CSCI 151   Intro to Computer Science & Programming   MW 09:30-10:45 Jour 234
Fall 2015

Instructor: Chris Jones
Office: JOUR 211 (follow the hallway to the left of JOUR 222 or to the right of JOUR 218)
Phone: 903-780-6998 (SCSI Dept. Office in JOUR 122-124 is 903-886-5409)
Email: chris.jones@tamuc.edu

Office HRS: M   2:00pm – 3:00   If I'm not in my office during office hours, look for a note on
              T   2:30pm – 4:30 *   the door. If there's no note, you can check with the CSCI dept
              W   2:00pm – 3:00   admin (Mrs. Rose) in Jour 122-124 to be sure I'm on campus.
                                                 Unless I'm sick that day, I'm probably on my way and will be there
                    or by appointment   as soon as I can.

Student Data and Resource site
http://1drv.ms/1TZzgya
- **Labs**
- **Quizzes**
- **Program examples**
- **PowerPoints**
- **Syllabus**
- **Assignments**
- **Extra Credit (If Applicable)**

Catalog Description:
This is a lecture and laboratory course offered to introduce basic concepts of computer science and programming. Topics include information and data representation, hardware, programming methodology, algorithm design, abstract data types, programming languages, operating systems, applications, and communications.

Prerequisite: Students planning to enroll for this course should have mastered computer essentials including interaction with a graphical user interface, text editor, and Web browser. If you prefer to use your own computer rather than university laboratory facilities, it is expected that you can download, install, and configure software. No experience in computer programming is expected or required.
If you already have some experience in computer programming, even in a language other than C++, you may want to consider bypassing CSCI 151 and starting in CSCI 152 instead.

Student Learning Outcomes:
(Measured by exam, quiz, lab, and homework assignment results)

1.   Show how computer hardware represents information.
2.   Describe the computer circuitry that harnesses the electrical flow.
3.   Explain how computing components may be combined to build computer systems.
4.   Apply general problem-solving strategies to the development of computer algorithms.
5.   Write programs in machine, assembly and high-level languages to express and implement
      Algorithms to solve problems.
6.   Identify and explain the application of abstract data types such as stacks, queues, lists, trees,
      And graphs.
7.   Apply the object-oriented methodology to computer problem solving.
8.   Explain the role of an operating system in managing and interacting with computer system
      Components including main and secondary memory.
9.   Utilize information system software to organize, manipulate, and secure data.
10. Describe ways computer networks are used to communicate and share resources and facilitate
      Web processing.

Recommended Text for Intro to Computer Science:
      Computer Science Illuminated by Nell Dale and John Lewis
      5th edition ISBN 9781449672843
or   6th edition ISBN 9781284055917
or   4th edition ISBN 9780763776466

A copy of the 4th and 5th editions of this book are on reserve in the university library.

Recommended Text for C++ Programming:
C++ Programming: from Problem Analysis to Program Design by D.S. Malik
5th edition   ISBN: 9780538798082
or   6th edition   ISBN: 9781133626381
or   7th edition   ISBN: 9781285852744

A copy of the 5th edition of the Malik text is on reserve in the university library.
Online Resources for C++ Programming:

Some recommended sites for explanations, compiler information, tutorials, sample code, and general how-tos:

C/C++ Programming Resource site   (http://www.cprogramming.com/)
C++ Resource Site   (http://www.cplusplus.com/)
SparkNotes Computer Science Study Guides   (http://www.sparknotes.com/cs/)
Microsoft Visual C++ Resources   (https://msdn.microsoft.com/en-us/vstudio/hh386302)
Microsoft Visual Studio Resources   (https://msdn.microsoft.com/en-us/vstudio/default.aspx)
Essays on Basic Topics in Computer Science   (http://www.aihorizon.com/essays/basiccs/)

Evaluation:

Your grade for the course will be based on the following approximate numbers and percentages:

	Intro to Computer Science (material from "Computer Science Illuminated")
10%	midterm exam (open study guide notes)
10%	final exam (open study guide notes)
	Intro to C++ Programming
30%	4 in-class quizzes (the 4 highest grades from 5 quizzes)
15%	programs (probably 6 -10 small programs)
10%	final exam
10%	weekly lab assignments
15%	Class Project

Letter grades for the course will be assigned according to this scale of percentage points:

90 – 100	A
80 – 89	B
70 – 79	C
60 – 69	D
0 – 59	F

You must earn an A on your own. Assuming that you have completed all assignments, lower borderline grades may be affected positively or negatively by factors such as:
- the class grade distribution
- your class attendance, participation, and behavior (including what should be common courtesies: no sound-producing device use in class, arrive on time, stay until class is over, avoid distracting other students)
- the pattern of your grades (quiz grades getting better throughout the semester or going downhill)

You need to give me a reason to think you deserve a grade higher than your percentage indicates, and you need to show me that you've made every effort to help yourself (at least attempting assignments, correcting programs, etc).

Additions to the syllabus:
Details of **program requirements** will be provided in a separate handout **General Policy for Programming Assignments** which we'll look at when the first program is assigned. Details of the **Social & Ethical Issues** component will also be provided in a separate handout in week 3. Both of these handouts along with the

syllabus and an explanation of **How the Gradebook works** will be available for download from the top-level folder in eCollege Doc Sharing.

Course Policies

Makeups:
If you miss a quiz, that will be the low grade (0 points) which will be dropped. If you know ahead of time that you'll have to miss a quiz, let me know and it may be possible for you to take a quiz outside of class. Taking a quiz *after* the rest of the class does is usually possible only if you can take it before the next class period when the graded quizzes will be returned to the class.

Attendance:
You are responsible for everything covered in all class meetings, whether you're in class or not.

You can earn **extra credit** for your class attendance at the rate of one-tenth of a percentage point for each complete class meeting attended (not including lab sessions). You must arrive on time and remain for the entire class. There are no exceptions for excused absences.

In order to be considered officially present for a class period, you must **sign the roster sheet** which will be passed around at the beginning of class. If you arrive late, it's your responsibility to sign the roster after class before you leave.

Drops:
If you are making an obvious effort in the course at the time you drop (still attending class, attempting program and lab assignments), you may drop passing no matter what your actual grade might be.   If you just disappear, your grade will be whatever you have actually earned at the end of the semester (usually a grade of F). If you find that you are unable to complete the course, please be sure to drop the course to avoid receiving an F; you will not be automatically dropped.

eCollege:

This is a ***web-enhanced*** course (using some of the capabilities of eCollege but not set up as a self-directed online course).

You can access eCollege either directly or through your myLeo account.

1. To access eCollege through myLeo, log on to your myLeo account and click on **eCollege** in the bar in the upper right of the screen under the myLeo logo (there may be an additional link available in the menu on the left side of your screen). You can bypass the university's home page by going to **http://online.tamuc.edu**

2. In eCollege click on the **MyCourses** tab (next to **Home**) in the upper left corner of the screen between the university logo and the date.

3. Scroll down to the bottom of the MyCourses screen and click on the class you want from the list of online and web-enhanced courses you may be taking this semester.

eCollege Announcements:
The course home page (the first page once you get into a specific course) contains an **Announcements** box. There is an announcement titled **What's Happening in Class** in which you will find information about each class meeting (what's due, what we did today, and what we'll probably be doing for the next class) in case you missed a class or forgot to write down an assignment. Click on an announcement name (or on the + sign to the left of the name) to open and view it. The most recent class period will always be at the top, but all previous class periods will remain in the announcement and can be viewed by scrolling down in the document.

Please be sure to check the eCollege Announcements frequently.

eCollege Document Sharing:

Input data files, sample programs, and most handouts can be downloaded from eCollege. All of these files can be found in the Document Sharing section. Click on **Doc Sharing** in the toolbar at the top of the screen. The files will be organized according to topic; files not associated with a particular topic will be listed under the default category of the course name.

In the **Categories** box, click on the category (folder) you want. Then in the box below (title is the category selected), click on the name of the file to be downloaded. You're given the option of saving the file or opening it directly in its application program (most are Word 2003 files, so the document will be opened in Word) and then saving it if you wish to keep a copy of it.

Compilers: All the computers in the Computer Science labs in Jour 101-102 and 200 have at least three C++ compilers installed: Microsoft Visual C++, Microsoft Visual C++ Express Edition (a more limited but possibly easier-to-use version of Visual C++), and Orwell Dev C++. Any C++ compiler you may have access to is fine for programming assignments for this course.

All of these compilers can be downloaded for free for installation on your own computer:

Orwell DevC++ 5.11 – click on the dark green rectangle labeled **Download**
http://sourceforge.net/projects/orwelldevcpp/

Visual Studio Express 2013 (includes C++, C#, Visual Basic, JavaScript, Visual F#):
http://www.microsoft.com/express/vc/

Visual C++ Developer Center (links for how-to videos, tutorials, etc. in addition to downloading):
http://msdn.microsoft.com/en-us/visualc/default.aspx

For some Visual Studio Tips and Tricks, go to:
http://www.cprogramming.com/visual.html

The entire Visual Studio (of which **Visual C++** and the express edition are a part) is available for free download by students enrolled in computer science courses at A&M Commerce. If you don't already have an account from enrollment in a computer science course in a previous semester, an account for downloading free Microsoft products will be set up for each of you after the 12th class day (when enrollments are stabilized).

Not on Computer Science lab computers, but available for free for your own computer:

Code::Blocks for **Windows** users: **http://www.codeblocks.org/downloads**

Code::Blocks for **Linux** users: **http://www.cprogramming.com/g++.html**

XCode for **Mac** users: **http://www.cprogramming.com/xcode.html**
You already have XCode on your system disk – the link is for help in installing it.

XCode can also be used with the Code::Blocks interface. See the file CodeBlocks for Macs in the top-level folder in eCollege Doc Sharing.

TENTATIVE SCHEDULE

Each week we'll be looking at a chapter in the textbook and also learning the basics of programming using the C++ computer programming language. Each chapter (except for Ch 1) in "Computer Science Illuminated" will have a quiz . The C++ programming quizzes will be taken in class.

Week	Date	Computer Science Illuminated	C + + Programming Activity
1	31 Aug – 4 Sept	Course overview Introduction to computer systems and programming concepts	Logic and programming
2	**7 Sept (Mon)** 8-11 Sept	Labor Day holiday – no class Monday Ch 1 The Big Picture (Computing Systems, Computing as a Tool and a Discipline)	Compiler demo
3	14-18 Sept	Ch 2 Binary Values and Number Systems	Basic elements of C++ Program coding and testing
4	21-25 Sept	Ch 3 Data Representation	Data types and identifiers
5	28 Sept – 2 Oct	Ch 4 Gates and Circuits	Interactive input and output **Quiz 1**
6	5-9 Oct	Ch 5 Computing Components	Arithmetic operators and expressions
7	12-16 Oct	Ch 10 Operating Systems	Decision structures (if statements)
8	19-23 Oct		Decision structures (if statements) **Quiz 2**
9	26-30 Oct	**Midterm exam** over Ch 2-5, 10	While loops
10	2-6 Nov **5 Nov (Thu)**	Ch 11 File Systems and Directories Last day to drop a class and remain enrolled in other classes	While loops Intro to the Class project
11	9-13 Nov	Ch 12 Information Systems	Value-returning functions **Quiz 3**
12	16-20 Nov	Ch 15 Networks	Value-returning functions
13	23-24 Nov **25 Nov (Wed)**	Ch 16 The World Wide Web **Thanksgiving** holiday begins at noon No class for MW section	Void functions
14	30 Nov – 4 Dec	Ch 16 The World Wide Web	Group variables (arrays) **Quiz 4**
1 5	7-11 Dec	**Final Exam over Ch 11, 12, 15, 16: MW: Wed 9 Dec TR: Thurs 10 Dec**	Group variables (arrays)
16	14-18 Dec **Final Week**		**Final Exam: MW: Wed 16 Dec 1:15-3:15 TR: Tues 15 Dec 4:30-6:30**

Some recommendations for a successful semester:

1) **Be here** as often as possible.

2) **Read assignments** and be ready for what we'll be talking about in class.

3) **Ask** if you don't understand something.

4) **Get help** (sooner rather than later) if you have problems:
   - dept lab help in Jour 200 and possibly Jour 101-102
   - the Academic Success Center also provides tutoring in the library for a wide variety of subjects
   - make friends with at least one person in class so you can compare notes or check for anything you might have missed
   - get a study group together
   .

5) **Stay caught up** as much as possible.

6) **Get started** on programs so that you have time to get help if you find you need some help.

7) **Do your own work.** Consult with others about problem-solving strategies, but **code it yourself**.

8) What you get out of any class depends to a very large degree on what you're willing to put into it. Get in the habit of writing little practice programs to try out new language features as we learn them. As you write more programs (even small ones), the process becomes easier, you're much more likely to remember how the language works, and you get much better at programming logic (the hardest part of computer programming).

9) Know your own limits and don't over-extend yourself any more than necessary.

University Policies and Announcements

"All students enrolled at the University shall follow the tenets of common decency and acceptable behavior conducive to a positive learning environment."
(See Student's Guide Handbook, Policies and Procedures, Conduct)

EARLY INTERVENTION FOR FIRST YEAR STUDENTS:

Early intervention for freshmen is designed to communicate the University's interest in their success and a willingness to participate fully to help students accomplish their academic objectives. The university through faculty advisors and mentors will assist students who may be experiencing difficulty to focus on improvement and course completion. This process will allow students to be knowledgeable about their academic progress early in the semester and will provide faculty and staff with useful data for assisting students and enhancing retention. Grade reports will be mailed by the end of the sixth week of the semester.

All students should be aware that plagiarism is a serious offense. This is true not only of written essays but also of work written in computer languages such as C++. Copying code for assignments from other students or the internet is not allowed. You may certainly discuss with one another the general aspects of programming assignments (like "what does this requirement mean?") and strategies for coding solutions for these assignments, but you must write the actual code for the programming assignments on your own.