The University of Texas at El Paso
Department of Computer Science
CS 3331 – Advanced Object-Oriented Programming
Summer 2020 Syllabus

1. General Information

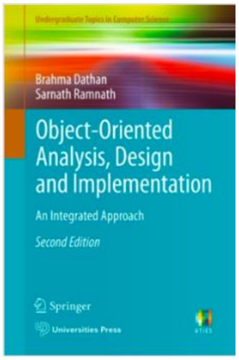
Instructor:
Daniel Mejia
Email: dmmejia2@utep.edu
Dates: June 8 – July 31, 2020
Meeting location: Blackboard Collaborative Ultra
Office: Microsoft Teams
Meeting Times: MTWRF – 10:30-11:35am
Office Hours: Monday & Wednesday 3-4pm, or by appointment (MS Teams)
CRN: 34193

Teaching Assistant (TA):
Khandoker Rahad
karahad@miners.utep.edu
Office Hours: MW 12-2pm
Blackboard Collaborate Ultra (TA Office Hours)

Prerequisites:
CS 2302 with a C or better

Textbook (Required):

1. *Object-Oriented Analysis, Design and Implementation: An Integrated Approach*. Brahma Dathan, Sarnath Ramnath. Springer, Universities Press, 2015.


Undergraduate Topics in Computer Science
Brahma Dathan
Sarnath Ramnath
Object-Oriented Analysis, Design and Implementation
An Integrated Approach
Second Edition
Springer
Universities Press

Free e-book available through the publisher (https://link.springer.com/book/10.1007/978-3-319-24280-4)

Recommended Books (Not required):

1. *Head First Design Patterns*. Eric Freeman and Elizabeth Freeman. O'Reilly 2004.

2. *Head First Object-Oriented Analysis and Design*. Brett D. McLaughlin, Gary Pollice, and Dave West. O'Reilly 2006.

3. *The Elements of Java Style*. Allan Vermeulen, et al. Cambridge University Press, 2000.

4. Martina Seidl, et al., *UML@Classroom: An Introduction to Object-Oriented Modeling,* Springer, 2015 ([e-book] through UTEP library)

5. Cay S. Horstmann, *Core Java Volume I - Fundamentals,* 11th edition, Prentice Hall, 2018 ([e-book] through UTEP library)

2. Objectives & Outcomes

Course Description
CS 3331 – Advanced Object-Oriented Programming. An in-depth exposure to the object-oriented programming paradigm, which builds upon programming experience gained in lower-level computer science classes. Emphasis on programming in an object-oriented language with which students are already familiar, and on requirements, testing, code reading, and comprehension.

The class presents a look into the growing capabilities of object-oriented programming as a way to model the real world. Additionally, this course will provide the fundamental knowledge to software modeling through the use of best practices. This course teaches students to approach program development using a systematic approach that is used in industry.

Learning Outcomes
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. Upon successful completion of this course, students will able to:

a) Explain the differences between an object-oriented approach and a procedural approach.

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:

a) Formulate use-case diagrams and scenarios to support understanding of user requirements.
b) Use object-oriented design notations, including UML class diagrams and state machine diagrams (optionally sequence diagrams) to model problem solutions.
c) Use basic object-oriented design patterns to structure solutions to software design problems.
d) Translate design features, such as classes and relationships, to implementations.
e) Use frameworks and library classes and methods, such as collections, GUI, multithreading, and networking, in problem solutions.

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:

a) Design and implement software employing the principles of modularity, encapsulation, information hiding, abstraction, and polymorphism.

b) Design, implement, and use classes and methods that follow conventions and styles, and make appropriate use of advanced features such as inheritance, exception handling, and generics.
c) Evaluate existing classes and software for the purposes of extension through inheritance.
d) Create API documents for classes, fields and methods.
e) Design and implement test suites for automated unit testing.
f) Re-factor existing source code to improve its design or efficiency.

3. Policies & Other Information

Grading:

- Quizzes/Homework – 15%
- Exam 1 – 15%
- Exam 2 – 15%
- Final Exam – 25%
- Programming Labs – 30%

The nominal percentage-score-to-letter-grade conversion 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

Additionally, any one of the following will result on a final grade of F, even if the overall average is greater than 60%.

- Earning an average of less than 60% on the programming lab assignments
- Earning a grade of less than 60% on the final exam
- Earning an average of less than 60% on Exam 1 & Exam 2
- Not submitting ALL lab projects by the end of the semester, even if they are too late to receive credit (lab projects should still be functional and will be tested to ensure functionality).

The instructor reserves the right to adjust these criteria downward, e.g., so that 88% or higher represents an A, based on overall class performance. The criteria will not be adjusted upward, however.

Class Assignments:
Reading, non-programming homework, quizzes, and programming lab assignments will be posted on Blackboard. It is your responsibility to check Blackboard for all assignments. All work must be done individually. While you may discuss the problem in general terms with other people, your answers and your code should be written and tested by you alone. If you need help, consult the TA or the instructor. All assignment submissions must be newly attempted, never before used original work.

Homework Assignments:

Reading and non-programming homework assignments are due at 11:59pm on the due date specified, unless otherwise indicated; after this a late penalty will be assessed (-10%). Many assignments will be required to be done on paper or through the use of an electronic tablet (white background). All reading and non-programming homework assignments must be handwritten in either pencil, black/blue pen (including electronic tablet), otherwise it will not be graded and will receive a grade of zero. The assignment must be clearly legible for credit. Students should submit a scanned PDF document (or PDF document created by tablet), using a Scanning App on a Smart Phone; if this is not possible, please contact the the instructor for alternative arrangements.

Programming Assignments:
Programming lab assignments are due at 11:59 pm. (Mountain Time) through Blackboard on the specified due date, unless otherwise specified. Late programming lab assignments will be accepted up to 24 hours after the due date/time for up to 70% credit. Accompanying lab reports must be turned in with the source code, should be typed and submitted as a PDF. Unless for unusual circumstances and at the discretion of the instructor, programming lab assignments will not be accepted for credit after 24 hours past the due date/time. All programming assignments are subject to a demo session with the TA or instructor. All programming lab assignments must be submitted prior to the end of the semester to receive a passing grade for the course, even if it is too late to receive credit. You should expect to spend at least 10-15 hours/week outside of class on reading and homework.

Exams:
There will be two (2) exams and one (1) final exam. Exams will be posted and submitted through Blackboard with appropriate due dates listed. The purpose of the exams is to allow you to demonstrate mastery of course concepts. Make-up exams will be given only in extremely unusual circumstances, and at the discretion of the instructor.

Quizzes:
The purpose of a quiz is to ensure that you have read the weekly reading assignment and to verify that you have mastered the major concepts of the course. Quizzes typically will be about 5-10 minutes in length. Missed quizzes cannot be made up.s

Attendance:
Students are expected to be participate in the course through the completion of all assignments, quizzes, and exams. Additionally, students should watch all posted video lectures that are provided by the instructor.

Technology:
Course content is delivered via the Internet through the Blackboard learning management system (LMS), supplemented by Microsoft. Teams. Ensure your UTEP MINERS account is working and that you have access to the Internet. You may use any of the primary Web browsers—Edge, Google Chrome, Firefox, Safari, etc. When having technical difficulties, try switching to another browser.

The use of laptops, cell phones, or tablets of any kind, will be necessary for this course. It is necessary for all students to have an Integrated Development Environment (IDE) for program development (e.g. Eclipse, IntelliJ, etc.) with the capability of running .java files. It may be necessary to have a cell phone with a PDF Scanning App (Adobe Scanner, Notes (iPhone), CamScanner, etc.) to scan homework assignments. You may use a tablet (iPad, Surface Pro, etc.) to handwrite certain homework assignments and submit as PDF documents.

You will need to have access to a computer/laptop, printer, scanner, a webcam, and a microphone. You will need to download or update the following software: Microsoft Office, Adobe, Flash player, Windows Media Player, QuickTime, and Java. Check that your computer hardware and software are up-to-date and able to access all parts of the course. If you encounter technical difficulties of any kind, contact the Help Desk.

Drop Policy:
To drop this class, please contact the Registrar's Office to initiate the drop process. If you cannot complete this course for whatever reason, please contact me. If you do not, you are at risk of receiving an "F" for the course.

Accommodations Policy:
UTEP is committed to providing reasonable accommodations and auxiliary services to students, staff, faculty, job applicants, applicants for admissions, and other beneficiaries of University programs, services and activities with documented disabilities in order to provide them with equal opportunities to participate in programs, services, and activities in compliance with sections 503 and 504 of the Rehabilitation Act of 1973, as amended, and the Americans with Disabilities Act (ADA) of 1990 and the Americans with Disabilities Act Amendments Act (ADAAA) of 2008. Reasonable accommodations will be made unless it is determined that doing so would cause undue hardship on the University. Students requesting an accommodation based on a disability must register with the UTEP Center for Accommodations and Support Services (CASS); please contact the office at (915) 747-5148, or by email to cass@utep.edu. Students are required to discuss their accommodations with the instructor for a proper plan to be made.

4. Standards of Conduct and Academic Dishonesty

Netiquette:
Always consider audience. Remember that members of the class and the instructor will be reading any postings. Respect and courtesy must be provided to classmates and to instructor at all times. No harassment or inappropriate postings will be tolerated. When reacting to someone else's message, address the ideas, not the person. Post only what anyone would comfortably state in a F2F situation. Blackboard is not a public internet venue; all postings to it should be considered private and confidential. Whatever is posted on in these online spaces is intended for classmates and professor only. Please do not copy documents and paste them to a publicly accessible website, blog, or other space. If students wish to do so, they have the ethical obligation to first request the permission of the writer(s).

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

A fundamental principle for any educational institution, academic integrity is highly valued and seriously regarded at The University of Texas at El Paso. More specifically, students are expected to maintain absolute integrity and a high standard of individual honor in scholastic work undertaken at the University. At a minimum, you should complete any assignments, exams, and other scholastic endeavors with the utmost honesty, which requires you to:

- Acknowledge the contributions of other sources to your scholastic efforts;
- Complete your assignments independently unless expressly authorized to seek or obtain assistance in preparing them;
- Follow instructions for assignments and exams, and observe the standards of your academic discipline; and
- Avoid engaging in any form of academic dishonesty on behalf of yourself or another student.

Graded work, e.g., homework and tests, is to be completed independently and should be unmistakably your own work (or, in the case of group work, your team's work), although you may discuss your project with other students in a general way. You may not represent as your own work material that is transcribed or copied from another person, book, or any other source, e.g., a web page.

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

- **Cheating**
    - Copying from the test paper of another student
    - Communicating with another student during a test
    - Giving or seeking aid from another student during a test
    - 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**
    - 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
- **Collusion**
    - Unauthorized collaboration with another person in preparing academic assignments

Collaboration:
Collaboration among students is strongly encouraged.
It is acceptable to:

- Talk with other students about approaches and ideas.
- Get ideas and extra information from the internet, books, etc.

However, it is not acceptable to:

- Share code with another student (if a piece of code is submitted by two or more students, both students are guilty of cheating, regardless of who wrote the original code).
- Use code acquired from an outside source (the internet, a friend, etc.)

- Look at another student’s code
- Debug another student’s code

Software to detect plagiarized programs are used; appropriate disciplinary actions will be taken as necessary.

A full description of the University Standards of Conduct and Academic Dishonesty can be found in the Handbook of Operating Procedures.

Professors are required to -- and will -- report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.