CIT163: Intro to Programming: C++
Course Syllabus – Spring 2017, TR 9:30 AM – 10:45 AM, ML 119

Instructor : Brian M. Morgan

Office : Morrow 114

Office Phone Number : (304) 696-6469

Office Hours : MWF: 8:30a – 10:30a
Other times by appointment ONLY please. My schedule is very busy during the semester, so please try to schedule appointments as far in advance as possible. It is often very difficult to set up appointments less than 24 hours in advance.
You can always search the hash tag #CIT163 on Twitter as I will update what we cover each day in class this semester: http://twitter.com/brianmmorgan/

E-Mail : brian.morgan@marshall.edu

Text : (304) 634-6736 (include who you are first text)

University Policies : By enrolling in this course, you agree to the University Policies listed below. Please read the full text of each policy by going to http://www.marshall.edu/academic-affairs/policies/
Academic Dishonesty / Excused Absence Policy for Undergraduates / Computing Services Acceptable Use / Inclement Weather / Dead Week / Students with Disabilities / Academic Forgiveness / Academic Probation and Suspension / Academic Rights and Responsibilities of Students / Affirmative Action / Sexual Harassment
E-mail Info: http://www.marshall.edu/muonline/e-mail/

Textbook:

The following textbook is required for the course:

> **C++ Programming: From Problem Analysis to Program Design**, 7th edition, by D.S. Malik; Course Technology; ISBN: 978-1-285-85274-4, 2015. E-Book Available from CengageBrain.com

Computer Requirements:

Supplemental materials can be found contained within the Blackboard Learn environment (http://www.marshall.edu/muonline/). I will be sending class announcements, updates, etc. using your Blackboard account (will discuss during the first lecture if necessary). Access to a web browser is required, as is Adobe Acrobat Reader (http://get.adobe.com/reader/), and Visual Studio Enterprise 2015 (or higher). This software package is available for free to students in this course (see http://www.marshall.edu/cos/software/ for specifics).

Course Description:

Concepts of software development and maintenance using C++, including syntax of the language, loops, functions, classes, decision structures, and file processing. Proper program design using object-oriented programming techniques are emphasized.

Credit:

The course is three (3) credit hours. It includes classroom lectures, exams, and various programming projects assigned as homework assignments. Students will participate in various aspects of projects that illustrate the implementation of concepts in general applications.

Pre/co-requisites:
N/A

Desired Objectives/Outcomes:
By the end of this course, you should be able to:

Course Student Learning Outcomes	How Practiced in this Course	How Assessed in this Course
Discuss Object-Oriented programming concepts	In-class examples, discussions, Chapter 10 examples	Project 9
Demonstrate basic console programming skills using C++	In-class examples, discussions, Chapters 2 through 10 examples	Projects 1 through 9 Final Exam
Develop software applications using Microsoft Visual Studio 2015	In-class examples, discussions, Chapters 2 through 10 examples	Projects 1 through 9 Final Exam
Demonstrate planning techniques for developing software applications	In-class examples, discussions, Chapters 1 through 10 examples	Projects 8 through 9 Final Exam
Work through programming logic	In-class examples, discussions, Chapters 2 through 10 examples	Projects 1 through 9 Exams 1 and 2 Final Exam

Instruction method:
There will be 2.5 contact hours of classroom lecture per week. Projects covering major topics are part of the course with lectures kept to a minimum. Content from each of the chapters will be enforced through programming projects and examinations, including a comprehensive take-home final exam. Students may work on their assignments/projects in University computing facilities or from home with an Internet connection and Visual Studio Enterprise 2015 installed on their PC (or XCode on a Mac).

Evaluation method:
Evaluation of student's performance will be based on the quality of their performance on homework assignments (programming projects) and exams.

Grading Policy:

2 in-class Exams (equally weighted)	300 points
Final Exam	200 points
9 Programming Projects (equally weighted)	500 points
Attendance	0

Assessment of Projects:
The grading of all homework assignments and projects will take into account:

1. Although the most important attribute of a program is correctness, grading will take into consideration such items as time and coding efficiency, **documentation**, etc.

2. Programs must have proper inline documentation and must be properly indented. Up to 20% will be deducted for poorly documented and/or poorly indented code.
3. All submitted code must compile to receive at least partial credit. Code that does not compile will receive 0 credit, **NO EXCEPTIONS**. This means you must debug your code before submitting.
4. When a method name and/or parameters are specified in an assignment's description, you must use that name and/or parameters and ensure the function works for all possible inputs (test!!).
5. Although interactions with other students are encouraged, you **must** compose your own answers, unless otherwise noted.

Individuals who utilize other people's thoughts or ideas must provide appropriate references to said resources, including any and all web resources consulted. Failure to provide such documentation will result in a failing grade for the assignment, and may result in a failing grade for the course.

Final letter grades are determined based on the following grading scale:

895 – 1000 points	A
795 – 894 points	B
695 – 794 points	C
595 – 694 points	D
Less than 595 points	F

The instructor reserves the right to change these values depending on the overall class performance and/or extenuating circumstances. Incompletes are rarely given and only in the case of a true emergency. I am unwilling to assign an Incomplete grade for students who find they have fallen behind on assignments, missed a test, or taken on too much in terms of schooling or work responsibilities. For these situations, if you are still within the university deadlines, dropping the course may be more appropriate.

Policy Statement:
My Academic Dishonesty Policy

Academic Dishonesty is defined as any act of a dishonorable nature which gives the student engaged in it an unfair advantage over others engaged in the same or similar course of study and which, if known to the classroom instructor in such course of study, would be prohibited. Academic Dishonesty will not be tolerated as these actions are fundamentally opposed to "assuring the integrity of the curriculum through the maintenance of rigorous standards and high expectations for student learning and performance" as described in Marshall University's Statement of Philosophy.

If you are found cheating on projects or plagiarizing answers from the Internet or other sources (among other things), **there will be no second chance**. Your penalty is that you will receive a failing grade for the course. In those cases in which the offense is particularly flagrant or where there are other aggravating circumstances, additional, non-academic, sanctions may be pursued through the Office of Judicial Affairs. Notice of an act of academic dishonesty will be reported to the Department Chair, Dean of the College of Science, and to the Office of Academic Affairs. Please refer to the Marshall University Undergraduate Catalog for a full definition of academic dishonesty.

Assignments*:* The course includes a number of assignments/projects. All assignments are due **BY THE BEGINNING OF CLASS** on their due date and must be submitted through the Blackboard Assignments tool. **NO LATE ASSIGNMENTS WILL BE ACCEPTED**. Homework deadlines will not be extended because you waited too late to start or because you did not allocate enough time before the deadline to work on it; likewise, they will not be extended because of hardware or network failure. If you wish to receive any credit for an assignment, you must submit whatever you have done, even if it is incomplete, by the deadline. Partial credit is usually preferable to no credit.

Please do not procrastinate in working on your assignments or trying to submit through Blackboard as many others have done in the past. If you wait until the night before a project is due to start on the project or the last minute to submit, chances are, you will fail. If you leave an assignment for the last minute and there are technical problems, you still must turn in whatever you have by the deadline.

You may submit multiple versions of assignments before the deadline as I will grade the latest pre-deadline submission. This is to encourage you to turn assignments in early (since you will know that you can always turn in an improved version if further inspiration arises).

Exams*:* There are THREE exams worth 50% of your overall grade. The first will come after Chapter 5, the second after Chapter 10, and a comprehensive take home Final exam (as scheduled). Exact dates and times of exams will be announced in class.

Make-up Exams and Late Penalty*:* Make-up exams will not be given, except under unusual circumstances and with satisfactory written justification. Any student who misses an exam due to an unexcused absence will receive a grade of 0 for that exam with no opportunity for make-up or substitution. University excused absences or those occurring with a good reason (and that reason must be given prior to missing the exam– call and leave a message if you have to) will be excused. Make-up exams must be taken within one week of the original scheduled date. The decision whether to give a make-up exam rests with the instructor.

Attendance Statement:

Regular attendance is expected. If you miss class, it is your responsibility to catch up on material missed, and it will not be my responsibility to catch you up on material missed during office hours, or re-lecture to you. "I wasn't there that time" is never an acceptable excuse. Please attempt to come to class on time, with your headphones put away and your cell phones turned off. If you must arrive late or leave early, please do so with the least possible distraction to other students.

Withdrawal Policy:

The University withdrawal policy is followed in this course. The last day to drop an individual course for the Spring Semester is March 17, 2017.

University Holidays:

The class is officially dismissed on the following dates:

Spring Break	March 21, 2017
	March 23, 2017

Topics and Methodology:

The following outline delineates the tentative class schedule with topics to be addressed during the course. Please note this is a tentative schedule and it may change upon class progress:

January 10	Overview of course and syllabus, introduction to Blackboard, how to create a C++ project in Visual Studio Enterprise 2015

January 12	Chapter 1
January 17	Chapter 2
January 19	Chapter 2
January 24	Chapter 2
January 26	Chapter 2
January 31	Chapter 3
February 2	Chapter 3 Project 1 Due Chapter 2, Page 121, Programming Exercise #19 Chapter 2, Page 122, Programming Exercise #25
February 7	Chapter 3
February 9	Chapter 4
February 14	Chapter 4 Project 2 Due Chapter 3, Page 183, Programming Exercise #5 Chapter 3, Page 183, Programming Exercise #6
February 16	Chapter 4
February 21	Chapter 5
February 23	Chapter 5 Project 3 Due Chapter 4, Page 257, Programming Exercise #7 Chapter 4, Page 260, Programming Exercise #17
February 28	Tomato Day/Review for Exam 1 Project 4 Due Chapter 5, Page 340, Programming Exercise #12 Chapter 5, Page 340, Programming Exercise #13
March 2	Exam 1
March 7	Chapter 6
March 9	Chapter 6
March 14	Chapter 6
March 16	Chapter 7
March 28	Chapter 7 Project 5 Due **All assignments from this point forward *MUST* use functions** Chapter 6, Pages 453-454, Programming Exercise #8 Chapter 6, Pages 455-456, Programming Exercise #14
March 30	Chapter 8
April 4	Chapter 8 Project 6 Due Chapter 7, Pages 517-518, Programming Exercise #7 Chapter 7, Page 518, Programming Exercise #9
April 6	Chapter 8/9

April 11	Chapter 9 Project 7 Due Chapter 8, Page 602, Programming Exercise #4 Chapter 8, Page 603, Programming Exercise #7
April 13	Chapter 9
April 18	Chapter 10
April 20	Chapter 10 Exam 2 Review Project 8 Due Chapter 9, Page 647, Programming Exercise #5 (will need to read and perhaps complete exercises 3 and 4 first) **HINT**: You may modify the composition of the struct initially mentioned in Exercise 3
April 25	Exam 2
April 27	Q/A Project 9 Due Chapter 10, Pages 730-731, Programming Exercise #4 Receive Take Home Final
May 2	Final Exam Due 8:00 am

For each topic discussed in the textbook, specific experience of other students and the instructor will be discussed to enhance the characteristics involved. Hands-on projects for the course will be based on creating databases for either real-world or fictitious needs. Additional material may also be covered in the class.

Every student is responsible for all material presented in class, including lectures, notes, and handouts. In the case you are not present for a class session, it is your responsibility to retrieve information about the material presented in that class. Class attendance is very important.

Effort Required:
As a 100-level course, this course is provided as an introductory course, but there will still be a considerable amount of development and research effort required of the student, especially since the technologies/practices used in the course build upon each other. With programming, you cannot start learning the different languages we will be using at different tiers without practice. This means you may have to play around with in-class examples, experimenting to see if something you are curious about really works like you think, doing further research on topics of interest, and so on. Programming courses can be notorious time eaters. Occasionally, a problem with code will take large amounts of time to locate and fix.

For every one hour in class, the student is expected to put in an effort of at least 2-3 hours outside the class for studying and completing assignments and projects. Upon background and preparedness, some students may have to put in additional effort. **PLEASE DO NOT PROCRASTINATE**. Procrastination and the placing of blame on other factors than yourself have become very large problems in college classes, and is often a bad approach to life. Prioritize, schedule, and take responsibility for your actions and you should do very well in this class. Starting early enough so that you have time to ask me questions when you run into problems can help with this (why spend 4 hours struggling with a frustrating roadblock the night before the assignment is due, when you can spend 10 minutes composing an e-mail early in the week, work on other problems while waiting for the answer, and then get a reply that makes everything clearer as soon as you read it?)

A Successful Student will:

- Attend every lecture
- Participate in class (asking questions, paying attention, taking notes, being attentive)
- Complete reading assignments in a timely fashion.
- Practice and "play" with posted examples.
- Ask specific questions -- in class, in lab, in office-hours, and in e-mail
- Read through each homework assignment as soon as it is posted
- Start working on each homework assignment early
- E-mail me with specific homework-related questions starting early in the week both to clarify what a question is asking for and when hitting roadblocks (being sure to include both the code involved and any error messages or descriptions of odd behavior)
- Always submit SOMETHING for an assignment, even if it is not complete
- Study with others for exams, practice explaining concepts to one another.
- Attempt every exam problem, and carefully study exams when they are returned.
- Practice programming at the different levels as much as possible

You are encouraged to ask me questions in class, in office hours, and by e-mail. The most successful students are those who are not afraid to ask questions early and often, who do the assigned reading, who attend lecture regularly, who start homework promptly after the required topics are covered in lecture, and who practice course concepts as much as possible.

It is better to ask a question sooner than later -- for example, it is better to send an e-mail with a specific question as soon as you think of it than it is to wait a day or two until the next class meeting or office hour. If you wait to ask such questions, you may not have time to complete an assignment. It is not a problem if you end up sending me several questions in separate e-mails (as you work on different parts of a project while awaiting earlier answers). That being said, I expect you to ask specific questions as overly vague or broad questions are very problematic. An example of an overly vague or broad question is: "Here's my assignment. Is it right?" I will not simply tell you if something is right. Be specific.

Communication:

The Discussion Tool within Blackboard and your MU E-mail account will be used to make any general announcements, last minute changes, etc. It is **advised** that you monitor your e-mail and Blackboard course discussions posts and E-mails at least once a day.

Note about cell phones in class:

In compliance with Marshall University’s cell phone policy, please set your cell phone ringer to "Vibrate Only" mode (or turn it off) before you enter the classroom. If I hear it ring in class, or vibrate excessively on your desktop, I get to answer it -> no exceptions.