

CIS 3301: Introduction to Business Application Programming Fall, 2019

Instructor: Professor Peter Kirs

Email: pkirs@utep.edu

CRN: 11941

Meeting Times: TR 10:30 – 11:40 PM, Room: BUSN 320

Office: CoBA 203; Office Hours: TR, 10:00 a.m. – 10:30 am, TR 3:00– 4:00 pm, and by appointment (I am generally here every workday)

Course Description

This course introduces the student to programming in the most widely used computer languages: C/C++. Emphasis will be on good programming practices, following top-down, modular, structured program design development, testing, implementation and documentation of business problems.

Course Expectations

You are expected to attend every class. If you do not come to class and work on improving your programming skills, you can't learn. I can facilitate your learning, but you are the only person who can develop your knowledge.

Why??

The explanation is very simple. How many of you could attend a cello concert, say by Yo-Yo Ma (probably the world's greatest living cellist) then go home and play the cello as well as he does? The answer is: none of you. The next question is, how many of you think that they can learn how to program by just watching me program in class? (Hint: the answer is the same). You can only learn programming through practice.

This class will be (almost) completely hands-on. Each day, you will be expected to enter in source code, compile, run the code, and correct the code that fails. Unlike many classes, you are strongly encouraged to make mistakes. You can only learn by making mistakes. This corresponds to every experience in your life. If you don't make mistakes, you will learn nothing.

This sounds very obvious, yet I have learned that most students don't believe it. Every time I finish introducing a new concept in class, I always ask "Who doesn't understand what I just said?" Typically, no one raises their hand. I understand that no one wants to seem to be stupid (and none of you are), but all of you are ignorant (note: stupid is defined as "characterized by or proceeding from mental dullness; foolish; senseless" whereas ignorant is defined as "lacking knowledge or information as to a particular subject or fact"). Hence, you are all ignorant -- that includes me (why else would be taking this course?). You can eliminate your ignorance by asking questions and demanding to understand what is being taught. If you don't, guess what term applies to you?

Need for Assistance

If you have any condition, such as a physical or learning disability, which will make it difficult for you to carry out the work as I have outlined it, or which will require academic accommodations, please notify me as soon as possible. The Center for Accommodations and Support Services (CASS) aspires to provide students with disabilities, accommodations and support services to help them pursue their academic, graduation, and career goals. Please note that outside of the class you can get assistance from the SAM-Cengage or Technology Support at the Library.

Academic Dishonesty

The University expects all students to conduct themselves with the highest level of integrity. You have the opportunity throughout your business career to demonstrate your own level of integrity. Similarly, in this class you will have an opportunity to demonstrate academic integrity. The two are inextricably linked. And let's not be naïve: issues of integrity are rarely black and white – they are invariably some shade of gray. I encourage you to think about the standard you set for business integrity in your career, and to implement that standard with respect to your academic integrity in this class. In so doing, you might want to keep the following quote from Samuel Johnson in mind: “The chains of habit are too weak to be felt until they are too strong to be broken”.

Academic dishonesty in any form will not be tolerated. For a definition of academic dishonesty and its consequences, see your student handbook.

For the purposes of this course, academic dishonesty includes, but is not limited to,

- Copying another student's solution or allowing another student to copy your solution.
- Removing or copying pages/problems from exams/quizzes, including those posted on the WWW
- Any other activity that jeopardizes the integrity of this course.

Evaluation:

Grades are not given; they are **EARNED**. You must work for it. Your grade will be based on results rather than on effort — your performance is an indicator of your ability to master the topic. Decide to work **NOW** for the grade you want. Students who keep up with the materials, do all of the assignments, and participate in the learning experiences typically do well. Your grade will be calculated using the following scale:

Grade	Level of Work	Percentage Range
A	Excellent, distinguished	90 – 100%
B	Very good, above average	80 – 89%
C	Average, Normal	70 – 79%
D	Below Average	60 - 69%
F	Failing	< 60%

A grade of "**I**" (**Incomplete**) can only be assigned in very extreme cases (e.g., death, being marooned on a deserted island with no electricity and thus no access to the internet for a very long period, as well as having broken fingers so that you couldn't type might (or might not) be sufficient). If you are taking this course under the S/U option, you must earn a "B" or better to receive a grade of "S."

Topics

Downloading Codeblocks (A free compiler)
Your First Program (Yes, we will start coding immediately)
Overview of Programming (what you are actually doing)
Basic Data Types
C/C++ Operators
Arrays and Strings
Flow Control
In-class Programming
File Input/Output
Structured data objects (Structs)
Searching

What dates are associated with the topics? I don't know. After each topic, I will ask if you understand the topic and if you have any questions. If I think most of you are struggling, we will keep on that topic. I have never completed the full list above.

Tests

We will have three in-class tests. The first two will test your general understanding of the topic (no programming), the third will ask you to reproduce a portion of your project.

You are not allowed to have telephones, notes, calculators, or access to the internet.

Project

The last few weeks of the semester will be devoted to in-class programming of your project. You will electronically submit your source code to me, and I will compile, run, and evaluate your project. Each semester, I receive multiple programs that all make the same errors, and have the same source codes. I will send these to the Dean of Students, and let her determine what should be done.

Keep in mind what I stated above, namely that "the fourth [test] will ask you to reproduce a portion of your project". Each semester, I also have students who turn in perfect projects, but who also turn in the fourth test, which are considerably easier than the project, but do not compile. Why do you think that is?

Summary

Be like Yo-Yo. Practice, practice, practice.