Course Syllabus – Fall-2016

Course Information

Course number: CS/CE/TE 1337 (Sections 503 and 505)
Course Title: Computer Science I
Credit Hours: 3
Class Schedule: Sections 505: Mon. & Wed. 5:30pm - 6:45pm.
                      Sections 503: Mon. & Wed. 7:00pm - 8:15pm.

Classroom: Sections 505: JO 3.516.
                      Sections 503: ECSS 2.305.

Instructor Information

Name: Dr. Ebenezer Oladimeji (Dr. O.)
Email: eao015100@utdallas.edu
Office: ECSS 4.403
Office Phone: (972) 883-4523.
Office Hours: Wednesday 3:30pm – 5:00pm, or by appointment.

TA/Grader: TBA

DO YOU NEED ASSISTANCE?

E-mail: The best way to reach me is via e-mail. I make every effort to respond within 24 hours. Please include your course and section either in the subject or the body of your e-mail (preferably on the first line if not in the subject). This will help me to address your e-mail as quickly as possible.

Help Desk: For help with issues regarding your computer, UTD maintains a walk-in help desk. Visit their Web site for details: http://www.utdallas.edu/ir/helpdesk/

Tutoring: For programming assistance in CS1336, please visit me, the TA, or the Mentoring Center. The schedule for the Mentoring Center will be released within the first week of classes. Once the Mentoring Center schedule for this semester has been released, an announcement will be posted on eLearning. If you need help, please make the effort to reach out. We can't help you if we don't know that you need help.

Resources: C++ language tutorial http://www.cplusplus.com/files/tutorial.pdf
        C++ reference: http://www.cppreference.com
        C++ tutorial http://www.learncpp.com/

WHAT DO I NEED FOR CLASS?

- **Notebook** – You are going to take a lot of notes and have quizzes. Bring paper to write on.

- **Writing Tool** – pen, pencil, crayon, etc. It's hard to take notes without one. You are free to use your own blood, but that may get messy.

- **Textbook**: Starting Out with C++, From Control Structures through Objects
  (8th edition – orange slice) ; Gaddis, Tony; Pearson Publishing ISBN 0-13-376939-9

  o As you read the text, watch the corresponding VideoNotes. The VideoNotes are available at http://www.pearsonhighered.com/gaddis/.

  o **NOTE**: VideoNotes are only available if your book comes with an access code. If your book does not have an access code, you can buy one online at the above address. The access code is not required for class, but some of you may find the material accessible with this code to be a good resource.

  o Students may use the 7th edition of the book. In doing so, students accept the responsibility of verifying page numbers for assignments as well as learning the C++ 11 topics not present in the 7th edition. I will not copy any information from the 8th edition for students, although you are free to take pictures of pages from my book during office hours

- **C++ Compiler (Required)**
  o All projects you submit will be compiled with MinGW 4.9.2. You may use any IDE that can utilize MinGW 4.9.2.
  o For Windows users, I recommend that you use the Code::Blocks 16.01 IDE. This is a free download for Windows. http://sourceforge.net/projects/codeblocks/files/Binaries/16.01/Windows/codeblocks-16.01mingw-setup.exe. This download includes the IDE and MinGW 4.9.2.

  o For Mac users, I recommend using xCode or creating a Windows partition to install MinGW and an IDE. Be advised that there is a Mac version of Code::Blocks, but it has been reported that it doesn't work with newer versions of the OS X operating system.

  o If a student uses a compiler other than MinGW 4.9.2 for development, he/she is responsible for verifying prior to submission that the code compiles properly with the stated compiler that will be used for grading (i.e. MinGW 4.9.2)

  o If you intend to use your own computers to write the class assignments, it is important that you get a compiler downloaded, installed, and running on your computer as soon as possible. If you don't have a computer, or if you're having problems getting a compiler installed, you should write your programs in the labs until the problems are resolved. Since there are many computers available on campus, problems with your local machines will not be accepted as an excuse for not doing the assignments or late submissions.

REQUIRED COURSE INFORMATION SECTION

Course Prerequisite: CS 1336 with a grade of C or better or equivalent.

Description of Course Content: Review of control structures and data types with emphasis on structured data types. Applies the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design. Includes basic analysis of algorithms, searching and sorting techniques, and an introduction to software engineering.

Student Learning Outcomes: After successful completion of this course, the student should have an:
- Ability to use single and multi-dimension arrays.
- Ability to implement linear and binary searches.
- Ability to implement simple sorting algorithms.
- Ability to implement structured data types.
- Ability to define and implement a class.
- Ability to use fundamentals of object-oriented design.

WHAT I EXPECT OF EACH STUDENT

- **Ask for help.** Email me or stop by during office hours. I want you to succeed. I would rather point you in the right direction so that you can complete an assignment instead of you remaining quiet and failing an assignment.

- **Ask questions any time!** During lecture, before/after class, during office hours, at 2 AM after a night clubbing the day before something is due, etc. I really mean any time. I will respond as soon as I can.

- **Take responsibility for your education**. I will treat this course as similar to a professional setting as I can. I do not plan to bore you with slides and expects you to memorize. I will teach by creating program examples in class. I will give challenging assignments to push you toward learning the intricacies of C++. Part of being a professional is learning how to teach yourself. I am going to guide you through the topics of the semester, but a significant portion of what you take with you to the next class will be things that you learned on your own.

- **Practice time management skills**. All assignments (homework and projects) are designed to be worked on over a period of days or weeks. I expect that you will work on the assignment a little at a time rather than waiting until a day or two before it is due. Those that procrastinate will find this class to be much harder than it should be and will face the risk of below average grades.

- **Attend every class**. Not only might you miss essential words of wisdom, you might miss a quiz as well. Make mistakes! This is how you learn how to do something. Don't be discouraged when something goes wrong. Programming takes lots of practice and mistakes will always happen. Study the mistakes you made so that you can learn the correct way to do it.

- **Read the chapter before the corresponding lecture** (see class schedule below). As part of class lecture, I will write programs that help illustrate the topics mentioned in each chapter. If a student doesn't have minimal knowledge of the concepts that will be covered for that chapter (which are gained by reading the chapter), it will be harder to get a deeper connection to what we are accomplishing in class.

- **Bring your textbook to class**. If you bought the physical version, I know it is heavy and you would rather leave it at home to collect dust. However, we will refer to the book frequently in class. Your book wants to be a part of your academic experience. Don't prevent your book from having an adventure with you.

- **Arrive to class on time and remain in class until dismissed**. Arriving late and leaving early cause disruptions to the other students in the class and to me. Should you need to leave early for a valid reason, please notify me in advance and sit near the door to limit the disruption. Repeat offenders will be penalized by replacing a previous quiz grade with a zero.

- **Don't sleep in class**. Let's be honest; programming in C++ is not the most exciting topic. Combine that with fatigue from late night gaming and/or study sessions and it is super easy to doze off. Fight it off. Bring in a caffeinated beverage of your choice, (Coffee or a Monster energy drink. Carry an emergency bottle of 5 Hour Energy in your backpack if need be. If students could learn C++ by sleeping, there would be no reason to get out of bed to attend class.

- **Don't pack up your things until class is over**. Most of the time we will go until the very last minute before ending class. Sometimes we might go over by a minute or two if I need to finish a discussion. If students start packing up before we are finished, it makes a lot of noise. That noise might prevent someone from hearing very crucial information such as what the next homework assignment is and when it is due. It also makes me think you are in a hurry to leave the awesome fun party we are having and hurts my feelings.

- **Computers in class**. You may bring your laptops to class to take notes. Many students like coding along with the instructor in class, however, this can potentially become a detriment to the student. While students are copying the code from the whiteboard, they are not concentrating on the logic or details behind the code. It is the logic and ideas behind the statements that are more important. All code written in class will be posted in eLearning after it is completed.

Departmental Policy on Class Attendance

The CS/CE department has a policy on class attendance: "Three consecutive absences lead to one letter grade drop. Four consecutive absences lead to an F." This policy can only be waived only when there is a legitimate reason for absence (e.g. sickness, university games, etc.), with convincing proof. Please note that a Doctor's note from a foreign country is not convincing enough.

THE INFORMATION YOU REALLY CARE ABOUT

Grading Scale:

98-100 A+	88-89 B+	78-79 C+	68-69 D+	Below 60 F
92-97 A	82-87 B	72-77 C	62-67 D	
90-91 A-	80-81 B-	70-71 C-	60-61 D	

The instructor reserves the right to adjust the average required to receive a particular letter grade.

Grade Components:

       Pre-Test: 5%
       Projects 1-5: 25% (5% each of 5 projects)
       Exam 1:   20%
       Exam 2:   20%
       Exam 3:   25%
       Others: 5% (In-Class Quizzes, Attendance, etc.)

Projects: Projects will be major programming assignments that supplement recently discussed topics and should be completed in two to three weeks. Projects are intended to take approximately 15-20 hours to complete; this includes the design, coding and testing process. Waiting until a couple of days before the due date to start the project is a bad idea. Not only does this introduce unnecessary stress into your life, it hardly ever ends well for the student. Most students score poorly on projects that are built in less than three days.

Projects are individual endeavors and students are not to work in groups on any project. Students are permitted (and I openly encourage students) to discuss the project. Feel free to share ideas on the logic, but **DO NOT SHARE ANY CODE**. All projects will be submitted in eLearning and will be compared for originality. Any projects that are approximate or identical copies will be reported to Judicial Affairs and I will accept their decision in regards to the grade if they believe that academic dishonesty has occurred.

Programming assignments will be graded on a 100 point basis. Not only will your project be graded on proper execution, but also things like efficiency, implementation and documentation. Keep in mind that you always want to write code that is easy to understand and is also easy to maintain. Fewer lines do not necessarily mean a better program. Please use comments liberally.

You are responsible for testing your project thoroughly before submission. I will not give you the test cases that will be used for grading before the project is due.

Quizzes: Quizzes may be given in class and are generally unannounced. No make-up quizzes will be given. Quizzes missed for an excusable reason (with valid documentation) will be exempted.

Tentative Course Calendar

Date	Class Material	Assignments/Due Dates
22-Aug-16	Intro to CS 1337, Intro to Code::Blocks/xCode	Review Chapters 1-8
24-Aug-16	Intro to C++, Arrays and Searching	
29-Aug-16	Searching and Sorting Arrays	Read Sections 12 (omit 12.7, 12.8)
31-Aug-16	Searching and Sorting Arrays Advanced File I/O	
5-Sep-16	**LABOR DAY (No Class)**	
7-Sep-16	Advanced File I/O	
12-Sep-16	Advanced File I/O	Read Chapter 19
14-Sep-16	Recursion	**Project 1 Due (9/15)**
19-Sep-16	Recursion	
21-Sep-16	**Exam 1 (Chapters 8, 12, 19)**	Read Chapter 9
26-Sep-16	Pointers	
28-Sep-16	Pointers	
3-Oct-16	Pointers	Read Chapter 11;
5-Oct-16	Structured Data	**Project 2 Due** Read Chapter 17.1 and 17.2
10-Oct-16	Structured Data	
12-Oct-16	Structured Data	
17-Oct-16	Enumerated Data Types, Unions	Read Chapter 10
19-Oct-16	Characters, Strings and the String Class	
24-Oct-16	Characters, Strings and the String Class	
26-Oct-16	Characters, Strings and the String Class	**Project 3 Due**
31-Oct-16	**Exam 2 (Chapters 9-11, 17.1, 17.2)**	
2-Nov-16	Introduction to Classes	Read Chapter 13;
7-Nov-16	Introduction to Classes	
9-Nov-16	Introduction to Classes	
14-Nov-16	More About Classes	3
16-Nov-16	More About Classes	**Project 4 Due** Read Chapter 14
21-Nov-16	**FALL BREAK (No Class)**	
23-Nov-16	**FALL BREAK (No Class)**	
28-Nov-16	More About Classes	
30-Nov-16	Inheritance, Polymorphism & Virtual Functions	Read Chapter 15
5-Dec-16	Inheritance, Polymorphism & Virtual Functions	
7-Dec-16	Inheritance, Polymorphism & Virtual Functions	**Project 5 Due Dec-10**
	Final Exam 3 (Chapters 13, 14, 15)	

Please note that the instructor reserves the right to modify this calendar as she deems necessary.
Please see eLearning for discussions/announcements regarding changes to the calendar.

Important Dates:

August 22 Classes start
September 5 Labor Day
September 7 Census Day
September 15 Project 1 due
September 21 Exam 1
October 5 Project 2 due
October 26 Project 3 due
October 27 Last Day to Withdraw
October 31 Exam 2
November 16 Project 4 due
December 10 Project 5 due
TBA (Finals Week) Exam 3

University Policies

For all other University policies, please visit http://provost.utdallas.edu/syllabus-policies/