



CS 332 – C++ Programming

Course Syllabus: Fall 2016

“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 (such as Skype [user id: kevinprose]) or BlackBoard Collaborate.

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++ How to Program, 9/E, Paul Deitel & Harvey Deitel, ©2014, ISBN-10: 0133378713, ISBN-13: 9780133378719, Publisher: Prentice Hall

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.

Lectures & Discussions:

An expanded schedule, including topics, assignments, readings, discussions, and due dates will be posted as part of the online environment.

Date	Topic	Text Ref.
Week 1 (Aug 22-29)	Introduction and software setup, first program	BlackBoard
Week 2 (Aug 29-Sept 5)	Introduction to C++ Programming	Chpt 2
Week 3 (Sept 5-12)	Introduction to Classes, Objects, and Strings	Chpt 3
Week 4 (Sept 12-19)	Control Statements: Part I; Assignment ++ and -- Operators	Chpt 4
Week 5 & 6 (Sept 19-Oct 3)	Control Statements: Part II; Logical Operators	Chpt 5
Week 7 (Oct 3-10)	Functions and an Introduction to Recursion	Chpt 6
Week 8 (Oct 10-17)	Exam 1 Mid-Term	
Week 9 & 10 (Oct 17-31)	Class Templates array and vector; Catching Exceptions	Chpt 7
Week 11 & 12 (Oct 31-Nov 14)	Pointers	Chpt 8
Week 13 & 14 (Nov 14-28)	Classes: A Deeper Look	Chpt 9
Week 15 (Nov 28-Dec 5)	Second Half Recap Assignment/Review	
Week 16 (Dec 5-8)	Exam 2 – Final (Note: December 8 is on Thursday, which will be last day to take the Final Exam)	

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/-fall2016text>

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.

<u>Assignments</u>	<u>Quantity</u>	<u>Pts. Per Assign.</u>	<u>Total Pts. Possible</u>
Homework assignments			700
Syllabus Quiz	1	10	10
Introduction/Bio Discussion Board Post	1	5	5

Which Compiler? Discussion Board Post	1	5	5
Chapter Homework Assignments	8	25	200
Programming Assignments	8	60	480
Exams			300
Mid-Term	1	150	150
Final	1	150	150
Total			1000

The letter grade is based on the following Grading Scale:

Total Pts. Earned	Ltr. Grade
900-1000	A
800-899	B
700-799	C
600-699	D
Below 600	F

Tests/Exams:

You will be required to take two exams in order to be successful in this course.

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.

CHAPTER HOMEWORK ASSIGNMENTS:

Numerous homework assignments will be given in an effort to reinforce the material covered in each of the chapter modules. The answer to these assignments should be typed unless otherwise denoted, and must be submitted through BlackBoard.

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, 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).

Programming Assignment Guidelines:

1. A copy of your source program and its output should be submitted in BlackBoard. Program output should be labeled (by the program) so that it is understandable to the reader, with the assignment number, date, and your name.
2. Testing is a critical part of the programming process. The burden of proof that a program works always rests with the programmer. Whether or not test data are provided, you must adequately test your program to insure that it works correctly in all reasonable cases.
3. Programming assignments may specify that a particular approach be used. Read the assignment carefully to be sure that you understand how the problem should be solved. Part of computer programming is being able to follow specification, and subsequently, your solutions should also do so.
4. Programs must be well structured, readable, and efficient. Use meaningful names, indentation, comments, and other elements of style discussed in the course. An unreadable program is not easily maintainable and, when put into production, will often cause more inefficiencies in the future than the savings realized at the onset of initial development.
5. Output should be neat, properly aligned, and have useful headings. Requests for interactive input should be preceded by a prompting message.
6. Do not jeopardize your grade by allowing others to copy your work. The penalties for giving and receiving help are the same.

Program assignments will be evaluated against a standard rubric that can be found on the class's BlackBoard site. Your submissions will be graded on the following general criterions:

- Program Specifications/Correctness (50%)
- Readability (20%)
- Documentation (20%)
- Code Efficiency (5%)
- Assignment Specifications (5%)

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.

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 expectantly 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.

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.

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 11, 2016**. 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. Assignments can be submitted for a 10% deduction per day for one week following the assignment due date (after the "late" week has passed, no assignments will be accepted). If there are extenuating circumstances, please contact your instructor.

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.

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
- Nikki Thomson (alternate) 903-223-3083 thomson@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.	1 Lab Test 1		1 Lab Test 1	
Implement different functions for input and output, various data types, basic operators, files and functions.	2 Lab Test 1,2			1 Lab Test 1,2
Demonstrate basic object oriented and structured programming concepts.		1 Lab Test 2	2 Lab Test 2	
Implement programming techniques to solve problems in the C++ programming language.	3 Lab Test 1, 2	2 Lab Test 1, 2		2 Lab Test 2
Apply the concepts and principles of the programming language to the real-world problems and solve the problems through project based learning.		3 Lab Test 2		3 Lab 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.				