**CSE 1325:** Object-Oriented Programming in C++

Summer 2017

**Class Instructor**: Dr.Carter Tiernan

**Office Number:** 620 Nedderman Hall

**Office Telephone Number:** 817 27**2 0113**

**Email Address:** [tiernan@uta.edu](mailto:tiernan@uta.edu)

**Website:** [www.uta.edu/faculty/tiernan](http://www.uta.edu/faculty/tiernan) then select the link for this class. Assignments for the class may also be posted on Blackboard.

**Office Hours:** Mon 2:30 – 5:00pm, Tues and Thurs 2:00 – 3:00pm and by appt. Subject to change on occasion. If you cannot meet with Dr. T during office hours, please contact her to set an appt. for another time.

**Section Information: CSE1325-**001

**Time of Class Meetings:** TR 3:30pm – 5:20pm, June 5, 2017 – Aug 10, 2017, Final on Tues, Aug 15.

**Place of Class Meetings:** NH 109

**Course Description:** Object-oriented concepts, basic Unified Modeling Language (UML) modeling, collection classes, generics, polymorphism, reusability, and introduction to design patterns. Projects involve extensive programming and include graphical user interfaces and multithreading.

**Description of Course Content:**

Object-oriented programming is one of the most important programming paradigms in modern software development. C++ is a commonly used object-oriented programming languages. The objective of this course is to introduce the fundamental concepts, principles, methods, and techniques of object-oriented programming using C++.

The topics covered in this course can be largely divided into two parts. The first part is focused on the foundation of object orientation. It covers the basic OO concepts, including objects, classes, object interaction, collections of objects, polymorphism, encapsulation, cohesion, reusability, and others. The second part covers the more advanced topics, including inheritance, abstract classes, generics, interface, exception handling, OO design and analysis, UML, and others. This course will also cover several topics related to ensuring the quality of object-oriented programs such as testing, debugging, refactoring, and design patterns. The projects in the class will provide applications of these concepts and will include GUIs, event-driven simulations, and multithreading.

**Student Learning Outcomes:**

* Understand the fundamental concepts of object-oriented programming, e.g., object, class, encapsulation, inheritance, abstract classes, and interfaces.
* Understand the fundamental principles of object-oriented design and analysis.
* Be familiar with the basic language constructs of C++.
* Be able to design, write, compile, execute, test, and debug object-oriented programs in C++.

**Required Textbooks and Other Course Materials:**

There are no required physical textbook for Summer 2017 CSE 1325. The primary course textbook is available as a PDF download or can be purchased used if desired. Dr. Tiernan will also use other books and online materials for reference as listed below and described during the class. We will also be using the NetBeans IDE in class but you may use any IDE. However, we will be compiling some programs on gpp on the omega system at UTA so you need to make sure you have your omega ID set up. There is a wealth of accessible material for C++ online. You may want to use some of the online C++ Tutorials that are available. Look for things that are the easiest for you to use.

The primary materials needed for the course are:

The Learn C++ tutorial/chapters at <http://www.learncpp.com/>

And the tutorial at Tutorials Point on C++ <https://www.tutorialspoint.com/cplusplus/index.htm>

The following books and/or materials may be referenced during the semester:

<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-introduction-to-c-january-iap-2011/>

MIT C/C++ course that moves FAST

<http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html>

Online C++ text (2003) Bruce Eckel

[*How To Think Like a Computer Scientist: C++ Version*](http://greenteapress.com/thinkcpp/index.html) Allen B. Downey, Available at

http://greenteapress.com/thinkcpp/index.html

See the additional resources list linked to the class website for more possibilities.

**Attendance:** While the instructor will not take roll after census day, there will be pop quizzes given during the semester. Pop quizzes will be given in class and based on the assigned chapters and class lectures. No make-up quizzes will be given.

**Grading**:

Tentatively, the final grade will be determined according to the following percentages:

Labs 46% {4 labs at 6, 14, 14, and 12 percent} See details below - last assignments will be due Final Review and/or Finals Week

Three (3) in-class quizzes 9% (3 at 3 percent each)

Midterm 20%

Final Exam 25%

*Extra Credit Service Learning 3% (OPTIONAL - see below)*

The first lab assignment will be offered early in the semester and is designed to help students practice with the basics of C++ programming (Input/output, class definition, IDE, etc.). Remember, all lab assignments are **individual effort only**.

CHEATING on exams, PLAGIARISM, or COLLUSION will not be tolerated.

**Descriptions of major assignments and examinations:**

There will be several homework assignments, semester long programming projects referred to as labs, a midterm exam, and a final exam. The final exam will be comprehensive. More detail is given below.

**Exams:** Material covered on the exams will be based on the assigned chapters and class lectures. All exams are mandatory. There are NO make-up exams after the scheduled times. If a student notifies the instructor IN ADVANCE, then an early make-up exam MAY be arranged at the discretion of the instructor. The instructor's decision is final. Make up exams must be arranged in advance and will be scheduled at the discretion of the instructor. All exams may be kept by the instructor.

**Lab Assignments**: No late lab assignments will be accepted without prior written permission from Dr. Tiernan. Lab assignments must run using the **gcc compiler on omega.uta.edu** . The only way to learn programming is to write programs and make them work. The process is to write a chunk of code, test the chunk of code, find the errors, debug the errors, test the code some more, find errors, debug, test, get it working then add another chunk of code and do the same process again. At the beginning of this course you will write programs that need to be developed in pieces. You already have some experience with this from your previous classes. The programs in this class are larger than those in previous classes. This helps you practice with the process of breaking large problems into reasonable sized chunks and turning your thoughts and algorithms into well-structured maintainable programs. As the semester progresses, you will be expected to write larger and larger programs. These will reuse ideas you learned earlier and will let you practice the process of writing testing and debugging.

Every lab assignment has a given due date. No late labs will be accepted without prior written permission from Dr. Tiernan.. (Five minutes late is still late.) Lab assignments will be posted on the Blackboard and additional reference will be on the class webpage as listed at the top of this syllabus.

Lab assignments must be individual effort. The Statement of Ethics linked to the class website details the definitions of collusion, plagiarism, and academic dishonesty as related to lab assignments in CSE1325.

All required labs must be submitted in order to have the possibility of making a grade of A. At least eighty-five percent (85%) of the required labs must be submitted in order for a student to pass CSE 1325 with a C or better. At least seventy-five percent (75%) of the submitted labs must receive passing grades (with no significant errors) in order to pass CSE 1325 with a C or better and at least two of these labs with a passing grade must be from the final three lab assignments.

All of the required lab assignments will be written in C++.

Each lab will be graded on a number of factors. **Always be sure that any C++ lab you submit will compile and run on the the gpp compiler on omega.uta.edu without errors even if it is not complete. You will receive partial credit for a working stubbed program. It is your responsibility to completely test your lab assignment PRIOR to submission.**

Programs that do not compile successfully (without compiler errors) will receive **zero (0) credit**.

Programs that do not execute successfully (without inappropriate termination) will receive **zero (0) credit.**

This means that ALL choices the user can make when running your code must work properly and ALL inputs (excluding ones the lab instructions say you don't have the deal with) must work properly even if the input value is invalid; for example, a negative value for a month must NOT make your program crash.

Labs that implement some, but not all, of the requirements must compile and run successfully without errors to receive partial credit. Examples of errors are:

Compilation errors - these occur when the program is being compiled and prevent creation of an executable.

Compilation warnings - these occur when the program is being compiled, are printed to the screen but still allow the creation of an executable file

Execution errors - these occur once the program has started running and cause the program to terminate in any way other than that defined by the programmer (ex: segmentation fault, divide by zero error, incompatible types, etc.)

**See the lab assignments, the class website and Blackboard for complete instructions on how to compile and submit lab assignments.**

Additional procedural information on lab assignments may be handed out or made available on the website as required. In particular, look for the re-grading policy to be posted with the first lab.

**Quizzes:**

Pop quizzes will be given in class and based on the assigned web material, web chapters chapters and class lectures. No make-up quizzes will be given.

**Dr. T’s philosophy about teaching programming::** The job of a computer scientist/ software engineer / systems analyst / computer engineer / etc. is to use the computer to solve a problem or create a product. These problem solutions and products are software programs. The task of CS/SE/analyst/engineers is determining what kind of solution is needed for a problem and then designing a program to implement it and then writing the code for that program and making it work. To learn all of the skills to do this well is the purpose of pursuing a 4-year degree. To do all of this, there are many things to learn and practice. Part, only part, of the practice is to write programs and learn how to do this well. However, this practice pervades the rest of the tasks so in order to be able to use the skill of writing programs, we have to teach you to write programs and we have to teach this first, like the ABCs, so that it can be used as a building block for learning all the other parts of being a computer scientist/ software engineer / systems analyst / computer engineer / etc.

So, programming is NOT the entire job of being a computer scientist/ software engineer / systems analyst / computer engineer / etc. but it is a significant part. Therefore we teach you programming FIRST so that you can use this skill while you learn the rest of the really interesting aspects of being a computer scientist/ software engineer / systems analyst / computer engineer / etc. throughout the rest of your degree. Learning to program is like learning your ABCs. You do this in order to learn how to do even more interesting things later. :)

**Drop Policy:** Students may drop or swap (adding and dropping a class concurrently) classes through self-service in MyMav from the beginning of the registration period through the late registration period. After the late registration period, students must see their academic advisor to drop a class or withdraw. Undeclared students must see an advisor in the University Advising Center. Drops can continue through a point two-thirds of the way through the term or session. It is the student's responsibility to officially withdraw if they do not plan to attend after registering. **Students will not be automatically dropped for non-attendance**. Repayment of certain types of financial aid administered through the University may be required as the result of dropping classes or withdrawing. For more information, contact the Office of Financial Aid and Scholarships (<http://wweb.uta.edu/ses/fao>).

**Americans with Disabilities Act:** The University of Texas at Arlington is on record as being committed to both the spirit and letter of all federal equal opportunity legislation, including the *Americans with Disabilities Act (ADA)*. All instructors at UT Arlington are required by law to provide "reasonable accommodations" to students with disabilities, so as not to discriminate on the basis of that disability. Any student requiring an accommodation for this course must provide the instructor with official documentation in the form of a letter certified by the staff in the Office for Students with Disabilities, University Hall 102. Only those students who have officially documented a need for an accommodation will have their request honored. Information regarding diagnostic criteria and policies for obtaining disability-based academic accommodations can be found at [www.uta.edu/disability](http://www.uta.edu/disability) or by calling the Office for Students with Disabilities at (817) 272-3364.

**Ethics and Academic Integrity:**

A Statement of Ethics is available on the class website for you to read, sign, return, and follow. This will be reviewed in class. Violators of the ethics code will be reported to the Vice-President for Academic Affairs and penalties will be levied as described in the Statement of Ethics.

**Student Support Services**:UT Arlington provides a variety of resources and programs designed to help students develop academic skills, deal with personal situations, and better understand concepts and information related to their courses. Resources include tutoring, major-based learning centers, developmental education, advising and mentoring, personal counseling, and federally funded programs. For individualized referrals, students may contact the Maverick Resource Hotline by calling 817-272-6107, sending a message to [resources@uta.edu](mailto:resources@uta.edu), or visiting [www.uta.edu/resources](http://www.uta.edu/resources).

**Electronic Communication:** UT Arlington has adopted MavMail as its official means to communicate with students about important deadlines and events, as well as to transact university-related business regarding financial aid, tuition, grades, graduation, etc. All students are assigned a MavMail account and are responsible for checking the inbox regularly. There is no additional charge to students for using this account, which remains active even after graduation. Information about activating and using MavMail is available at <http://www.uta.edu/oit/cs/email/mavmail.php>.

**Student Feedback Survey:** At the end of each term, students enrolled in classes categorized as lecture, seminar, or laboratory will be asked to complete an online Student Feedback Survey (SFS) about the course and how it was taught. Instructions on how to access the SFS system will be sent directly to students through MavMail approximately 10 days before the end of the term. UT Arlington’s effort to solicit, gather, tabulate, and publish student feedback data is required by state law; student participation in the SFS program is voluntary.

**Final Review Week:** A period of five class days prior to the first day of final examinations in the long sessions shall be designated as Final Review Week. The purpose of this week is to allow students sufficient time to prepare for final examinations. During this week, there shall be no scheduled activities such as required field trips or performances; and no instructor shall assign any themes, research problems or exercises of similar scope that have a completion date during or following this week *unless specified in the class syllabus*. During Final Review Week, an instructor shall not give any examinations constituting 10% or more of the final grade, except makeup tests and laboratory examinations. In addition, no instructor shall give any portion of the final examination during Final Review Week. During this week, classes are held as scheduled. In addition, instructors are not required to limit content to topics that have been previously covered; they may introduce new concepts as appropriate.

**Service Learning Extra Credit:**

This class will offer the opportunity for you to earn up to 3% extra credit for the semester grade by performing some service to the CSE department, the UT Arlington College of Engineering, the University, or the community. The service must be completed and documentation turned in by the Friday BEFORE Dead Week. The maximum service extra credit is 3% regardless of the amount of service your eventually complete. The service learning credit structure is as follows:

3% extra credit is available for :

- 4 or more hours of service to the CSE dept. or the College of Engineering documented by a faculty or staff member, OR

- 6 or more hours of service to the University, documented by a faculty or staff member, OR

- 8 or more hours of service to the community with Dr. T’s prior permission and documented by a staffer of the community organization;

2% extra credit is available for :

- 2 to 4 hours of service to the CSE dept. or the College of Engineering OR

- 4 or more hours of service to the University OR

- 6 or more hours of service to the community;

1% extra credit is available for :

- 2 to 4 hours of service to the University OR

- 4 or more hours of service to the community

CSE dept. service includes volunteering at approved events, joining ACM, GDC, or AAAI and assisting with their events, or other CSE service approved in advance by Dr. T. Approved events include the RoPro Contest (Feb.), Engineering Saturday (Sept., Nov. and Apr.), FIRST Tech Challenge (Mar.), FIRST Lego League (Nov.), and the RoPro Camp (July). This is not a complete list but it gives some examples.

College service includes volunteering at the Engineering Welcome Bash, joining ESC/ JCEO, volunteering at Engineering Summer Camps, and assisting with ESC events such as Engineers Week, or other College of Engineering service approved in advance by Dr. T.

University service includes activity with UTA Volunteers, FLOC, or the BIG EVENT, or other UTA service approved in advance by Dr. T.

"Community" service includes volunteering at public schools, parks, animal shelters, and homeless shelters or other community service approved in advance by Dr. T.

Other venues for service (within UTA or within the community) must be approved in advance by Dr. T prior to your service.

**Miscellaneous:**

The class syllabus, schedule, and other information will be available on my website and on Blackboard as it is developed. YOU are responsible for checking the website and Blackboard regularly for information such as due date changes and assignments.

If you require accommodation based on disability, I would like to meet with you in the privacy of my office during the first week of the semester to ensure that you are appropriately accommodated.

If you are considering dropping this class please come discuss your performance in the class with Dr. Tiernan so that you can make the best choice.

Students who are members of the Honors College may wish to take this course for Honors credit. If you wish to do so, please provide me with an Honors Credit Contract (downloaded from http://honors.uta.edu/documents/credit.pdf). You and I will together determine an appropriate supplemental assignment to justify the awarding of Honors credit. If you are not in the Honors College and would like to learn more about the benefits of membership, visit the website at http://honors.uta.edu/, where you will find an application form for electronic submission.

Grading issues:

Requests for re-evaluation of assignments are limited to seven (7) calendar days after the assignment is returned. Every assignment submitted for re-grading must be given to the instructor (exams and quizzes) or lab instructor (labs) in its entirety and will be completely re-graded. Papers will not be re-evaluated in the classroom or lab.

Applications for excluding (or replacing) the grade in a course are available online from the Registrar's office (Office of Records) and must be turned in to the Registrar before the last drop day of the semester in which the course is being retaken.

Semester grades will be available via the UTA website, after the Registrar has completed processing the semester grades.

**How to be successful in Dr. T's class:**

For lecture material - Read the textbook in advance. Come to class with material already a little bit familiar then hear it in class again

For lab assignments - START EARLY! Read and follow ALL the instructions. Don't pick and choose which ones you pay attention to. Use a highlighter to go over the assignment the first time and a pen or pencil to go through it a second time.

Design your lab first, then build it piece by piece. Make each piece work and TEST it before you do more. Save a version of the tested partial program then make a copy and continue working on the copy. This means you always have a working version that you could turn in. Test your program by trying out all the possible things that can be wrong and make sure that your program keeps running. If Dr. T says the program does not have to handle a particular problem then you can ignore that one.

How to get help:

This course offers assistance through weekly office hours for the course instructor and the TA. USE THEM. Go talk to the TA as soon as you can just so you can meet them. Once you meet them, it is easier to ask them for help. Go talk to Dr. T in her office as soon as you can. Same reason.

It can also be helpful to meet occasionally with your colleagues from the class to discuss approaches to problems, get debugging help, compare results, and so on. You may NOT write C++ code with another person but you may ask for help with your existing code and you may ask for and offer help in planning your code and in looking at the results of the code.