CS 332 – C++ Programming

Course Syllabus: Fall 2018

"Texas A&M University-Texarkana is a comprehensive regional University that provides students with academically challenging, engaging, and rewarding educational experiences through quality teaching, scholarship, student support services, co-curricular programming, research, and service."

Dr. Kevin P Rose

Phone: (903) 466-1115 (cell)
Email: krose@tamut.edu
PLEASE USE BLACKBOARD MAIL FOR CLASS CORRESPONDENCE

Office Hours	The best way to communicate with me is via Bb email. You should hear back from me within 24 hours, excluding weekends/holidays. In addition, I am very welcome to set up a specific time I can "meet" with you via phone, chat, video conference (Zoom – Room 382-742-2059).

The information contained in this syllabus is subject to change without notice. Students are expected to be aware of any additional course policies presented by the instructor during the course.

Catalog Course Description:

This course introduces students to C++ programming language, a dominant language in the industry today. Students will be taught the fundamentals of programming. These concepts are applicable to programming in any language. Topics covered include basic principles of programming using C++, algorithmic and procedural problem solving, program design and development, basic data types, control structures, functions, arrays, pointers, and introduction to classes for programmer-defined data types.

Prerequisites:

COSC 1315 – Introduction to Computer Science, or Instructor Permission

Required Textbook(s):

C++ Programming: From Problem Analysis to Program Design, 8/E, D.S. Malik, ©2018, Publisher: Cengage Learning

Reading assignments may also be made from the Internet and other external sources.

Resources:

Your instructor can be your greatest resource. I am here to assist you in learning the material and helping you earn the grade you wish to earn in the course. This is my commitment to you. Please utilize this "resource" by contacting me with any matter you feel I can assist you with, both within this class, or your university success in general. It is your responsibility to learn the material, but this can often be best accomplished by initiating contact with me on topics you need clarification or further assistance. Please do this!

Student Learning Outcomes:

1. Analyze the given problem statements to create basic program designs.

2. Implement different functions for input and output, various data types, basic operators, files and functions.
3. Demonstrate basic object oriented and structured programming concepts.
4. Implement programming techniques to solve problems in the C++ programming language.
5. Apply the concepts and principles of the programming language to the real-world problems and solve the problems through project-based learning.

Class Calendar:

Week	Start	End	Topic	Text Ref.
1	27-Aug	2-Sep	An Overview of Computers and Prog. Languages	Chpt 1
2	3-Sep	9-Sep	Basic Elements of C++ (Week 1)	Chpt 2
3	10-Sep	16-Sep	Basic Elements of C++ (Week 2)	
4	17-Sep	23-Sep	Input/Output	Chpt 3
5	24-Sep	30-Sep	Selection -- Control Structures I (Week 1)	Chpt 4
6	1-Oct	7-Oct	Selection -- Control Structures I (Week 2)	
7	8-Oct	14-Oct	Repetition -- Control Structures II (Week 1)	Chpt 5
8	15-Oct	21-Oct	Repetition -- Control Structures II (Week 2)	
9	22-Oct	28-Oct	User-Defined Functions (Week 1)	Chpt 6
10	29-Oct	4-Nov	User-Defined Functions (Week 2)	
11	**5-Nov**	**11-Nov**	**Mid Term**	**Online**
12	12-Nov	18-Nov	Arrays and Strings (Week 1)	Chpt 8
13	19-Nov	25-Nov	Arrays and Strings (Week 2)	
	22-Nov	**23-Nov**	**Thanksgiving Holiday (University Closed)**	
14	26-Nov	2-Dec	Classes and Data Abstraction (Week 1)	Chpt 10
15	3-Dec	9-Dec	Classes and Data Abstraction (Week 2)	
16	**10-Dec**	**12-Dec**	**Final Exam**	**Online**

Please Note: For a full listing of institutional dates (i.e. drop dates, graduation filing dates, etc.), the Texas A&M-Texarkana Academic Calendar can be found online at:

http://catalog.tamut.edu/academic-calendar/ - Fall2018

Evaluation/Grading Policy:

Grades of A, B, C, D, and F will be determined by the student's achievement of a total number of possible points based on the below listed categories.

Assignments	Quantity	Pts. Per Assign.	Total Pts. Possible
Homework assignments			825
Syllabus Quiz	1	20	20
Introduction/Bio Discussion Board Post	1	20	20
Academic Integrity and Collaboration Acknowledgement	1	10	10
Chapter Homework Assignments	10	25	250
Programming Assignments	7	75	525
Exams			200
Mid-Term	1	100	100
Final	1	100	100
Total Possible Points			1025

The letter grade is based on the following Grading Scale:

Total Pts. Earned	Ltr. Grade
>= 900	A
800-899	B
700-799	C
600-699	D
Below 600	F

Tests/Exams:

You will be required to take both exams in order to be successful in this course. You will received an 'F' in the course if one or both tests are not taken.

Exam 1 – Mid-term – Material covered during first half of the semester;
Exam 2 – Final – While the focus of this exam will be on material covered in the second half of the semester, due to the building-block nature of computer programming, an understanding of the material covered in the first half of the semester will be necessary.

The exams will consist primarily of short answer, code writing, and "play the computer", styled questions.

If you have a conflict with the exam dates, you must contact me prior to the exam week to attempt to resolve the conflict in a mutually agreeable manner.

Assignments:

SYLLABUS QUIZ:

The syllabus quiz will cover the contents of the syllabus and attempt to ensure we have a mutual understanding of how the course will operate and expectations of all parties. This quiz will be offered through BlackBoard and should be taken during the first week of class.

ACADEMIC INTEGRITY ACKNOWLEDGEMENT:

You are required to read the *Academic Integrity and Collaboration* section of this document and acknowledge you understand the parameters of the course as they relate to assignments and exams and the concept of academic dishonesty.

CHAPTER HOMEWORK ASSIGNMENTS:

Numerous homework assignments will be given in an effort to reinforce the material covered in each of the chapter modules.

PROGRAMMING ASSIGNMENTS:

Instructor commentary: First, I want you to be successful in this course. As a production programmer, my experience leads me to believe the only way to learn to program *is to write programs*. Additionally, the skill of problem solving and logic determination can only be gained through experience, and takes time and effort. This can be very frustrating, and takes a level of determination focused on solving the problem and reaping the intrinsic reward of knowing you "did it". The skill is very rewarding, and I am fully prepared to assist you in building a strong foundation in coding, but I cannot do this alone. Programming is a skill that is unlike many other topics, in that exposure to the material is not enough; it is a skill that must be practiced through true hands-on activities and requires much thought and concentration.

Please be prepared to spend at a minimum six-seven hours per week on this course (recall, this is an online class, so in exchange for face-to-face class time, you must still be prepared to spend the time you would have spent in class immersing yourself in learning the course material).

Over Collaboration:

You are expected to turn in your own original work. Getting help in finding an error is encouraged, but copying other student's work or code from other sources is forbidden and will result in a grade of zero for that assignment or for the entire course. You need to be able to explain your program – See Academic Integrity and Collaboration Policy.

Computer crashes, Unexpected Deletions, Power Failures, Dog Ate My Homework:

As a student developing a professional skill, it is imperative that you begin practicing the discipline you will need to be successful in your career. In the computer programming field, this is never more apparent than that associated with time management and disaster recovery. The statement that all professional programmers fully understand is *"not if a disaster will occur, but when"*. We all make mistakes, including accidentally deleting our primary source code file – **we must practice having a recent back up**. The power will unexpectantly go off – **we must practice having a recent back up**. Many other "my dog ate my homework" issues will arise – **we must practice having a recent back up**. Bottom line -- we know we are in a technology-oriented field and must take precautions against data loss, etc. These issues will not be reasons for delayed homework submissions.

Regrade Request:

If you feel there was an error in the grading of your submission, you may request the submission be re-evaluated. This may be done through the Regrade Request link within course.

Student Responsibilities/Expectations:

Online Access: Students must login to the class during the first week of the semester and post to the discussion board through BlackBoard (Introduction/Bio Posting). Submission of assigned work will count as communication and serve to reflect your commitment to the class for enrollment purposes.

Class Attendance: Participation on the class website will constitute “attendance.” Withdrawal requests MUST BE initiated by the student. The last day for a student to drop a course with a grade of "W" is **Friday, November 16, 2018.** Requests for withdrawal become official and effective the date they are received in the records office. Students who stop participating in the class, but fail to drop the course will *earn* an “F” for the course.

Late Work: To be considered “on time,” **all work must be submitted the day it is due**. This means before midnight.

Extension Tokens: You will be given five “Extension Tokens” that may be used to “buy” more time on an assignment. These virtual tokens can be used at any point for any assignment. Each token can be used for a single day extension (no deduction in points). You must email me (see format below) before the due date and not during the extension period. You may only use two tokens, at most at any one time (i.e. two tokens = 2 days extension with no deduction). **Once you have used all of your tokens, you will be assessed a 15% late penalty per day late (maximum of three days late)**. If you have tokens, you must use them, you cannot choose to hold tokens and accept the 15% per day penalty. No assignments will be accepted after three days past due date.

Your balance of tokens will be kept in the Bb Gradebook.

No assignment turned in after the due date will be graded without the submission of an acknowledgement. **You must submit either a late token email or a late penalty email to prompt late work to be graded**. The submission of an extension token consists of sending an email to me through BlackBoard with the following:

Late Token Email

Subject: LATE <assignment name>

Content:
“I am submitting _____ (max 2) tokens for the assignment <assignment name>. It was due on ______, and I submitted it on ______________.

This leaves me with ____ late tokens.”

Late Penalty Email

Subject: LATE <assignment name>

Content:

"I am submitting <assignment name> late. I do not have any late tokens remaining."

Advice: Choose when to use your late tokens wisely. Try your best to get your assignments turned in on time, especially early in the semester before your workload piles up. If you are going to use a token, take your time, 1 minute late is the same as 23:58 minutes late. Learning is the goal. I want you to master the content, but we must be responsible in our learning.

Academic Integrity:

Academic honesty is expected of students enrolled in this course. Cheating on examinations, unauthorized collaboration, falsification of research data, plagiarism, and undocumented use of materials from any source constitute academic dishonesty and may be grounds for a grade of 'F' in the course and/or disciplinary actions. For additional information, see the university catalog.

IMPORTANT: The Collaboration and Academic Integrity Policy is attached to this document. Please read the document thoroughly, and ask any questions you might have. You will be required to sign an acknowledgment of understanding during the first week of the course.

A&M-Texarkana Email Address:

Upon application to Texas A&M University-Texarkana an individual will be assigned an A&M-Texarkana email account. This email account will be used to deliver official university correspondence. Each individual is responsible for information sent and received via the university email account and is expected to check the official A&M-Texarkana email account on a frequent and consistent basis. Faculty and students are required to utilize the university email account when communicating about coursework.

Drop Policy:

To drop this course after the census date, a student must complete the Drop/Withdrawal Request Form, located on the University website (https://www.tamut.edu/Admissions/Enrollment-services/Registrar/Dropping.html) or obtained in the Registrar's Office. The student must submit the signed and completed form to the instructor of each course indicated on the form to be dropped for his/her signature. The signature is not an "approval" to drop, but rather confirmation that the student has discussed the drop/withdrawal with the faculty member. The form must be submitted to the Registrar's office for processing in person, email Registrar@tamut.edu, mail (7101 University Ave., Texarkana, TX 75503) or fax (903-223-3140). Drop/withdraw forms missing any of the required information will not be accepted by the Registrar's Office for processing. It is the student's responsibility to ensure that the form is completed properly before submission. If a student stops participating in class (attending and submitting assignments) but does not complete and submit the drop/withdrawal form, a final grade based on work completed as outlined in the syllabus will be assigned.

Student Support Services:

Here at TAMUT there are many support service offerings including academic advising, student success center, career services, developmental education, disability services, first-year experience, testing and writing centers. To access links for support visit the Student Support website.

Disability Accommodations:

Students with disabilities may request reasonable accommodations through the A&M-Texarkana Disability Services Office by calling 903-223-3062.

Student Online Education Technical Assistance:

Solutions to common problems and FAQ’s for your web-enhanced and web courses are found at this link: https://www.tamut.edu/Academics/Online-Education/index.html. I would like to encourage you to go through all the links under the “Online Readiness” heading, especially if this is your first online course. If you cannot find your resolution there, you can send in a support request detailing your specific problem by emailing helpdesk@tamut.edu.

Helpdesk contacts (office hours are: Monday - Friday, 8:00am to 5:00pm)

- Julia Allen (main contact) 903-223-3154 allen@tamut.edu
- Linda Scott (alternate) 903-223-3152 scott@tamut.edu

Course relation to ABET learning outcomes

This course addresses the following Accreditation Board for Engineering and Technology (ABET) outcomes including the ability to:

- (3c) Formulate or design a system, process, or program to meet desired needs.
- (3e) Identify and solve applied science problems.
- (3g) Communicate effectively.
- (3k) Use the techniques, skills, and modern scientific and technical tools necessary for professional practice.

MAPPING among course learning-objectives and ABET student learning outcomes and problems where outcomes are assessed				
Outcome Related Course Learning Objective	**ABET Outcomes**			
	3c	3e	3g	3k
Analyze the given problem statements to create basic program designs.	Program Assgn. Test 1		Program Assgn. Test 1	
Implement different functions for input and output, various data types, basic operators, files and functions.	Program Assgn. Test 1,2			Program Assgn. Test 1,2
Demonstrate basic object oriented and structured programming concepts.		Program Assgn. Test 2	Program Assgn. Test 2	
Implement programming techniques to solve problems in the C++ programming language.	Program Assgn. Test 1, 2	Program Assgn. Test 1, 2		Program Assgn. Test 2
Apply the concepts and principles of the programming language to the real-world problems and solve the problems through project based learning.		Program Assgn. Test 2		Program Assgn. Test 2
Objective addresses outcome: 1 = slightly, 2 = moderately, 3 = substantively				
(3c) Formulate or design a system, process, or program to meet desired needs. (3e) Identify and solve applied science problems. (3g) Communicate effectively. (3k) Use the techniques, skills, and modern scientific and technical tools necessary for professional practice.				

Academic Integrity & Collaboration Policy

Introduction

This document seeks to establish a basic understanding of how to appropriately collaborate with others without violating the academic integrity of the course. It is my goal to create an environment where each person and the discipline itself is respected and all are focused on learning.

I want you to learn from your fellow students, as this is often the best approach both in and out of academic settings, but the work you submit must be your own. Therefore, the following guidelines give you some structure to follow in your pursuit of such during this course. As with all academic questions, if you may be misunderstanding something, or need clarity, please don’t hesitate to ask.

Academic Integrity for Exams

All exams for this course will be completed individually. Copying, communicating, or using disallowed materials during an exam is cheating, of course. Students caught cheating on an exam will receive an F in the course and will be reported to the Dean of Students for further disciplinary action.

Collaboration Policy for Programming Labs

Collaboration is encouraged during the weekly programming labs. Students are allowed to talk through and assist each other with these programming exercises. Students may ask for help from each other. But each student must write up and debug their own lab solutions on their own computer and be prepared to present and discuss this work with the instructor to receive credit for each checkpoint.

As a general guideline, students may look over each other's shoulders at their peer’s computer screen — this is a good way to learn about IDEs, code development strategies, testing, and debugging. However, looking should not lead to line-by-line copying. Furthermore, each student should retain control of their own keyboard. While being assisted by a classmate, the student should remain fully engaged on problem solving and ask plenty of questions. Finally, other than the specific files provided by the instructor, electronic files or file excerpts should not be shared or copied (by email, text, Dropbox, or any other means).

Homework Collaboration Policy

Academic integrity is a complicated issue for individual programming assignments, but one we take very seriously. Students naturally want to work together, and it is clear they learn a great deal by doing so. Getting help is often the best way to interpret error messages and find bugs, even for experienced programmers. Furthermore, in-depth discussions about problem solving, algorithms, and code efficiency are invaluable and make us all better software engineers. In response to this, the following rules will be enforced for programming assignments:

- Students may read through the homework assignment together and discuss what is asked by the assignment, examples of program input & expected output, the overall approach to tackling the assignment, possible high-level algorithms to solve the problem, and recent concepts from lecture that might be helpful in the implementation.

- Students are not allowed to work together in writing code or pseudocode. Detailed algorithms and implementation must be done individually. Students may not discuss homework code in detail (line-by-line or loop-by-loop) while it is being written or afterwards. In general, students should not look at each other's computer screen (or hand-written or printed assignment design notes) while working on homework. As a guideline, if an algorithm is too complex to describe orally (without dictating line-by-line), then sharing that algorithm is disallowed by the homework collaboration policy.

- Students are allowed to ask each other for help in interpreting error messages and in discussing strategies for testing and finding bugs. First, ask for help orally, by describing the symptoms of the problem. For each homework, many students will run into similar problems and after hearing a general description of a problem, another student might have suggestions for what to try to further diagnose or fix the issue. If that doesn't work, and if the compiler error message or flawed output is particularly lengthy, it is okay to ask another student to briefly look at the computer screen to see the details of the error message and the corresponding line of code.

- Students may not share or copy code or pseudocode. Homework files or file excerpts should never be shared electronically (by email, text, LMS, Dropbox, etc.). Homework solution files from previous years (either instructor or student solutions) should not be used in any way. Students must not leave their code (either electronic or printed) in publicly-accessible areas. Students may not share computers in any way when there is an assignment pending. Each student is responsible for securing their homework materials using all reasonable precautions. These precautions include: Students should password lock the screen when they step away from their computer. Homework files should only be stored on private accounts/computers with strong passwords. Homework notes and printouts should be stored in a locked drawer/room.

- Students may not show their code or pseudocode to other students as a means of helping them. Well-meaning homework help or tutoring can turn into a violation of the homework collaboration policy when stressed with time constraints from other courses and responsibilities. Sometimes good students who feel sorry for struggling students are tempted to provide them with "just a peek" at their code. Such "peeks" often turn into extensive copying, despite prior claims of good intentions.

- All collaborators (classmates, tutors, upperclassmen, students/instructor via LMS, etc.), and all of the resources (books, online reference material, etc.) consulted in completing this assignment must be listed in the README.txt file submitted with the assignment.

These rules are in place for each assignment and extends two days after the submission deadline.

Cheating

Cheating is a very serious offense. If you are caught cheating, you can expect a failing grade and initiation of a cheating case in the University system. Cheating is an insult to other students, to the instructor, and to yourself. If you feel that you are having a problem with the material, or do not have time to finish an assignment, or have any number of other reasons to cheat, then talk with the instructor. Copying the work of others is not the solution.

To avoid creating situations where copying can arise, never e-mail or post your solution files. You can post general questions about interpretation and tools but limit your comments to these categories. If in doubt about what might constitute cheating, send the instructor email describing the situation.

Students found in violation of the academic dishonesty policy are prohibited from dropping the course in order to avoid the academic penalty.

Academic Dishonesty in the Student Handbook

Refer to http://catalog.tamut.edu/academic-information/ for further discussion of academic integrity and Texas A&M-Texarkana Student Honor Code.

Acknowledgments: Parts of this document are based upon the following:

- *Ransselaer Polytechnic Institute (RPI) – CSCI 1200 Data Structures Fall 2017 syllabus*