve excellence with open access.

Additionally, from my experience in CS1, witnessing a broad lack of confidence in my students while being convinced that they are much more capable than they give themselves credit for, I started dedicating a significant amount of time in class conducting what I call "Anchoring". In "Anchoring", I seek to put students in situations in which they perform tasks that are seemingly familiar, and hence, they feel comfortable completing them, only to then show them that what they just did is what they would have thought of to be a complex task if I had used a different vocabulary to present it. As a result, I aim to have students build bridges between familiar tasks and vocabulary and new tasks and vocabulary, hence building on their existing set of skills. Since being invited to become a UTEP EDGE Fellow, I have discovered that this approach to teaching is called ***asset-based teaching***. It is very aligned with my philosophy in which I aim to celebrate students' prior experiences and skills and build on that, rather than identify weaknesses and set to fix them. My attitude towards teaching is indeed not one where I come to class to ``teach'' students content I bring. Instead, I come to class planning to make them realize how much they know already. A large part of my work then consists in building their confidence in skills they have but are not aware of. Once this is done, I can then work with them to take them a few steps further. What that means is that I do not position myself as their ``fixer'', someone who will fix the problems of our ``unprepared'', ``possibly unavailable'' and ``unmotivated'' students: instead, I acknowledge their background and however ``unprepared'' one might think they are, I believe and share with them repeatedly that they already know a lot, that they can do the work, and I demonstrate that to them. I find that, all the more in our institution where students come with a wide range of demonstrated skills, it would be contradictory to our mission of access to accept them in the classroom only to remind them and make them feel unfit, unprepared, while, when looking closely, they are not at all unfit, they just

come with a possibly non-traditional set of skills, but still with skills. This attitude is now a central part of my philosophy (which I share on my blog: http://martineceberio.fr/blog/cs1-philosophy-change-a-kind-approach).

Finally, in the middle of this change of philosophy, I started feeling that the traditional way of assessing with grades that add up to a final grade (all grades included) was at odds with my growth mindset and asset-based approach to teaching. It felt much more honest to adopt a "forgiving" approach in which students should be able to recover from mistakes made early in the semester and literally not have to suffer from the weight of a bad grade in their final average if I am true to my philosophy of growth. Moreover, I felt that if I was true to our institution's mission of access, I should support my students to learn and grow. I started to adopt a ***competency-based assessment*** style. Now, let me make clear that I also believe that students should be able to compute their current grade throughout the semester to have an indication of their performance: it is really important for them, all the more for entering students, whom I have taught primarily over the last 3 years. As a result, I share with them the relative weight of every type of grade they will get during the semester (this is what is in the syllabus). However, I share with them that the grade they can compute will only be a lower bound of their actual final grade. The reason is that I also share with them that I am not interested in grading their speed of learning but rather what they become, whether they will be ready for the next course in sequence or not. In doing this, I set the atmosphere in the class to be one where failing is ok, where taking time to learn is ok, because not understanding something only means that a student does not understand it yet, not that he or she will not understand it. As a result, my way of assessing students is very much based on competencies rather than semester-long performance. I find this way of doing to be much truer to my belief that students come with different backgrounds and learning styles and should not be penalized for starting from farther or taking longer to learn a concept, as long as it is acquired by the end of the semester.

Competency-based assessing frees me and the students in my class from the fear of failing. Every quiz and exam is now an instrument that helps us determine "what's next": students can use them as indications of what is expected of them, what they should study more, what they need to seek help about, etc. with the ultimate goal of learning.

These changes have affected my whole approach to teaching and I have extended them to all courses I have taught in recent past (since 2015). Those changes helped me be more truthful to my values, hence more genuine with my students, and I believe that my genuine sharing of my values and philosophy in class with students allows me everyday to better serve them, retain them, and help them be more successful.

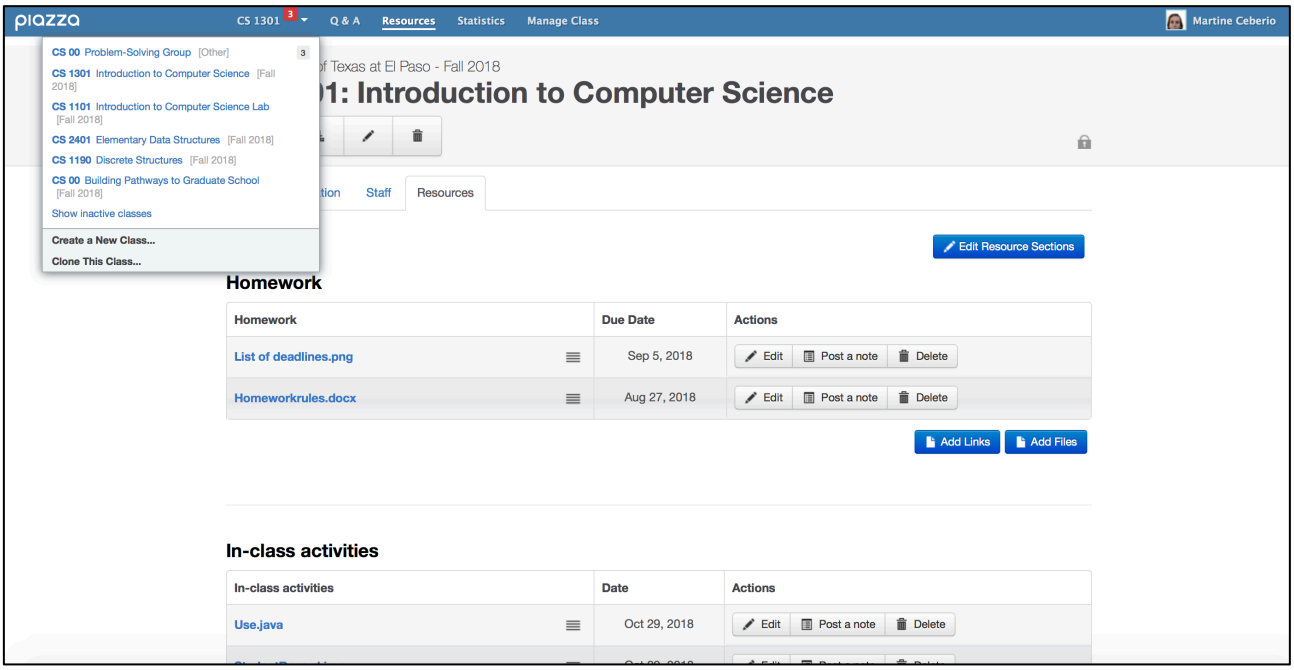
Integrating Online Tools

Understanding the needs of students, their study habits / trends, being able to provide relevant feedback, being able to identify at-risk students and to contact them timely to provide help are all elements that contribute to student success. However, faced with growing enrollment, keeping up with timely assessment and providing timely feedback to students is a challenge. Yet, this is one that needs to be addressed to ensure that we provide support to our students. Over the semesters, I have looked for ways to keep engaging and supporting our students despite the growth in enrollment. In particular, I have included the use of multiple online platforms to provide support to our students but also grading support to my team of teaching assistants and to myself as well. I go over these tools in what follows.

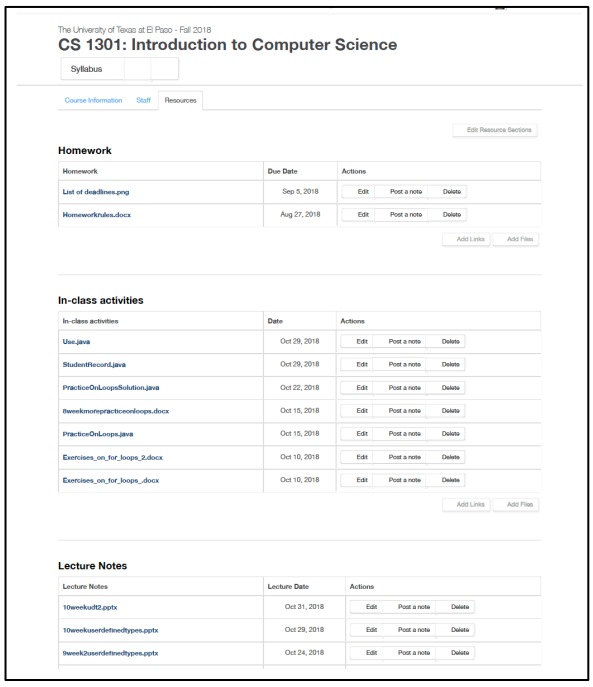
Discussion Forum and Material Sharing: Piazza

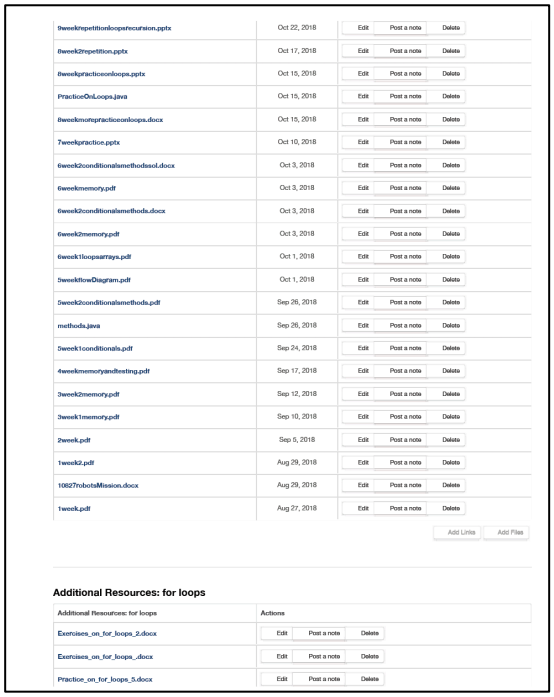
I started by using Piazza as a way to keep communication open with my students and to share all relevant material for each class in one place, but also to encourage students to collaborate with each other, to seek help and to provide help (instructors have the ability to flag any contribution as a good note, a good question, or a good answer: this is encouraging to students).

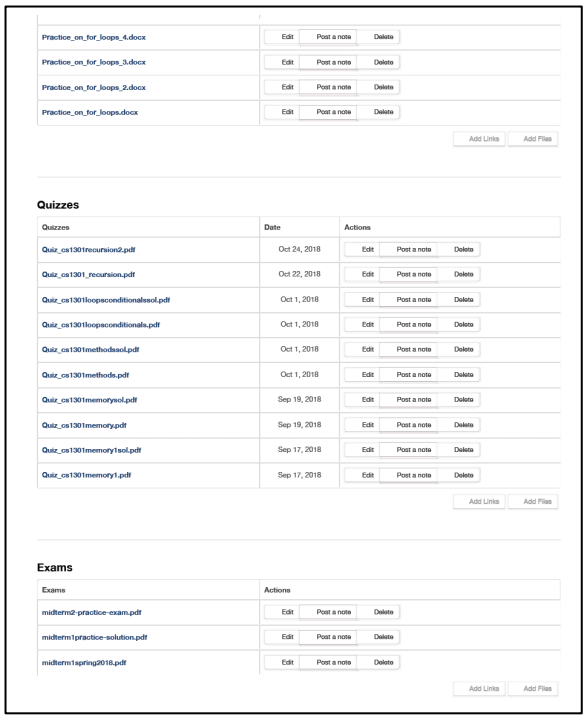
Below is a screen shot of piazza, showing the list of courses that I host on piazza for Fall 2018: CS1301, CS1101, CS2401, but I also use it for our on-going problem-solving course dissemination initiative and for my Google exploreCSR workshop participants.



Below is a print out of the resources section of my Fall 2018 CS1301 course: we can see that I share all relevant material with my students, including homework, in-class activities, lecture notes, quizzes and their solutions, exams and practice exams, etc.

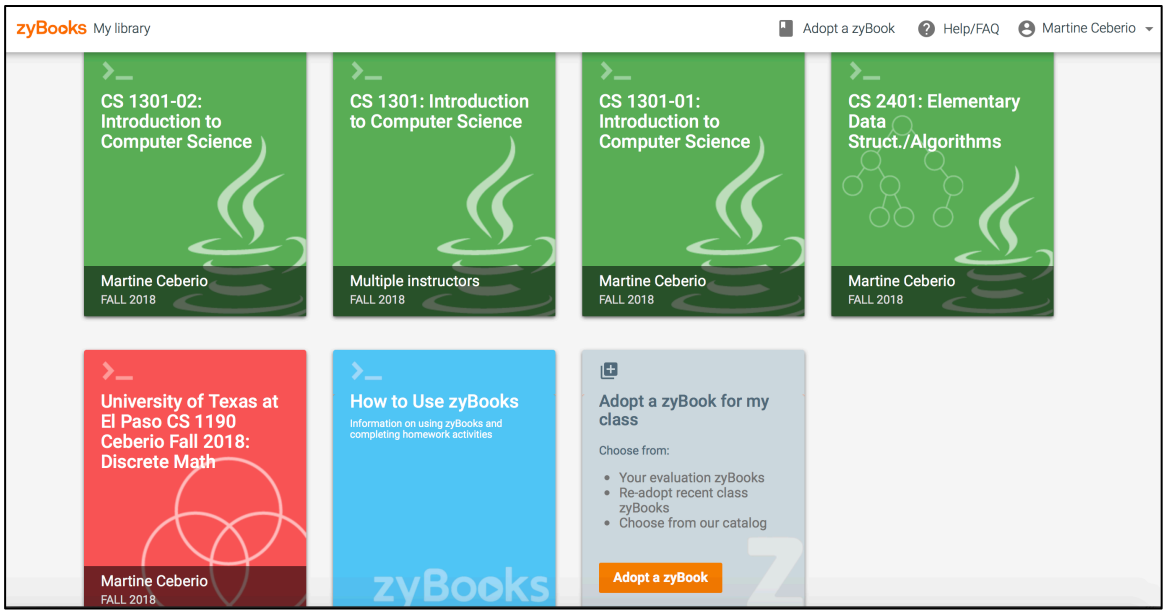






Online Automatically Graded Reading and Homework Assignments: Online textbook Zybook

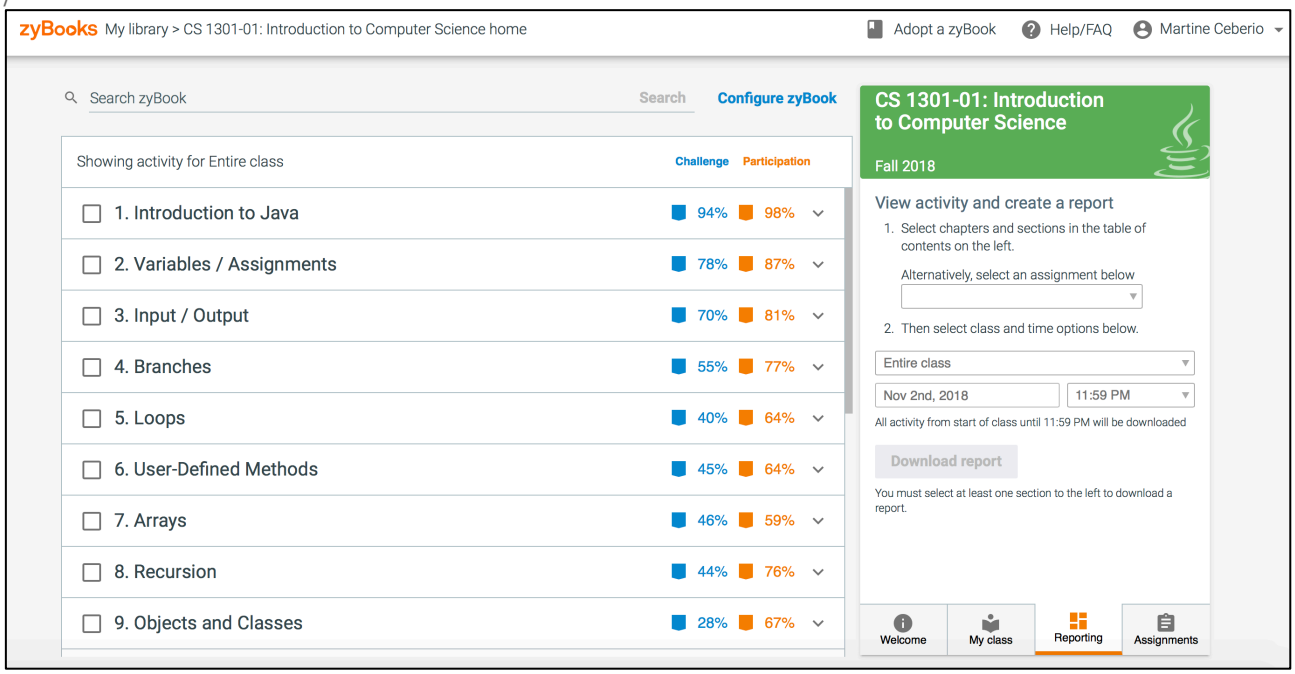
Below is a list of the online textbooks I use for my fall 2018 courses. We can see that I use a zybook for all my scheduled courses: CS1301, CS2401, but also for the discrete math course I supervise.



Faced with growing classroom sizes, I need to ensure that I can still assign timely and frequent homework to my students, and that they receive timely feedback on their homework assignments timely. Zybook online textbook provide high-quality

content and the ability for instructors to assign and track homework completion. Homework is blended into reading so I can track both homework completion and reading. As a result, I can assign homework, students can get instant feedback on their homework, and most importantly, I can act on the homework completion reports I receive: reports help me identify students who may be struggling or simply not adopting healthy study habits. Either way, this allows me to contact these students and propose help. Feedback and connection are essential to student success.

Below is a screenshot of a zybook view in which I can see the overall performance of the class on each section of the book. Individual performance is also available and invaluable.

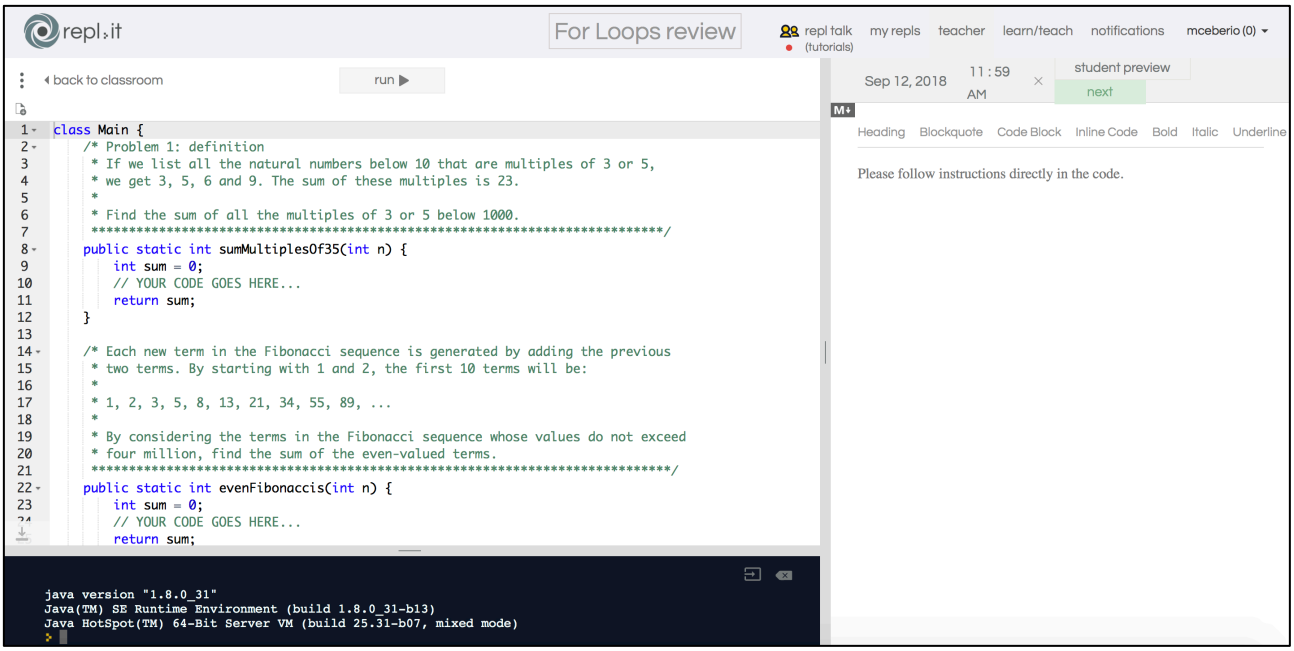


Online Quizzing Tool: Socrative.com

In order to best serve each of my students, I need to understand their struggles, their current standing, so that I can adjust the pace of my course and provide individual feedback. I find quizzes to be ideal to provide me with this sort of individual and global information, on an almost daily basis. However, with growing enrollment, I either should not give as many quizzes because it would take too long otherwise, or compromise my time for further help during office hours (or my TA's time, when TAs are pretty busy with large labs already). In short, traditional quizzing is pretty much impractical. In Spring 2016, I found Socrative to be a convenient online quizzing tool with automated grading. I design and keep all of my quizzes, and I can provide for each question some feedback to students who might fail any question. This tool has allowed me to keep the pace of my quizzes, hence of my understanding of my students' standing and struggles without compromising my time.

Instant Feedback and Automated Grading Online Tools: Repl.it, CodingBat.com & HackerRank.com

Easing the TA's work in lab so that they can spend more time one-on-one with students is essential with growing enrollments. I have started using tools that allow instant feedback with practice, such as CodingBat (for CS1301) or HackerRank (for CS2401), and I am always looking for new tools that are intuitive for our students to use. I explored mimir but did not find it easy enough yet, and I am now starting to use Repl.it (*see screenshot of a Fall 2018 CS2401 assignment below*). However, I need to keep exploring it to become fluent in it and efficiently guide students through it.



Course Creation and Revamping

In an effort to always serve our students better, I have created a number of courses over the years as well as constantly updated the courses I have repeatedly taught. My work, described earlier in this document, on the creation and refinement of problem solving courses illustrates my approach to continuous quality improvement. In what follows, I am describing my work on other two courses: CS1 (CS1301/1101) and Discrete Math.

Revamping CS1.

Since spring 2015, CS1 has evolved in many ways. We started by revisiting the outcomes of the course. We kept all the outcomes of the previous version of this course and we added a few more: namely, multi-D arrays, recursion, and linked-lists. The rationale behind these additions was that we wanted to expose the students to some key topics of CS2 (a.k.a., CS2401: Elementary Data Structures) at a level 1, because I had observed, while previously teaching CS2, how students would be frightened by the novelty of these topics when these topics were in fact not hard, but just had to be introduced in context in CS1 to provide students with the big picture of storage for instance or repetition. So instead of giving them the impression that "we are only going to do repetitions through loops and then, you'll see something more complicated called recursion", we introduced both concepts at the same time. Instead of telling them that "they could only store data in 1D arrays", we introduced the concept of multi-D arrays right away, while providing more practice on 1D and some on 2D. The result of this was that students showed less fear when covering these topics in CS2 and performed better. We also observed that, starting in spring 2015, we have been retaining women in CS1 at a higher rate than before.

In spring 2016, I received funding from Google to further the changes started in CS1. This second redesign focused on providing more tailored service to students. Namely, this is when I started using an online quizzing system, Socrative.com, to be able to assess my students more often despite growing class sizes. I started Saturday sessions with my students who wanted to catch up or go further than what we were studying in class. I used Google EngageCS resources (labs reviewed by educators) and I also worked on labs that I could contribute to Google EngageCS.

A major change was then brought by the departmental decision to split our original CS1401 into a lecture, CS1301, and a lab, CS1101, effective in spring 2017. This split was motivated by the fact that students often fail the course because

of poor performance in labs and we wanted to allow them to retake only the lab part (CS1101). This is to acknowledge that our students face unique constraints and challenges outside the classroom, having to work full time or supporting a family, their parents. Because of this, we often observed that students were struggling to meet the deadlines for labs, or even to complete the labs, making them fail both the lecture and the lab part of CS1401. By splitting it into a lecture and a lab, we offered more flexibility to our students.

We did have to somehow modify the outcomes of the course because of the split into CS1301 and CS1101. However, overall, the outcomes remained the same when looking at the two courses together. What I did, however, was that I designed a "layered" plan of instruction where I first teach all topics (except objects) at a high level, trying to make students connect their daily experiences to all the topics covered in CS1 (there are very natural examples of that), and then I came back to these topics more in depth. Each semester is therefore organized in 4 phases:

- Phase 1 -- general coverage of all topics. This phase is crucial to their realization that they already know most of what we are going to cover during the semester. For instance, I put a lot of emphasis in connecting all I present to situations in their daily lives.
- Phase 2 -- tinkering: we start using memory, conditionals, repetitions, methods, but mostly from given code and algorithms.
- Phase 3: doing -- the students can now implement their own solutions to problems, we also dig deeper into repetitions with recursion.
- Phase 4: creating -- while still using all we've learned so far, we go over objects and conclude with some implementation and the use of linked-lists. This "layered" approach allows me to go over concepts several times during the semester. It allows students to have more time and many more opportunities to acquire and demonstrate skills.

This last change was supported by UTEP's STEM Accelerator program and by our departments' NSF RED project.

More recently, inspired by my participation in Google's Faculty in Residence program in summer 2018, I effected more changes to CS1, as documented in the first part of this dossier.

Discrete Mathematics

Discrete Mathematics, a 3-credit-hour course, has consistently been a sore point of our CS students' curriculum. We have worked with the Math department to increase how explicit the relevance of its content is made, with improved student success.

The recent decision of the CS department to consider offering 1-credit-hour courses led to the proposal to offer Discrete Math in 2 pieces: one 1-credit-hour course focusing on propositional logic, sets & functions, and induction to be taken concurrently with CS1 so as to provide relevant support, and one 2-credit-hour course covering the rest of the 3-credit-hour course content (including: counting, more of predicate logic, combinatorics).

I took the lead in designing this new course sequence whose first part is offered for the first time in fall 2018. The second part will start being offered in spring 2019.

LETTERS OF SUPPORT

1. Letters of Support from students

The following are letters from one of my teaching assistants and three of my past undergraduate students:

Letter from Angel Garcia

Angel Garcia is a Ph.D. student in Computer Science. He has been my Teaching Assistant for CS1301/1101: Introduction to Computer Science for a several semesters.

October 22, 2018

University of Texas System Board of Regents
Outstanding Teaching Awards Selection Committee

To whom it may concern:

I am Angel F. Garcia Contreras, currently a Ph.D. student in the Computer Science Department at the University of Texas at El Paso. I first met Dr. Martine Ceberio back during my Masters studies, when she taught the Logical Foundations of Computer Science course. After that class, she invited me to join her research group, Constraint Research and Reading Group (CR^2G). She was the chair of my MS. Thesis Committee, and is now the chair of my Dissertation committee.

Throughout the years, I have known Dr. Ceberio mostly as a researcher. It is only when I started to work under her as a Teaching Assistant for her Introduction to Computer Science class that I truly got to know her as an educator. During my time as her TA, I have had a first-row seat into how she structures her class curricula and how she engages her students. As an educator, her priority is to ensure that students succeed at the course. This does not mean the students should just do the course work to get a good grade, but they learn the skills that will propel them to success in future classes and in their professional life. She structures her Intro to Computer Science curricula to make sure that the students understand not just the basics of programming, but of *problem solving*, a skill essential to Computer Scientists yet only recently identified and promoted as such.

Dr. Ceberio is an instructor that listens. She pays close attention to her students' performance, concerns, and needs. To make sure that her students learn and grow, she also seeks to improve her own teaching skills. Throughout the three semesters I have been her Intro to Computer Science Teaching Assistant, the learning outcomes remain the same. However, each iteration of her course involves changes to the curriculum, improvements based on her own reading and the techniques learned from education courses and workshops. Last summer she participated in a Google camp in which she learned techniques that she is introducing to the classroom. One such technique is role-based problem-solving, in which we assign the students a problem to solve, while each student has a different role with various responsibilities. The objective is to get them used to explaining out loud their thinking process while receiving feedback and being guided to a solution. The outcome of these exercises is to help the students learn to work with this type of problem-solving exercise, which is frequently used in technical interviews for big companies like Microsoft and Google.

Before being Dr. Ceberio's TA for Intro to Computer Science, I was a TA for junior and senior level courses. At that point in their degree, the students are used to programming, and to work under a more demanding course workload. My obligations for these courses and my interactions with the students were closer to what I would expect from a colleague in a professional environment. In contrast, students in the Intro classes are completely different: we have hopeful Computer Science freshmen, more experienced Engineering students who want to expand their skill set, even students from seemingly unrelated fields like Psychology and Liberal Arts who want to learn programming because it will help them with their research.

At the beginning, I approached the Intro classes in the same way I worked in the higher-level courses. I had high expectations for the assignments, and graded them accordingly. The first times I met with Dr. Ceberio to discuss the grades, she encouraged me to be less strict, to try to understand the students as they learn programming, which might be a radical departure from the the assignments and topics they had learned in the past. It took some time and more feedback to start developing my own teaching style, less concerned with trying to get the students to submit perfect assignments, and more focused in helping the students truly understand and learn the basics of programming. Her guidance has helped me see the need for a more engaging approach to teaching Computer Science.

It is thanks to my experience working and teaching under Dr. Ceberio's wing that I have come to discover a passion for teaching. She is not just my professor, but a mentor, and role model. I sincerely hope some day I can follow on her footsteps to engage, inspire, and teach students with the same passion, dedication and care that she has for her own teaching.

For all the reasons outlined above, I highly recommend Dr. Martine Ceberio for the University of Texas Regents' Outstanding Teacher Award.

Best regards,

Angel F. Garcia Contreras
Ph.D. Student
Department of Computer Science
The University of Texas at El Paso

Letter from Gerardo Uranga

Gerardo, a senior undergraduate student in Computer Science, was a student in my CS1301/1101 course in Fall 2015.

Dear University of Texas System Board of Regents Outstanding Teaching Awards Selection Committee,

It is an honor to be writing this letter in support of one of the most pivotal heroes in my life, Dr. Martine Ceberio, as a potential recipient for the Outstanding Teacher Award. I first met Dr. Ceberio back in the Fall of 2015 when she introduced herself as my Introduction to Computer Science professor for that semester. Unbeknownst to her, it was the second time I took the class, and more importantly, that was the semester I had just returned to college after having previously dropped out. With the uncertainties and emotional trauma I carried with me regarding higher education and my career choice of Computer Science, I did not believe that, at the time, these thoughts would all be addressed during one semester. However, throughout her lectures, Dr. Ceberio displayed such passion and dedication while teaching the subject material that I was persuaded to not only interact more in class discussions, but to begin reading external material to supplement my learning. Both of which I had never done in classes prior. She took notice of this effort, as I distinctly remember her adapting both a lecture and a lab assignment on string manipulation after I had proposed an optional solution of solving the problem she gave by looking at the individual characters of a string. This was the first time I had encountered this type of dynamic teaching, one in which the audience's knowledge, curiosity, and interactions impacted the way the professor taught and modified the course on the fly. I was not the only student this happened to. Because of this experience, I gained not only stronger motivation in both the education system and self-taught education, but the knowledge that Dr. Ceberio truly cared for my learning experience. Outside of her classroom, she provided additional extra credit work for completion of challenging homework activities that would greatly help in not only future CS courses, but also during technical job interviews. I not only finished the semester with an A, but had also been convinced to continue with my education.

She would later go on to strongly encourage and recommend me as a participant for a summer research internship at Stanford University, where I experienced a world I had never been exposed to before. This event would serve as the catalyst for a domino effect that began a series of events that led me to obtain a full time offer to work as a Software Developer at Microsoft upon graduation. I found out later that I was one of many students she would assist in obtaining not just the same internship, but also similar success in their career trajectory.

Dr. Ceberio, regardless of whether she is your teacher at the time or not, impacts your life in ways that it is difficult to express or return the favor for. The least I can do is support her selection for this year's Outstanding Teacher award, and I hope that the board can recognize the same qualities of excellence she holds as an educator inside and outside of the classroom as I do.

Sincerely,
Gerardo Uranga

Letter from Sebastian Nunez

Sebastian, a senior undergraduate student in Computer Science, was a student in my CS1190 course in Fall 2017. He was also an undergraduate instructional assistant for my CS1301/1101 course in fall 2018.

October 24, 2018

Dear University of Texas System Board of Regents Outstanding Teaching Awards Selection Committee:

It is with great pleasure that I recommend Dr. Martine Ceberio for the UT Regents' Outstanding Teaching Award. I became acquainted with Dr. Ceberio as her student for the course Problem Solving and Algorithms. A year later I began working under her as her instructional assistant for Introduction to Computer Science. As my instructor she showed a deep comprehension of the teaching material, and as my supervisor she continues to display this deep comprehension. She is an excellent instructor and supervisor.

Dr. Ceberio displays a deep passion for teaching and assisting her students. She is a "hands on" instructor that wants her students to learn through their own experiences, through her lectures she would introduce new concepts and different methodologies to my class. If we were confused on the topic she would demonstrate patience and collaborate with us to help us reach a solution. We would go through different problems in our class. When one of us had a question, she would answer the question in a way that the entire class could follow her explanation. I tend to have a rather unorthodox way of thinking which leads towards my problems and solutions requiring more contemplation than orthodox problems and solutions. For this reason, she would work with me to understand how I reached the problem and/or solution to make sure that I comprehended the material. Other professors tend to try and guide me towards an orthodox solution. I am very grateful towards Dr. Ceberio for her guidance throughout the course, especially in concerns to my unorthodox solutions.

As her instructional assistant I have become further acquainted with her teaching strategies. She wants her students to gain experience and truly understand the concepts being covered, this being the reason why they have in class assignments and review sessions. She is also very compassionate and empathetic towards her students, while remaining fair and objective. If a student is absent within reasoning she will catch the student back up on the material, and if needed create an appointment with the student around his schedule. This is one of her many strengths that I greatly admire, it inspires me to do my best to become someone capable of assisting my students in the same way.

In short, I highly recommend without reservation for the UT Regents' Outstanding Teaching Award.

Sincerely,

Sebastian Nunez

Letter from Sairy Cohen

Sairy, a junior undergraduate student in Computer Science, was a student in my CS1301/1101 course in Fall 2016.

Date: October 22, 2018
To: The University of Texas System Board of Regents Outstanding Teaching Awards Selection Committee
From: Sairy Cohen

My name is Sairy Cohen and I am a student at the University of Texas at El Paso. I am currently a Senior, with an expected graduation date of December 2019. I had the pleasure and opportunity to have Dr. Ceberio as my professor for Intro to Computer Science Fall 2016. As a student with no background in Computer Science, I had many difficulties starting out. Dr. Ceberio provided help by guiding and encouraging me to continue in this field.

As a freshman in college, classes can be really intimidating. Dr. Ceberio's teaching styles help us to get out of our comfort zone, work in pairs, discuss our solutions and ideas, and not be afraid to ask questions. Dr. Ceberio takes her position very seriously, and this is something that I admire about her. She is a hardworking person and a motivator, she constantly encourages her students to achieve their goals and learn more about themselves and their peers. Her activities were really challenging, but this helped us be more capable problem solvers. I can honestly say that Dr. Ceberio is one of the smartest and best professors at my university. She goes beyond the limit, helping her student succeed in this field. Dr. Ceberio is a professor that cares about her students, both academically and personally. Beyond that, I also had the opportunity to participate in her research in Spring 2017. This research was really challenging because as a Sophomore I did not know much about it, but she helped me by providing links and information to learn more. Having first-time experience in this area helped me discover and learn about different Computer Science fields.

Dr. Ceberio is consistently helping students to learn more about Computer Science, not only inside the classroom but outside campus. She served as my advisor for the Association of Computing Machinery Council on Woman (ACM-W) and Google's Ignite CS associations, where we thought elementary to high school students about Computer Science. She is encouraging elementary to high school students to pursue this career and provides information sessions to students to learn more about this career. I admire her effort in involving women in Computer Science. I can really say that Dr. Ceberio has impacted my life in many different ways. Dr. Ceberio truly believes in the potential in each and every one of her students and is willing to help them achieve it. She encourages them to look for opportunities even in their first year of college. She encourages all of her students to attend conferences and competitions. She even invites various speakers to motivate her students. Even though I am not her student anymore she has persistently challenged me to become a better version of myself, which ultimately helped me grow as a person and as a professional.

In conclusion, I have no doubt that Dr. Ceberio is a very outstanding and encouraging person. Her influence is present both in her students and the CS department as a whole. Truly a role model, Dr. Ceberio is one of the best professors this department has to offer. Dr. Ceberio is an inspiration to me and to others. This is why I think Dr. Ceberio is worthy of receiving the UT Regents' Outstanding Teaching Award.

If you have any questions, comments, or concerns, do not hesitate to contact me at secohencruz@miners.utep.edu

Sincerely,

Sairy Cohen

2. Letters of Support from Peers and Department Chair

The following are letters from colleagues whom I have worked with over the last few years:

- Dr. Christina Convertino: is an assistant professor in Teacher Education. Dr. Convertino has been part of the NSF-funded RED program (NSF IUSE/PFE: Revolutionizing Engineering Departments) I am a Co-PI of. In this role, and in the aim of understanding our computer science department context and practices in the classroom, Dr. Convertino has observed my Introduction to Computer Science course (CS1401 later changed to CS1301/1101) for 2 full semesters as well as two minimesters of teaching Introduction to Problem Solving (CS1190) that I had designed with Google (she actually also video-recorded a full minimester). In addition, she conducted focus groups with my students to further understand how they experience their curriculum in computer science.

- Dr. Heather Thiry: is a research associate at the University of Colorado, Boulder. Dr. Thiry is an external evaluator for the NSF RED project I am Co-PI of. As such, she has observed my classes on multiple occasions and has conducted focus groups with my students.

- Mr. Lorne Needle: is a lead for Scaled Learning and Partnerships at Google. Mr. Needle was our main point of contact at Google for the joint development of curriculum on problem solving, which I was involved in and of which I led the development of the Introduction to Problem Solving course. When I taught Introduction to Problem Solving in Fall 2017, Mr. Needle collaborated with me on the so-called Google end-of-the-semester project description. We also had multiple discussions about activities to be given to my students, which resulted in some new activities like a blind (or magic) labyrinth team challenge. He also attended the students' final presentations and participated in their evaluation.

Letter from Dr. Christina Convertino

THE UNIVERSITY OF TEXAS AT EL PASO

UTEP

Teacher Education

October 26, 2018

Dear UT System Regents' Outstanding Teaching Award Committee:

I am pleased to write this letter of support for Dr. Martine Ceberio's portfolio, and I highly recommend her as a recipient of the Regents' Outstanding Teaching Award. I have known Dr. Ceberio since 2014 when we first met and began to work together on a NSF funded project to transform computer science education. Since that time, I have observed her Introduction to Computer Science course on multiple occasions. During the fall 2015 semester, I observed the course as part of research related to the NSF project. In spring 2017, Dr. Ceberio asked me to conduct an observation of the introductory course as part of a peer teaching evaluation. During spring 2017, I observed her Problem Solving course. In this letter, I address the course observation, which occurred as part of the peer teaching evaluation.

Introduction to Computer Science is a required course for computer science majors, however, students pursuing other majors, e.g. engineering, also enroll in the course. On the day that I conducted the observation for the peer teaching evaluation, there were approximately 40 students in the class. The objectives for that class meeting, which Dr. Ceberio communicated at the outset, were to complete a prior activity on "code tweaking" and then to introduce "testing." During the one-hour and twenty minutes of class time, I observed multiple instructional practices that exemplified thoughtful, engaging, and student-centered teaching. In particular, and what I want to emphasize in this letter was how Dr. Ceberio used communication to engage and assess student understanding throughout the entire class meeting.

First, Dr. Ceberio used a conversational style of communication, which created a classroom environment, where students were notably comfortable asking questions, and sharing their ideas. She was also very strategic and thoughtful in her use of communicative cues to guide students in making connections with concepts and activities that had occurred in prior class meetings, and which, would be important in the future. For example, she used phrases, like," last time, you had a main method, once you understand you will go to the next variation of this method (…)." Midway into the class meeting, it was obvious that the complexity of content had increased. In response, Dr. Ceberio begin to use communicative cues, such as, "what is important is the building block in this code, this line is crucial it tells you how to reverse characters," while she simultaneously used the white board to visually demonstrate how to solve the problem. In this example, what was most effective and insightful about Dr. Ceberio's pedagogical approach was that she was intentionally

El Paso, Texas
79968-0574
(915) 747-5426
Fax: (915) 747-741

scaffolding student learning using multiple supports, i.e., communicative cues and visual representations of the process.

There is no doubt from my observations that teaching Introduction to Computer Science to undergraduate students who are eager to code and program but instead are required to learn complex, yet fundamentals skills and foundational knowledge is a challenge. However, because of Dr. Ceberio's clear commitment to her students as well as her apparent understanding about students, she made an explicit effort to bridge what might have otherwise been a disconnect between students' immediate desire to code and the focus of the introductory course, as when she told students, "What I meant to teach you is making you become more fluent in taking some code and tweaking it, this is important because you will find yourself doing this a lot, you will get some code from some company, or it is your code but your client wants a change fast so you have to know how to do this." In another example, she made explicit connections between the reason students need to learn the material (which, she admitted can be tedious at times), and their future professional lives. "We teach you to do this but then when you get into the world, you will have to use this," after, which she described examples of people who design tests for plane and cannot ask for redundant tests, because of cost and time. Thus, validating students' interest, academic learning, and professional development.

In sum, Dr. Ceberio demonstrates high levels of dedication, interest, and engagement in her teaching. She is a reflective practioner who is responsive to student learning in the moment, and is willing to make adjustments in her class objectives in order to scaffold student learning. She uses humor and a personable affect to encourage student engagement, and she knows when and how to use examples to illustrate complex ideas. I also know that Dr. Ceberio consistently engages in professional development as well as course development and revision as a routine part of her teaching practice. Dr. Ceberio is a role model for faculty, who are committed to student learning but need to understand more about what it takes to practice outstanding teaching. That said, Dr. Ceberio has my highest recommendation for University of Texas Regents' Outstanding Teaching Award.

Sincerely,

Christina Convertino, PhD
Assistant Professor of Sociocultural Foundations
College of Education

Letter from Dr. Heather Thiry

University of Colorado Boulder

ETHNOGRAPHY & EVALUATION RESEARCH
CENTER TO ADVANCE RESEARCH AND TEACHING IN THE SOCIAL SCIENCES
UNIVERSITY OF COLORADO BOULDER
580 UCB
BOULDER, CO 80309-0580

October 24, 2018

Dear UT System Regents' Outstanding Teaching Awards Committee,

I am writing in enthusiastic support of the selection of Dr. Martine Ceberio, Associate Professor of Computer Science at the University of Texas El Paso (UTEP), for the University of Texas System Board of Regents Outstanding Teaching Award. I have known Dr. Ceberio for several years through our involvement in national and departmental computer science education initiatives, and I have seen first-hand her unwavering commitment and support to her students and her teaching.

As part of the external evaluation of two National Science Foundation-funded initiatives, I have observed her courses, interviewed Dr. Ceberio about her approach to teaching, and interviewed her students about their experiences in her introductory computer science courses. As a result of this data gathering, I most enthusiastically recommend Dr. Ceberio for this teaching award. Dr. Ceberio is deeply dedicated to her students and uses a variety of strategies to ensure their learning. No student is left behind in her computer science courses and many of her students credit her introductory course with their decision to major in computer science.

In particular, I have observed Dr. Ceberio's CS1 course, the introductory course in the computer science major. This course is known nationwide for having high failure rates, prompting many talented, but underprepared, students to leave the major before they have even begun. At UTEP, many of the entering computer science majors have had little prior exposure to computational thinking or computer science. Dr. Ceberio recognizes the limits of many students' incoming knowledge and experience and approaches her CS1 course with an eye toward helping all students develop the abilities and capacity to succeed in the computer science major. She cultivates a growth mindset in students by assuring them that they are capable of mastering computer science, even if they have little prior experience. When I observed her course, Dr. Ceberio continually encouraged students, commenting while presenting new material, "If you feel like you don't get this, don't worry. More practice is coming." She constantly reiterated to students that they could master challenging material through practice and she provided the instruction and support to facilitate their mastery. Throughout the lecture session, Dr. Ceberio continually encouraged her students, "You can do this. You will get this with practice." Over the course of the class, I could see a positive change in the demeanor of students who initially looked worried or frustrated about the challenging new material. By the end of the class, these initially hesitant students were jumping into problems and sharing ideas and solutions with their partners.

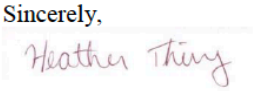
Dr. Ceberio has also embraced evidence-based practices in her teaching. Dr. Ceberio employs an active learning approach and allows students the opportunity during every class to practice their new computing skills with peers. Active and collaborative approaches are still not common within computer science, yet research has consistently shown that they lead to stronger learning outcomes and increased student engagement. Indeed, I observed her courses using a rubric developed for observation in postsecondary STEM classrooms. Using this rubric, Dr. Ceberio's students were rated as highly engaged during 92% of course time, relatively rare levels of student engagement for STEM courses. Dr. Ceberio fostered such high engagement through her clear, practical and interactive mini-lectures, and by providing students the opportunity to apply their new learning in hands-on, collaborative exercises.

Dr. Ceberio thoughtfully engages her class in active learning and explicitly guides her novice students through the computational thinking processes that they will need to approach computing problems. For instance, her instructions for group programming activities clearly guide students through the steps of computational thinking in the following way: "[WHAT] Design a tool that allows meaningful interaction with a user. [HOW] What should the tool do? [CODE] Write an algorithm of the tool." Success in computer science rests on the "How" element of this process, and Dr. Ceberio repeatedly emphasized this for her students, noting "Make sure you think about the how. If you don't know where you are going, you are not going to get there. The important part is the planning and idea generation." In interviews, her students stated that through this approach, "I'm understanding what is being presented, but also *how* to do it." In this way, Dr. Ceberio fosters a collaborative and creative atmosphere in her class where students can freely generate ideas, debate strategies, and implement solutions to complex problems.

Dr. Ceberio is always available to help and support students, both in and out of her classroom. She is very attendant to students during group activities and will help to guide struggling teams toward solutions, not by feeding them a solution but by encouraging their critical thinking skills to come up with a solution on their own. One of her students stated in an interview, "She will always notice if I am struggling and ask if I need help. This support gives me hope of graduating in computer science." Students also commented that they feel that Dr. Ceberio is one of the most approachable faculty in the department and they always feel that they can go to her for help outside of class. Some students have credited their persistence in the major to Dr. Ceberio's exceptional instruction in the CS1 course, but also to her continued interest in them as they advance in the major. Dr. Ceberio clearly cares very deeply about her students and her teaching.

For these reasons, I most enthusiastically recommend Dr. Martine Ceberio for the Regents Teaching Award.

Sincerely,



Heather Thiry, Ph.D.
Research Associate
Ethnography & Evaluation Research
University of Colorado, Boulder

Letter from Mr. Lorne Needle

Google

Letter of Recommendation for Martine Ceberio for the UT Regents Outstanding Teaching Award

Google Inc.
1600 Amphitheatre Parkway
Mountain View CA 94043

650 253-0000 main
google.com

October 25, 2018

Dear UT Regents:

I am honored and delighted to recommend Professor Martine Ceberio of the University of Texas at El Paso (UTEP) for the UT Regents Outstanding Teaching Award. As a formal partner and peer I've had the opportunity to work closely with Professor Ceberio and observe her work and impact for the last 18 months. In my experience she is an exceptional educator who's able to not only create and refine excellent curriculum, but also deliver it successfully with her highly effective, engaging, and accessible classroom teaching approach and presence.

I began working with Professor Ceberio in early 2017 as part of Google's partnership with the Computing Association of Hispanic Serving Institutions (CAHSI) to develop and deliver a series of courses in problem-solving. Dr. Ann Gates of UTEP is one of the co-chairs of CAHSI, and she and UTEP served as the leads for this critical and high-profile project. Therefore it was imperative that UTEP identify a top faculty member to build the courses, and it quickly became evident to me why Professor Ceberio was chosen. Her creativity and expertise in curriculum design, her skilled teaching, and her ability to test and then refine course material with students were all essential to developing the courses.

Professor Ceberio brought an informed perspective and first-hand experience to the problem-solving courses. Independent of our project she took the initiative to start a problem-solving club at UTEP in 2015. It was valuable for students and by 2017 had evolved into a 3-credit hour summer course. That prior work became an important input to the set of three 1-credit hour courses that CAHSI and Google collaborated to build. In particular she created and was the first to teach UTEP's CS1190, "Problem Solving 1." Through the spring and summer of 2017 Professor Ceberio worked closely with me, faculty from two other universities, and a Google engineer to create the courses. She and I also had multiple conversations where we discussed the course content in detail. She provided thoughtful, detailed drafts of the course outline and materials, and was always well-prepared for the team's meetings. Her skill with pedagogy became apparent in her selection, sequencing, scaffolding, and adaptation of course activities. Professor Ceberio demonstrated a powerful balance of justified confidence in her own sound approach with openness to the rest of the team's input and materials, including real-world practice problems generated by Google. It was also evident that the faculty from other schools, some senior to her, respected her skill and took her input seriously. As we've moved forward and refined the courses each semester, Professor Ceberios has been an important voice in our assessment of what worked and what needed to be refined. In fact she

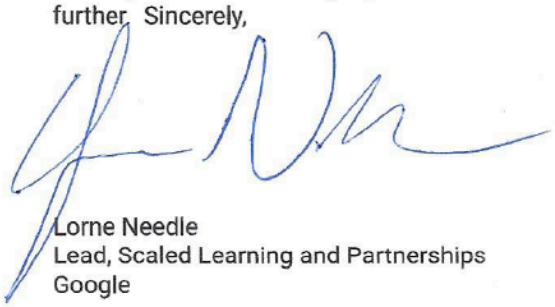
has written several reflections on the courses that have been useful inputs to the evolution of the curricula.

During fall 2017, once the courses were ready, I had the opportunity to observe Professor Ceberio teaching CS1190, and to serve as a judge and provider of feedback for students' final project presentations. I saw her do a number of things that are the hallmarks of a great educator. Professor Ceberio has a unique ability to balance rigor and demanding material with kindness and a spirit of inclusion and collaboration in the classroom. For example she appropriately pushed the students to learn and apply the IDEAL problem-solving framework we had selected to anchor the course. When students fell short, she gave them detailed and thoughtful feedback, and expected them to try again and deepen their learning. She did this with a warmth and empathy that motivated the students to persist until they learned and could apply the tools and skills.

As a result, it was clear that students had increased their proficiency with problem-solving by the end of the course. In fact, in the follow-up evaluation conducted by CAHSI, students reported both that they had learned new skills and thoroughly enjoyed the course. In my observation Professor Ceberio is able to achieve this balanced and impactful teaching style because she takes an asset-based approach, which she learned in part as a UTEP EDGE Fellow. She truly believes her students are talented and capable, and even when they struggle she focuses on their skills and kernels of success they can build from, rather than deficits.

I was so impressed with Professor Ceberio's accomplishments with the project that I recommended her for Google's Faculty in Residence program during summer 2018. This is a highly competitive program where Google hosts faculty from around the country to get exposure to Google's cutting-edge products and technology, which they use to design new curricula. There were many people I could have recommended. However given her outstanding work Professor Ceberio was at the top of my list. I believe it was a great professional development experience for her and that it helped her become an even more effective educator.

Thank you for considering my recommendation. I'm available if you'd like to discuss anything further. Sincerely,



Lorne Needle
Lead, Scaled Learning and Partnerships
Google

Letter of Support from Computer Science Department Chair: Dr. Ann Gates

THE UNIVERSITY OF TEXAS AT EL PASO

Ann Q. Gates
Chair
Computer Science

November 2, 2018

Dear Review Committee:

It is with great pleasure that I recommend Dr. Martine Ceberio, an Associate Professor in the Department of Computer Science at the University of Texas at El Paso (UTEP), for the UT System Regent's Outstanding Teaching Award. Dr. Ceberio is an exemplar educator who works tirelessly to ensure that students succeed in the computer science (CS) program. She excels in the classroom through her pedagogy and curriculum development, involves high school students in her research group, mentors and develops undergraduate research students, sponsors the female student organization, and contributes to departmental and college community outreach activities.

Dr. Ceberio was recently named to the first cohort of UTEP EDGE Fellows, who are committed to holistic student success through student development, co-curricular engagement, and professional preparation. In addition, the Fellows bring collective knowledge, expertise and experiences that serve as a basis for promoting an asset-based approach to student success across campus.

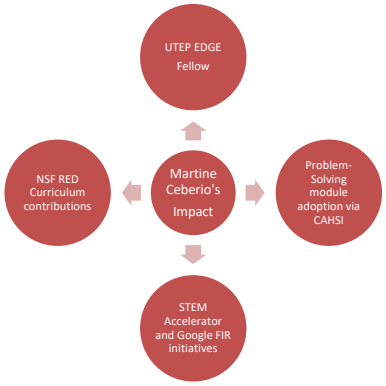


Figure 1: Summary of the impact of Dr. Ceberio's work across local, state, and national initiatives.

As shown in Fig. 1, Dr. Ceberior's work has been synergistic and impactful to the university, state, and national efforts focused on preparation of a diverse STEM and computing workforce. The remaining paragraphs provides a summary of her contributions to support my endorsement.

Teaching. Dr. Ceberio has taught a variety of courses at all levels across the undergraduate and graduate curriculum over the last 15 years, and her student evaluations have consistently been above average. With her strong mathematical background, she excels in the theoretical classes that she teaches, e.g., Automata.

The courses in which she has had the most impact are the Introduction to Computer Science (CS1401, recently changed to CS1301+CS1101, and referred to as CS1) and the newly created Introduction to Problem Solving (CS1190). Dr. Ceberio's CS1 course employs an asset-based approach, i.e., one that acknowledges the assets that the students bring to the course. The structure and activities that she introduced to CS1 builds students' problem solving, confidence, and team skills, providing students' EDGE advantages that contribute to their success.

In 2016, Dr. Ceberio received competitive funding from the College's STEM Accelerator initiative focused on supporting regional teams of education and workforce partners to increase the number of students who will earn a STEM credential. The initiative provided professional development opportunities that included training on problem-based learning. Dr. Ceberio was also selected to participate in Google's Faculty in Residence program, which exposes faculty to software engineering practices and pedagogy with the aim of redesigning a CS introductory course that includes motivating and engaging activities that build students' competencies. The introduction of these high-impact practices is improving retention and workforce preparation of our students, thus, contributing to the goals of the initiatives.

In 2016, the department received from the National Science Foundation from the Revolutionizing Engineering and Computer Science Education (RED program), which emphasizes establishing innovative and inclusive departments and developing technical and professional threads across the major. As described earlier, Dr. Ceberio has taken the lead in redesigning the introductory courses. The RED program's external evaluator and ethnographer, who conduct observations, surveys, and focus groups with students, confirm the level of engagement, care, and rigor that Dr. Ceberio brings to her course (refer to letters). Her efforts contribute to the departmental goal of creating inclusive departments and creating experiences to support students' identity as a computer scientist.

At a 2016 faculty retreat, the department recommended creating one- and two-credit-hour courses in problem solving and breaking the current three credit-hour Discrete Mathematics course into one- and two-credit-hour courses. The intent of the problem-solving course is to connect freshmen to the department by providing them an additional CS course that coaches them on strategies that transfer to how they tackle programming assignments. The intent of the latter is to allow students in the CS1 course to learn propositional and predicate logic that is currently covered in the sophomore-level Discrete Mathematics and to reinforce the application of logic in programming constructs learned in CS1.

Dr. Ceberio stepped up to develop the pilots for both courses. The problem-solving course became part of an initiative with Google and the Computing Alliance of Hispanic-Serving Institutions (CAHSI), an NSF-funded national INCLUDES alliance centered on the recruitment, retention, and advancement of Hispanics in computing. Dr. Ceberio developed the curriculum for the one-credit *Introduction to Problem Solving* course, and other CAHSI faculty developed the curriculum for the *Computational Problem Solving* and *Algorithmic Problem Solving* courses. The process of developing the courses included attending a workshop and video-conference calls with Googlers, Dr. Johannes Strobel, a renowned engineering educator, and other CAHSI faculty. The courses are being piloted and adopted by thirteen institutions (see https://events.withgoogle.com/cahsi-problem-solving-courses/#content).

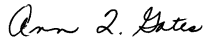
Extracurricular Activities. Dr. Ceberio is a strong advocate for the participation of women in CS as evidenced by her leadership in establishing the ACM-W student chapter and leading the National Center for Women & Information Technology (NCWIT) Aspirations in Computing activities. She also advises the Society for Industrial and Applied Mathematics (SIAM) student chapter, the CS CLIO chapter that focuses on outreach, and the Harmony Miners Association (alumni from Harmony Science Academy).

Undergraduate Research. Dr. Ceberio has an outstanding record of involving and mentoring undergraduate students in her research group. She helped students secure funding from CAHSI, COURI, LSAMP, and REU programs over the years. Refer to her CV for a list of publications with student co-authors many of whom are undergraduates. In addition, Dr. Ceberio has involved high-school students in research during summer through the College of Engineering's Nexus program.

In closing, I encourage the selection committee to recognize Dr. Ceberio for her efforts centered on student learning and success. She is an exemplar for what it means to be an outstanding educator inside and outside the classroom and a role model for how to integrate research and education.

Please feel free to contact me if you need any additional information at agates@utep.edu.

Sincerely,



Ann Gates, Ph.D.
Chair of the Computer Science Department
AT&T Distinguished Professorship

500 W. University
Admin 209
El Paso, Texas
79968-0500
(915) 747-5680
FAX: (915) 747-6474

SUPPLEMENTAL MATERIALS

1. Student Evaluations of Teaching

Since spring 2015, I have taught a number of undergraduate courses, including one that was cross-listed with a graduate course (CS4365/5354). Below is a table summarizing the list of these courses by semester, their enrollment, along with the ratings received for both instructor and overall course.

Term	Course#	Course Title	Enroll.	Instructor	Course	N
Spring 2015	CS1401 CS3350	Introduction to Computer Science Automata	42 48	4.55 4.0	4.82 4.0	22 13
Fall 2015	CS1401	Introduction to Computer Science	39	3.4	3.6	10
Spring 2016	CS1401	Introduction to Computer Science	47	4.12	4.06	18
Summer 2016	CS4365	Topics in Soft Comp.: Problem Solving and Algorithms	10	4.5	4.33	6
Fall 2016	CS1401	Introduction to Computer Science	45	5.0	5.0	5
Spring 2017	CS1301	Introduction to Computer Science	45	4.44	4.44	9
Fall 2017	CS1301 CS1190 CS1190	Introduction to Computer Science Introduction to Problem Solving Introduction to Problem Solving	51 12 6	4.44 5.0 N/A	4.44 4.0 N/A	9 1 N/A
Spring 2018	CS1301	Introduction to Computer Science	50	4.11	4.55	9
Fall 2018	CS1301 CS1301 CS2401	Introduction to Computer Science Introduction to Computer Science Elementary Data Structures and Algorithms	50 40 13	4.4 N/A	4.5 N/A	5 N/A

Note that the decline in the rate at which students complete course evaluations (as visible in my table) coincides with our institution's change from paper evaluations to online evaluations of courses.

Comparison to relevant department ratings is only available for fall 2016 and spring 2017 when department average ratings were as follows:

Term	Average Instructor Rating across CS Department	Average Course Rating across CS Department
Fall 2016	4.30	4.31
Spring 2017	4.28	4.20

From the students' comments on the course evaluations, I consistently receive comments about my availability and patience in explaining concepts / answering questions. Here are a few examples of such comments (verbatim):

- CS1401 – Spring 2015:
  - Instructor is great, they know what they are talking about and knows how to explain confusing concepts and make them seem simple.
  - This course provided a challenging environment that pushes a student to learn so much during the semester which was very fun.
  - The instructor was really nice, very informative, and she really helped me to adjust to a new environment. I came from an early college and I was blown away by how different the worlds are.
  - Dr. Ceberio was an amazing professor. Explains the concepts very well, and extremely helpful.
  - The instructor is always available during her office hours and she always explains everything with very simple examples. Also, the instructor explains everything in a very simple way during class for full comprehension. When the problem is hard to solve, she explains to everybody how to solve it step by

step, but of course she always make you strive to find the answer by yourself by just leading you to the next step.
    - Best class Ive taken at UTEP! Professor Ceberio is fair and very helpful.

- CS3350 – Spring 2015:
    - She seemed genuinely interested in making sure we were learning what we were supposed to be learning
    - She is easy to pay attention to, very patient with questions, and is very nice and respectful. Her notes were very easy to follow, and she provided numerous examples that accelerated my learning.

- CS1401 – Fall 2015
    - Very helpful instructor. Possibly best instructor I've had to date.

- CS1401 – Spring 2016:
    - she was very organized on her lectures, and she was always encouraged participation and questions we may have had in class. everything from having to follow instructions and doing homework, to the labs, were very great and helpful in funding our understanding of the class overall. i know that it will prepare me for future courses i may need to take in order to continue to pursue my career. overall, i loved the class and the professor, im really going to miss it.

- CS4365 — Summer 2016:
    - The materials of the course will not only prepare the students for other courses but also for the interviews and improves the thought process for problem-solving. This is an awesome course.
    - This should be a regular semester course instead of summer.

- CS1301 – Spring 2017:
    - Great Course

- CS1301 – Fall 2017:
    - She is knowledgeable and openly welcomes questions. She is extremely dedicated to her students and learned a lot. I was thankful to have her as my professor.
    - Great class, Dr. Ceberio really cares.

- CS1301 – Spring 2018:
    - Dr. Ceberio and her teaching assistants were extremely helpful throughout the semester. Every question I had was answered diligently with good explanation and patience.

2. Interaction with undergraduate students beyond the classroom

As mentioned in my teaching philosophy, I believe that teaching and mentoring are a way of life rather than something that only takes place in a classroom or in formal settings. I believe that if we all work together, students, teachers, and I will only become better, train better new generations of engineers, professionals who are more likely to collaborate (seeing other do just that), who do not silo themselves into their own discipline but instead explore and identify connections and needs. As a result, I take on every opportunity to mentor students and teachers of the area.

Because I believe in it, and even more so as a UTEP EDGE Fellow, I take advantage of every encounter with students (in a hallway, at the local campus coffee place, in an elevator, etc.) to interact with them, probing them about their current courses, whether they are doing fine or not, need help or not, are involved in research or not, etc., and I use these few minutes to help connect them to relevant resources (help, research mentor, advisor, etc.). This is so informal that it is almost impossible to document with hard evidence.

However, I also mentor students and teachers in settings that are slightly easier to document. This is what I describe in the next few sections of this document.

Undergraduate Research Mentoring

Within my research group, CR2G:

Since I joined UTEP back in 2003, I have always invited and included undergraduate students in my research group (CR2G, Constraint Research and Reading Group: cr2g.constraintsolving.com). From my own experience as a student in France, undergraduate students were never involved in research and I saw no reason for it to be this way. Instead, I believe that, to be successful in research, motivation is the best "skill" anyone needs. So once I became a professor, I made sure to invite and include undergraduate students in my research group. Often, students will approach me about joining my research group, wondering what they should know, at what level in their Computer Science curriculum they should be to be able to join my group. I always make a point to explain to them the way I see it: knowledge can be learned, motivation not so much. If they are motivated to contribute to a research area, they will be able to fill in their knowledge gaps. I also insist, when meeting students about to embark in research for the first time, that if they are not successful, it is most likely because the research field was not a good fit for them, not because they are not suited for research.

Concretely, I have had an average of 5 undergraduate students in my research group per semester consistently over all semesters since joining UTEP. This number peaked at 12 in Fall 2016 / Spring 2017. It is interesting to note that not all of my undergraduate students are computer science majors. Students talk about my research group among each others and students from other majors (Physics, Math, Mechanical Engineering, mostly) have been part of my research group over the semesters. This is a great opportunity for me to reinforce the fact that we are not to work in silos but rather to seek collaboration and synergy with other disciplines.

Over the semesters, I have been able to fund some of these students via NSF REU Supplemental funding (for the NSF CAREER grant I received, 2010 – 2017). However, I mentor my UG research students to seek scholarships and they have been particularly successful at obtaining them: every semester since early 2015, I have had between one and three students recipients of scholarships from UTEP's Campus Office for Undergraduate Research Initiative, from UTEP's LSAMP, or receive UG research stipends from the Computing Alliance for Hispanic-Serving Institutions (CAHSI).

In addition to my undergraduate students' success in being funded, joint publications further illustrate my level of involvement with my undergraduate students. In my Curriculum Vitae available at the end of this document, I highlighted in blue all publications that are with at least one undergraduate student and I underlined the names of the undergraduate

students involved. Over the last 6 years (since the beginning of 2012), I had 18 joint publications with undergraduate students. My undergraduate students also present at least once yearly at a research symposium or workshop, and possibly (depending on the student's involvement) at a national venue.

Outside of my research group:

Not every student is interested in pursuing the type of research I conduct, and this is absolutely fine. However, all students should be able to discover what research is. Every semester, when I advise the group of CS students assigned to me for academic advising (about 50+ of them), I make sure to probe them about research: are they involved in research or not yet, have they ever considered joining a research group, do they know what research is, etc. I encourage those not yet involved in research to consider trying it and provide some guidance about how to pick a research area and group. I then offer to facilitate a meeting with their desired research mentor. Over the semesters, the students being amenable to trying research has seemingly grown.

Recently, I applied to and received a Google exploreCSR grant: this grant is a pilot program from Google that aims at encouraging undergraduate women in computing to consider graduate school and research. This grant is a great opportunity for me to be able to plan a year-long program for undergraduate women in computing, which consists of:

- A 2-day workshop on campus; and
- A follow-up yearlong webinar series and research mentoring.

This effort is a partnership of three regional institutions: UTEP, New Mexico State University, and El Paso Community College. We expect between 70 and 80 participants for our on-campus workshop at UTEP, during which we will go over what graduate school is, the potential struggles but most importantly the expected benefits of it, as well as research-onboarding activities and pairing with mentors. We expect that this program will contribute to increasing the number of women seeking graduate studies in computing-related areas, and the success rate and advanced degree attainment rate for women transitioning from EPCC. More generally, we anticipate that holding this workshop will promote practices that are conducive to inclusiveness and increased confidence in students from underrepresented groups.

Student Organizations Advising

Advising student organizations is something I discovered when joining UTEP. This is not something that is common practice in France where I completed all my studies. As a student, I was involved in student organizations and in University governance, but never had a faculty advisor to provide guidance. Early on during my tenure at UTEP, I have been tasked with advising a student organization. I discovered how beneficial this could be. Advising organizations is yet another way to interact with students and yet another platform to provide mentoring.

I started by advising our local ACM (Association for Computing Machinery) student chapter. ACM is a major organization for computer scientists and our department has had a chapter for many years. After advising it for several years, in spring 2012, I handed it over and in summer 2012, created a new chapter: UTEP ACM-W student chapter, the part of ACM that is concerned with the participation of women in computing, which I am passionate about improving. The chapter was born and students were on board with it. This chapter contributed to a clearer communication to students that the participation of women in computing is important and necessary, but yet a sore point of the field that needs to be addressed as best as we can.

Concretely, the chapter has been very active and has been successful in acquiring funding for its initiatives. In spring 2014, they received NCWIT Seed Funding to develop a mentoring program for CS Undergraduate students. In spring 2016, they decided to put a lot of their chapter's focus on outreach to local schools to raise awareness about computer science and break perceived gender barriers. They applied and were awarded Google IgniteCS funding to put in place formal and informal programs to teach computer science to young students at a middle school of El Paso. In spring 2017,

they received their second Google IgniteCS award for a similar program at a high school of El Paso. They remained active with Google IgniteCS programs until Google closed the program in spring 2018. Yet, they decided to continue holding the program, just now calling it igniteCS and no longer Google igniteCS.

As I handed this ACM-W chapter to a colleague after 6 years advising it, the former officers decided to create a new student organization, **CLIO**, entirely dedicated to outreach programs. They are still running their igniteCS program (currently at Bowie High School, El Paso) and the CLIO officers and I are working together on putting together a proposal to start an AI4All program to be held in summer 2019. This program aims to involve young students (high-school) with AI and Machine Learning as early as possible. From my interaction with local K-12 teachers, I plan to include some of them as well as mentors and yearlong advocates at their respective institutions.

Additionally, I am advisor to two more student chapters: the SIAM student chapter and the Harmony Miners Association. Both chapters are composed not only of undergraduate students and are not only computer scientists either.

K-12 Students Mentoring

In 2010, I had the opportunity to partner with the local Community College (EPCC) and participate in a research-mentoring program for high-school students during summer. This experience opened my eyes on the potential of involving students in research even earlier than when they start their higher education journey. Involving students much earlier allows us to spark interest earlier and keep the interest up so students eventually may decide to join a computing degree. This is essential for recruitment.

Since then, I have been involved in the following efforts/programs:

Below is a picture from the 2017 award ceremony at UTEP.

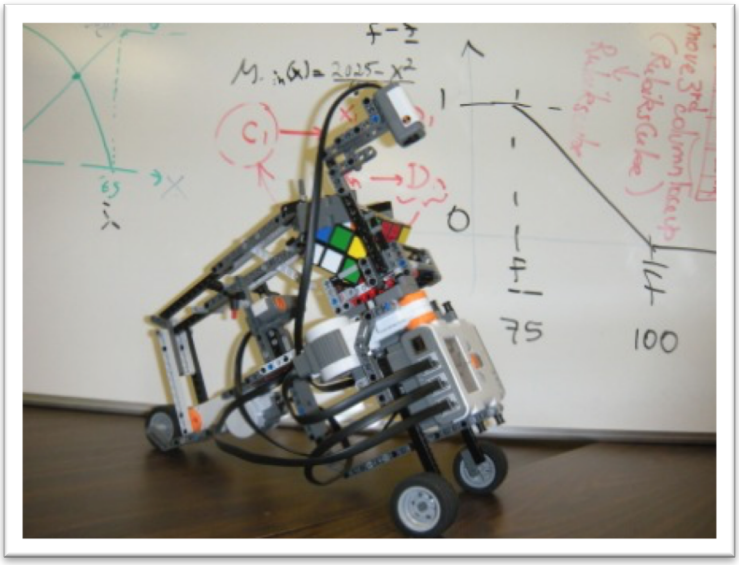
- ***NCWIT AiC:*** NCWIT is the National Center for Women & IT. AiC is its annual program for high-school young women with aspirations in computing. This program is instrumental in retaining these women in the field at a time (high school) when studies show that their interest in computing drops at higher rates than at any other time in their education. This program builds on regional affiliate programs who recruit participants from their local communities to participate in the program / competition. Participants hence enter two competitions: one

locally and one nationwide, with then two chances to be recognized. I was the coordinator of the El Paso NCWIT AiC Regional Affiliate from 2011 to 2018. During that time, we were able to raise awareness about this program to local schools and teachers through outreach programs and collaborations and grow the number of participants (25 in 2017-2018). My involvement consisted in:

- Reaching out to schools (counselors, teachers, principals) and students (from other programs I run) to attract participants;
- Evaluating applications nationwide;
- Ranking my regional applicants based on evaluations of their applications;
- Organizing an annual award ceremony (fundraising and logistics).

I handed over the leadership of this initiative to a colleague starting in fall 2018, as I was preparing to take on other efforts, such as the Google exploreCSR program mentioned above.

- ***NEXUS Shadowing Program:*** UTEP's College of Engineering has a program in place for high-school students to participate in shadowing programs in the College's research labs (with volunteering faculty members). Since 2010, I have volunteered to participate in this program as research mentor, welcoming high-school students to my research lab. Every summer since then, I have had between 2 (in the early years) and 6 high-school students interning in my lab. The program I run is in fact not exactly a "shadowing" program. Computer science being not too dangerous for students (there are no safety concerns as would be in chemical or other labs), I instead built a summer curriculum for students who come to my lab. This curriculum includes:



  - <u>An introduction to programming and projects:</u> with Scratch, Python, and Java. Students then have to pick (define and pitch) two projects of their own: one project to be implemented in Scratch and one open-ended project with free choice of language. As a result, they also learn how to put a design document together and to present it. They present updates of their work at each weekly research group meeting.
  - <u>Matlab programming:</u> Since I use Matlab for part of my research and I want the interns to fully experience life in a research lab, I have them take a brief introduction to Matlab.
  - <u>Command line and testing for research:</u> It is essential to me that my interns feel valued in my research group and feel instrumental to our research progress. Since a summer may be too short to fully include them in the type of research we do (development of optimization and simulation algorithms for decision under uncertainty), I involve them in testing. My area of research is highly experimental and testing is

crucial to demonstrate effectiveness. We train the interns to use our tools and task them with testing and reporting. This gives them a truthful taste of research. They report their results weekly to the group. Over the summers, interns have taken on very varied projects. One still dear to me is one of the first ones: a robot that could solve a rubics cube. A picture of it is available below. Many students who have been part of NEXUS with me ended up in computing programs at UTEP or elsewhere. Many of the female interns applied to NCWIT AiC as a result and were successful.

Such programs are instrumental in raising awareness and interest of high-school students in computing, and in keeping interest up in young women in high school. However, studies show that it is very important to start as early as possible.

- ***Excites Summer Camps:*** In 2015, I had the opportunity to build material for engineering Excites summer camps held at UTEP and handled by the College of Engineering, for middle-school students. These summer camps were supposed to be a combination of computer science and civil engineering. I partnered in designing the camp with my colleague Ivonne Santiago (from Civil Engineering at UTEP): she designed the civil engineering part and I designed the computer science part. I then held a few of these camps and trained undergraduate students to keep holding them for the rest of the summer. Since then, the computing modules I designed have been used every summer.

  Below is a picture of Excites summer camp students engaged in a robot programming activity: a race we put together to make the activity playful.

Mentoring of and reaching out to K-12 students is essential to ensure that we spark interest in children as soon as possible, that we make them curious. The above efforts are my major efforts in this direction. I also regularly offer daylong visits to schools of the area, with general presentation of computer science and opportunities for the K-12 students visiting our department and hands-on activities in computer science (e.g., robots programming, 3D videogame programming, cryptography). Such visits are an opportunity for me to start mentoring relationships with teachers. This leads me to the next section.

Interaction with and Mentoring of K-12 Teachers

Reaching out to K-12 students and increasing the pipeline of students who choose computing (recruiting them and retaining those who have early aspirations) would not be possible (or nearly as efficient) if it were to ignore the teachers. I value interactions with teachers as we are all on the same boat: all concerned with the success of our students, wanting the best for them. As a result, I have sought interaction with and mentoring of teachers from early in my career and have been consistently involved in school and STEM program boards, in school visits, presentations to teachers, and teacher conferences since 2012.

- ***School and program boards:*** Being a member of a school board is a great opportunity to make a connection between UTEP and the local schools, to share what is important at that time for our university, our programs, and advise the local schools and programs about what they should do or not, connect them to the relevant people, program. Overall, it helps make the gap between institutions (K-12 and higher education) less palpable.
  In addition, networking is very important when you want to effect change: participating in boards allows meeting people who have similar interest in student success and together we are more likely to succeed.
  I have been an active member of school boards of our local community since 2012. I have been a member of the board of several local STEM programs since 2015.

- ***Holding school visits to UTEP / holding sessions at schools:*** The more I can interact with schools, the better. This creates a connection with the teachers and the students. Transition to the university all of a sudden does not feel that daunting for the students and teachers have a connection to ask questions to on behalf of their students. Additionally, this is yet another opportunity to share opportunities for K-12 students on campus.
  I regularly either hold tours of my department or go visit schools and give talks or hold booths about our program at their events (about twice a year).

- ***Presentations to teachers' events or conferences:*** Efforts lead to more opportunities. My efforts reaching out to schools, to students, holding visits and participating in school events, lead to regular invitations to give talks to teachers: at conferences such as the TnTc annual conference (Teachers Networking Technology and Content), at the Annual CISD Professional Development Conference at UTEP, as plenary speaker for teachers retreats (such as at the El Paso High School in January 2018), as invited speaker for the Hour of Code event of El Paso ISD in December 2017.
  Such interventions allow me to further forge connections with teachers and school, hence allowing us to work more effectively together towards student success.

  Here is a tweet about my invited talk at EPISD for their 2017 Hour of Code event.

Recently, the Ysleta School District Director of Innovative Learning invited me to be part of a grant proposal that the Ysleta School District is putting together, to participate in it as a mentor to the project's summer camps for middle-school female students.

3. Curriculum Vitae

Most of the information available in my CV, starting on the next page, is about my work since 2012. Highlighted in blue are all activities (publications, service, professional development) related to undergraduate students' success (including outreach to younger students – K-12 – and to their teachers). In particular, only publications are provided that involve at least one undergraduate student: they are highlighted in blue and the undergraduate student's name is underlined. A more complete version of my CV with publications not only involving undergraduate students is available on my website.

Martine Ceberio

Associate Professor of Computer Science
Computer Science Department
University of Texas at El Paso
500 West University Avenue
El Paso, Texas 79968-0518

Phone: (915) 747 5480
email: mceberio@utep.edu
http://www.cs.utep.edu/mceberio/

Education

Ph.D. May 2003, *Department of* **Computer Science**, *University of Nantes, France*
Dissertation title: "Contributions to numerical under and over-constrained CSPs: Symbolic Tools and Flexible Constraints"

D.E.A. 1999, *Department of* **Computer Science**, *University of Nantes, France*
(D.E.A.: Diplôme d'Études Approfondies)

M.S. 1997, *Department of* **Mathematics**, *University of Nantes, France*

B.S. 1995, *Department of* **Mathematics**, *University of Poitiers, France*

Employment

June 2018
Faculty in Residence, *Google, Mountain View, CA*

September 2012 – Present
Associate Professor, *Computer Science Department, The University of Texas at El Paso*

August 2004 – August 2012
Assistant Professor, *Computer Science Department, The University of Texas at El Paso*

August 2003 – August 2004
Visiting Assistant Professor, *Computer Science Dept., The University of Texas at El Paso*

September 1999 – May 2003
Instructor and Research Assistant, *Computer Science Dept., University of Nantes, France*

Honors and Awards since 2012

- UTEP EDGE Faculty Fellow, January 2018 – Present
  The UTEP EDGE program is grounded on recognizing that students enter UTEP with many talents, great strengths, and big dreams. The UTEP Edge develops these assets through

a variety of high-impact experiences made possible by the expertise and dedication of our faculty, staff, alumni, and community partners.

- Invited Plenary Speaker at the 17th International Symposium on Scientific Computing, Computer Arithmetics and Verified Numerics, September 2016
- Faculty Co-author of Outstanding Paper Award. Joint Annual Conference of the North American Fuzzy Information processing Society NAFIPS'2015 and 5th World Conference on Soft Computing, August 2015
- National Science Foundation CAREER Award recipient, 2009-2016, $564,000.

Selected Publications since 2012 – with undergraduate student co-authors, or related to education

☐ Selected Chapters in Scholarly Books and Monographs

1 article co-authored with an undergraduate student, out of 17 chapter articles

Ch1 Paden Portillo, Martine Ceberio, and Vladik Kreinovich, "Towards an Efficient Bisection of Ellipsoids", In: Martine Ceberio and Vladik Kreinovich (eds.), **Constraint Programming and Decision Making**, Springer Verlag, Berlin, Heidelberg, 2014, pp. 137-142.

☐ Selected refereed Journal Articles, published or accepted in Final Form

7 co-authored with undergraduate students out of 13 refereed journal articles

J7 Leobardo Valera, Angel Garcia, Jesus Padilla, and Martine Ceberio, "Towards Predicting the Behavior of Large Dynamic Systems, using Reduced-Order Modeling and Interval Computations". Submitted to the Journal of Granular Computing, December 2017.

J6 Anthony Welte, Luc Jaulin, Martine Ceberio, and Vladik Kreinovich, "Avoiding Fake Boundaries in Set Interval Computing", **Journal of Uncertain Systems**, 2017, Vol. 11, No. 2, pp. 137-148.

J5 Anthony Welte, Luc Jaulin, Martine Ceberio, and Vladik Kreinovich, "Computability of the Avoidance Set and of the Set-Valued Identification Problem", **Journal of Uncertain Systems**, 2017, Vol. 11, No. 2, pp. 129-136.

J4 Vladik Kreinovich, Martine Ceberio, and Quentin Brefort, "In category of sets and relations, it is possible to describe functions in purely category terms", **Eurasian Mathematical Journal**, 2015, Vol. 6, No. 2, pp. 90-94.

J3 Quentin Brefort, Luc Jaulin, Martine Ceberio, and Vladik Kreinovich, "Towards Fast and Reliable Localization of an Underwater Object: An Interval Approach", **Journal of Uncertain Systems**, 2015, Vol. 9, No. 2, pp. 95-102.

J2 Xiaojing Wang, Martine Ceberio, Shamsnaz Virani, Angel Garcia, and Jeremy Cummins. "A Hybrid Algorithm to Extract Fuzzy Measures for Software Quality Assessment". **Journal of Uncertain Systems**, 2013.

J1 Vladik Kreinovich, Christelle Jacob, Didier Dubois, Janette Cardoso, Martine Ceberio (2012). "Failure Analysis of a Complex System Based on Partial Information about Subsystems, with Potential Applications to Aircraft Maintenance". **Journal of Applied and Computational Mathematics**, 11(2), 165-179.

☐ **Selected Refereed Conference Proceedings (peer reviewed)**

8 co-authored with undergraduate students out of 21 refereed conference proceedings articles

C8 Leobardo Valera, Angel Garcia, Jesus Padilla, Martine Ceberio, and Luis Bravo, "Handling Uncertainty in the Finite Element Method Using Interval Constraint Solving Techniques". To be published in the proceedings of the European Safety and Reliability Conference (ESREL 2018), Norway, June 2018.

C7 Anthony Welte, Luc Jaulin, Martine Ceberio, and Vladik Kreinovich, "Robust Data Processing in the Presence of Uncertainty and Outliers: Case of Localization Problems", In the Proceedings of the **IEEE Series of Symposia in Computational Intelligence SSCI'2016**, Athens, Greece, December 6-9, 2016.

C6 Esquinca, A., Villa, E. Y., Hampton, E. M., Ceberio, M. C., Wandermurem, L. S., (2015). "Latinas' resilience and persistence in computer science and engineering: Preliminary findings of a qualitative study examining identity and agency." Proceedings of the 2015 **Frontiers in Education**.

C5 Quentin Brefort, Luc Jaulin, Martine Ceberio, and Vladik Kreinovich, "If We Take Into Account that Constraints Are Soft, Then Processing Constraints Becomes Algorithmically Solvable", Proceedings of the **IEEE Symposium on Computational Intelligence for Engineering Solutions** CIES'2014, Orlando, Florida, December 9-12, 2014, pp. 1-10.

C4 Brefort, Q., Jaulin, L., Ceberio, M. C., Kreinovich, V. Y., (2014). "If We Take Into Account that Constraints Are Soft, Then Processing Constraints Becomes Algorithmically Solvable". (pp. 1-10). Proceedings of the **IEEE Symposium on Computational Intelligence for Engineering Solutions**, Orlando, Florida SSCI'2014, December 9-12, 2014.

C3 X. Wang, M. Ceberio, S. Virani, C. Del Hoyo, and L. Gutierrez. "Fuzzy measure extraction for software quality assessment as a multi-criteria decision-making problem". Proceedings of the **2012 International Conference on Software Engineering Research and Practice**, Las Vegas, NV, July 2012.

C2 X. Wang, A. F. Garcia Contreras, M. Ceberio, C. Del Hoyo, L. C. Gutierrez, and S. Virani. "Interval-based algorithms to extract fuzzy measures for software quality assessment". Proceedings of the **Annual Conference of North American Fuzzy Information Processing Society (NAFIPS'2012)**, Berkeley, CA, August 2012.

C1 Xiaojing Wang, Angel Garcia Contreras, Martine Ceberio, Christian Del Hoyo, Luis Gutierrez, "A Speculative Algorithm to Extract Fuzzy Measures from Sample Data", Proceedings of the **2012 annual international conference of Fuzz-IEEE (Fuzz-IEEE'12)**.

☐ **Selected Refereed Workshop Proceedings (peer reviewed)**

1 co-authored with undergraduate students out of 2 refereed workshop proceedings articles

W1 Luis Gutierrez, Martine Ceberio, Vladik Kreinovich, Rebekah L. Gruver, Marianna Pena, Matthew J. Rister, Abraham Saldana, John Vasquez, Janelle Ybarra, and Salem

Benferhat, "From Interval-Valued Probabilities to Interval-Valued Possibilities: Case Studies of Interval Computation under Constraints", Proceedings of the **6th International Workshop on Reliable Engineering Computing REC'2014**, Chicago, Illinois, May 25-28, 2014.

☐ **Selected Conference / Workshop Abstracts (peer-reviewed)**

1 co-authored with undergraduate students out of 27 refereed conference/workshop abstracts

A1 Martine Ceberio, Miguel Argaez, Luis Gutierrez, Leobardo Valera. "Using Interval Constraint Solving Techniques to Solve Dynamical Systems". **CORS/INFORMS 2015 Meeting**, Montreal, June 2015.

☐ **Edited Research Books**

B2 Martine Ceberio and Vladik Kreinovich (eds.), Constraint Programming and Decision Making: Theory and Applications, Springer Verlag, Berlin, Heidelberg, 2017.

B1 Martine Ceberio and Vladik Kreinovich (eds.), Constraint Programming and Decision Making, Springer Verlag, Berlin, Heidelberg, 2014.

☐ **Contributed Presentations Related to Education**

Ed10 Panelist at the CS4All Knowledge Forum. Panel on "Transition from Post Secondary to Industry: What do Students Need in Order to Make this Transition Successfully?". September 12-13, 2018.

Ed9 "Computer Science Opportunities for Middle and High-School Students" at the 2nd Annual Canutillo ISD Professional Development Conference, GRIT (Growth, Resilience, Innovation, Tenacity). The University of Texas at El Paso, August 15-17, 2018.

Ed8 "Computational Thinking in the Classroom" at the 2nd Annual Canutillo ISD Professional Development Conference, GRIT (Growth, Resilience, Innovation, Tenacity). The University of Texas at El Paso, August 15-17, 2018.

Ed7 "Innovative Teaching - Bilingualism and Learning Across the Disciplines", a UTEP EDGE and Center for Faculty Leadership and Development workshop, with co-presenters E. Mein and A. Esquinca – February 2018.

Ed6 Presentation to El Paso High School teachers at El Paso High School, about Computational Thinking in the Classroom, across Disciplines – Feb. 2018. Audience $\approx$ 100.

Ed5 Presentation at the EPISD Hour of Code event for Teachers – EPISD, Dec. 2017. Audience $\approx$ 60.

Ed4 Contributed presentation at the Teacher Networking Technology Conference in November 2015 in El Paso about "Computational Thinking in the Classroom". Audience: about 35 teachers from all disciplines, from K-12.

Ed3 Contributed presentation at the Teacher Networking Technology Conference in October 2014 in El Paso about "Coding your way through school". Audience: about 50 teachers from all disciplines, from K-12.

Ed2 Presentation to the Clint Independent School District about Computer Science, May 2014

Ed1 Invited speaker for a Webinar for all teachers of Ysleta School District about how they can bring computer science in their classroom and what they can do it they are CS / math teachers, March 2014

Grants and Contracts since 2012

☐ **Total Grants and Contracts since 2012**

Since 2012: Total is $4,085,714 (**$1,419,905** as PI).

- **Federal funding: $385,905** as PI and $5,517,552 for projects in which I am co-PI
- **Army funding: $999,000** as PI and $190,000 for projets in which I am co-PI
- **Industry funding: $41,000** as PI and $35,000 as co-PI.
- **University funds: $25,000** from URI and two IDRs (IDR1 & IDR2)

☐ **Federal grants related to education, since 2012**

1. **NSF IUSE/PFE RED – Co-PI** *IUSE/PFE:RED: Toward a Model of Change for Preparing a New Generation for Professional Practice in Computer Science.* July 2016 – June 2021. Amount: $1,919,849.
2. **NSF Research on Gender – Co-PI** *Latinas in Computer Science and Engineering: A Qualitative.* September 15, 2012 – September 14, 2016 extended. Amount: $524,960.

☐ **Other grants related to education, since 2012**

1. **Google – PI** *Google exploreCSR: Building Pathways to Graduate School* Received in Summer 2018, Project for Fall 2018 and Spring 2019. Amount: $35,000.
2. **STEM Accelerator Fund:** *CS1 Course Redesign (cont'd).* Amount: $3,500, Spring 2018.
3. **STEM Accelerator Fund:** *CS1 Course Redesign.* Amount: $8,500, Fall 2016 & Spring 2017.
4. **Google CS Engagement Award:** Ceberio, Martine (PI). *Revamping CS1 to increase retention.* Amount: $5,000. (January 2015 – December 2015).

Service / Outreach since 2012

Selected Professional Service since 2012

- President of **NAFIPS**, January 2019 – December 2020 (NAFIPS is the North American Fuzzy Information Processing Society)
- **Editor in Chief** of the journal **Reliable Computing**
- Member of **IEEE Technical Committee on Soft Computing**, since March 2016: Awarded 2018 "IEEE Most Active SMC Technical Committee Award" at the SMC'2018 Award Banquet. This is the third time this TC received this award.

- **Conference organizer and chair or co-chair of program committees** for a number of conferences, including: NAFIPS (2012, 2014, 2016), CoProD workshop series since 2008 (http://coprod.constraintsolving.com), ACM SAC (Symposium on Applied Computing) Knowledge Representation and Reasoning (KRR) 2016, 2017, 2018.
- **Member of Program Committees**, including: RCRA 2017 (Rappresentazione della Conoscenza e Ragionamento Automatico), IAE/AIE (International Conference on Industrial Engineering, Other Applications of Applied Intelligent Systems), FLAIRS (the Florida AI Research Society), IJCAI (International Joint Conference in Artificial Intelligence)
- **Reviewer for a number of conferences, journals, and grant proposals**, including: [**conferences and workshops**] workshops at CP, ECAI (the European Conference on Artificial Intelligence), FIE (the Frontiers In Education conference), Mexican International Conference on Artificial Intelligence (MICAI), Workshop on Engineering Applications (WEA), ICTCS 2014 (the Italian Conference on Theoretical Computer Science), AI*IA 2016 (the XV International Conference of the Italian Association for Artificial Intelligence); [**journals**] Reliable Computing, INFORMS Journal of Computing, Information Sciences, Journal of Experimental and Theoretical Artificial Intelligence, the Annals of Mathematics and Artificial Intelligence, Artificial Intelligence, Journal of Logical and Algebraic Methods in Programming, Transactions on Mathematical Software, AAAS-Science; [**proposals**] NSF panels in CISE (2012, 13, 14, 15, 17, 18, 19)
- Reviewer for: Grace Hopper Conference's 2014 scholarship applications, DoD 2014 Star Award, NCWIT Collegiate Award 2015, 2016, 2019, NCWIT AiC 2011-2019, NCWIT AiC Educator Award 2015, 2019.
- **Students External Advisor and/or Reviewer**
  - External reviewer of a PhD dissertation for the Computer Science program at the University of Paris 6, France, 2017.
  - Co-Supervisor of 2 graduate student from ENSTA France (advisor: Luc Jaulin), interning in the TRACS lab at UTEP for five months from April 2014 to August 2014, and for three months in summer 2016 (TRACS is the lab on Theoretical Research driven by Applications in CS, which includes my research group CR2G: cr2g.constraintsolving.com)
  - External reviewer of a PhD dissertation for the Executive Board of the Italian Association for Logic Programming (GULP), 2012.

☐ **Professional Societies Membership**

- ACM (Association for Computing Machinery), ACM-W (ACM's committee on Women), INFORMS, IEEE, AAAS (American Association for the Advancement of Science),

Institutional Service since 2012

☐ **Department Committees**

- Current assignments
  - **Academic advisor to undergraduate students** – about 50+ per semester

- Member of the **CS Undergraduate Curriculum Committee** – August 2013 – **present**.
- **Chair** of the Undergraduate Fundamentals course sequence Committee – May 2015 – **present**.
- Member of the **Faculty Evaluation Committee** – February 2015 – **present**.
- In charge of the **Computer Science Department's course schedule** – 2007 to Spring 2010, August 2012 to January 2017, and **Fall 2018, Spring 2019**.
- **CS Faculty Search committee** – Fall 2018 – Spring 2019.
- Member of the **Computer Science Advancement of Women in Computing** committee – August 2015 – **present**

• Previous assignments

- Founder and **advisor of the ACM-W chapter at UTEP** – June. 2012 to May 2018
  As the ACM-W advisor, I have guided and supervised the ACM-W students in the following projects that were funded by NCWIT or Google:
  * NCWIT Seed Fund: in spring 2014 to develop a still existing mentoring program for CS UG students
  * Google IgniteCS program: in spring 2016, ACM-W was awarded its first Google IgniteCS project to put in place formal and informal program to teach computer science to young students at a middle school of El Paso. In spring 2017, they received their second award for a similar program at a high-school of El Paso.
  * In fall 2017, they are very active in helping with the NCWIT Aspirations in Computing program and they are working on developing and submitting a Google First project.
- Chair of the Programming Languages course Committee – August 2013 – May 2015.
- **Webmaster** of the Computer Science website – August 2011 – August 2014.
- Part of the **CS ABET preparation Committee** – September 2012 – Fall 2013
- **Chair of the 2013 CS Faculty Search Committee** – August 2012 May 2013
- **CS Faculty Search Committee** – December 2011 – Spring 2012

☐ **College Committees**

- Presenter and mentor at a University-wide (led by the College of Engineering) workshop for junior faculty on the NSF CAREER grant program – February 2018
- Member of the **Task force on Faculty Success**. March 2013 – August 2017.
- Member of a team part of the **NCWIT Extension Services** (along with Ann Gates, Miguel Velez-Reyes, Pat Nava, Gabby Gandara) who worked on **increasing the number of female students in Computing**. Fall 2012 – Summer 2014.
- Member of the **Facilitation Team For Information and Security**. September 2011 – November 2012.

☐ **University Committees**

• Current Assignments

- Member of **COURI's Board of Advisors**: COURI is the Campus Office for Undergraduate Research Initiatives at UTEP – March 2015 – **present**.
- Member of **UTEP's Mama PhD** group – September 2010 – **present**.

- Previous Assignments
  - Member of the **Executive Council of the Faculty Senate** as representative of UTEP's College of Engineering – September 2015 – August 2017
  - **Vice-President of the Faculty Senate**. September 2014 – August 2015.
  - Member of the **Executive Council of the Faculty Senate** – as **Secretary** (September 2012 – August 2014) as such:
    * Representative of this council on the IT standing committee of the Faculty Senate (2013-2014)
    * Representative of this council on the UGCC and Student Grievance Committee standing committee of the Faculty Senate (2014-2015)
  - Member of the **Executive committee of the Computational Sciences Program** – September 2008 – June 2015.
  - Member of the **Board of the Women's Resource Center** (now Student Resource Center) – September 2011 – August 2014.
  - Member of **UTEP's Undergraduate Curriculum Committee** (standing committee of the Faculty Senate) – September 2011 – August 2014
  - **Member of the Computational Sciences Faculty Search**. September 2013 – April 2014.
  - **Faculty Senate** member. September 2010 – August 2012.
  - **Women's Advisory Council to the President**. Chair: Sept. 2010 – December 2012; Member: Sept. 2010 – Sept. 2010; Past chair: Jan. 2013 – Dec. 2013; Sept. 2018 – present.

Local / State Outreach since 2012

Note: All of the activities listed under "Local and State Outreach" are relevant to education. Moreover, most of the these activities contribute to my goal of increasing the participation of women in computing fields.

- **Advisory Boards' membership:** Member of the board of advisors of 6 schools of El Paso over the last 7 years, currently 5 (indicated with an asterix *). Bel-Air's T-STEM Academy*, Parkland's T-STEM Academy*, Harmony Science Academy of El Paso*, Eastlake High School CSE program*, El Paso High School STEM program*, Saint Patrick's Elementary and Middle School
- **Faculty advisor** for summer research projects for high-school students (2010 through 2017). Nexus program at UTEP: summer internship for high-school students in my research lab. *Notably: an unprecedented high-number of interns participated in summers 2014 and 2016: 7 female high-school students)*
- **NCWIT Aspirations in Computing Regional Affiliate Competition Coordinator**
  - Coordinator of the El Paso affiliate, fall 2015 to spring 2018

- Coordinator of the El Paso/Las Cruces affiliate from 2011 to spring 2014
  15 schools of El Paso/Las Cruces and the wider area have participated in the competition, and over 50 young women have been honored.

- **Presentations about computer science** I regularly give presentations about computer science, at UTEP or at various schools of the El Paso area. In particular, in fall 2016 and fall 2017, I gave talks to high-school young women every day of our e-Week, reaching out to about 100 women in one week. In addition, some of my past talks include the following:

  - **Presentations to K-12 Students**
    * Presentation to El Paso High School students visiting UTEP – 2018. Audience ≈ 80.
    * Presentation to the Girls-Who-Code group from Harmony Science Middle School of El Paso – UTEP, 2017. Audience: 8.
    * Presentation to an all-girls summer camp at Fab Lab El Paso – 2016
    * Presentation at Harmony Science Academy of El Paso about computer science and careers (2014)
    * Invited speaker at the New Mexico Celebration of Women in Computing, Las Cruces, NM (2012).
  - **Presentations to College Students**
    * Guest speaker at the Annual banquet of UTEP's SWE student chapter in 2018
    * Talk about being a professor in computer science, to doctorate students, 2015

- **Career Fairs/Days presenter:**

  - Harmony Science Academy: promoting Computer Science, (2016)
  - Girls Powered Event presenter at Eastwood High School in El Paso (2016)
  - Ibero Academy: Presentation about Computer Science to Kindergartdeners, 1st graders, and 2nd graders (2014)
  - Loretto Academy of El Paso – all-girls middle and high school (2011, 2012, 2014)

- **UTEP tours and open house events**

  - Hosted a day of Computer Science for Bel-Air High School in 2017 (about 50 students)
  - Hosted a day of Computer Science and Engineering for Saint Patrick's Elementary School – 3rd to 5th grade – 2017 (about 60 students)
  - Hosted a day of Computer Science and Engineering for Saint Patrick's Middle School, 2016 (about 80 students)
  - Hosted a day of Computer Science for Bel-Air High School, 2016 (about 50 students)
  - Participates in UTEP's Orange and black Days, and other events such as Open houses annually
  - Regularly prepares presentation material and train my research team students to give overviews of CS to visiting students.

- **High-school classroom innovation:** Computer Science and Language Learning, Loretto Academy of El Paso (Fall 2013).

- **Judge:**

- Science Fair judge at Harmony Science Academy Middle School, 2017.
- Science Fair judge at St Patrick's Elementary and Middle School, 2016 and 2017.
- Chapin High-School **Senior Project Symposium** (2011).

- **Other**
  - Hour of Code at St. Patrick's Elementary and Middle School (2016)
  - Mentornet mentor in 2012 and 2013
  - Faculty advisor: Harmony Science Academy Alumni Association at UTEP since 2015

☐ Professional Development

The following are meetings I have attended in recent years and that contributed to my professional development (these include either training or informative meetings about grant programs).

- Industry Programs: Faculty in Residence (FIR) at Google, Summer 2018; FIR Alumni meeting, Google, 2019; FIR Mentor, Google, 2019.
- Workshops: Rethinking Engineering Education at Hispanic Institutions, 2018, UTEP; Reporting and Using your Data for Curricular Improvement, organized by Dr. Toni Blum at UTEP, 2017; Large Class Seminar on Peer-Led Tutoring, 2017, UTEP; Teaching and Motivation, by Olin College instructors, organized by UTEP STEM-Accelerator Project team, 2016; Problem-Based Learning, 2013, 2014; Problem-Based Learning workshops: 2012, 2013, 2014; Affinity Research Group training: 2011 2012; CRA-W Career Mentoring Workshop: 2016.
  - Participation at the Olin workshop led to my application to funds to help me redesign CS1. I did that in spring 2017.
- Conferences: NCWIT Summer: May 2013; Participated in the "Networking Technology & Content Conference", El Paso, TX. (Nov. 2014 and 2015) ; CE21 Community Meeting, 2014; Gender Summit: November 2013; NSF CE21 Community Meeting: 2011, 2012, 2014; CRA Career: Washington DC, 2012; CRA-W: Atlanta, 2012.
- Continuing Education Program, Flipped Learning Brown Bag discussion, Center for Research in Engineering and Technology Education (CREaTE), UTEP, 2014.
- Program: Leadership Development Institute at UTEP: 2012-2013