

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.

Logistics: =====

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

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

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

UTEP CS1301 Ceberio Spring 2017

Communication platform: 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/spring2017/cs1301/home>

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. For those who wish 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.

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, weekly quizzes, class participation, 3 mid-term exams, student engagement, and a final exam.

The approximate percentages are as follows:

- 15% Homework
- 20% Quizzes
- 50% Exams (3 mid-term exams and 1 final exam)

- 8% Student Engagement in Computer Science
- 7% 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: 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 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 usually will usually be on-line quizzes on socrative.com. There will be no make-up on missed 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.

Exams: There will be 3 midterm exams and one final exam. All four exams together will weigh 50% 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 5, week 10, 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.

Student Engagement in Computer Science: During the course of the semester, you must engage as a computer scientist in activities as shown below, in a way that you cumulate at least 8 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):

- **2 points** for each of the following:
 - Write a summary of a seminar you attended (proof of attendance needs to be provided as well);
 - Attend 6 workshops provided by undergraduate TAs;
 - Participate in a research project as a research human subject;
 - Participate in a Department's open house as a volunteer student;
- **4 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;
- **6 points** for each of the following:
 - Be an active participant in Google IgniteCS program (or equivalent);
 - Be an active undergraduate researcher in one of the Computer Science Research labs.

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.

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 main social, ethical, and legal issues
3. Computing as a profession, from required knowledge and skills to major career options
4. Computer representation of simple data types and operations, including operations with binary numbers
5. Technical aspects of computing, including memory, operating systems, editors, interpreters, compilers, debuggers, and virtual machine
6. Differences among programming languages
7. The purpose and use of exceptions

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. 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 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