MONTGOMERY COLLEGE
Rockville Campus
Engineering, Physical and Computer Sciences Department
CMSC204 Computer Science II

Instructor Information

Name:
Mailbox Location:
Email:
Office Location:
Office Phone:
Office Hours:

Course Information

Semester:	Course CRN:
Class starts:	Class ends:
Midterm Exam 1:	Final Exam:
Midterm Exam 2:	
Check MyMC class schedule for your Specific Deadline to Drop without a grade W or to change from audit to credit or from credit to audit.	Check MyMC class schedule for your Specific Deadline to drop a class with a W grade.
Online via Blackboard and MC Linux Server	Check MyMC class schedule for your Specific Refund Deadlines.

Course Description

Builds on concepts introduced in CMSC 203 , emphasizing writing larger programs and designing and implementing classical abstract data types such as list, stack, queue, binary search tree, graph, priority queue, hash table. Topics include string processing and recursion; data abstraction, encapsulation, and structure implementation; object-oriented program design; specification, implementation and application of these traditional ADTs. The course also emphasizes dynamic memory allocation, search and sorting algorithms, and introduces algorithm complexity. Designing and implementing advanced-level programming assignments are an integral part of the course.
PREREQUISITE(S): *A grade of C or better in* CMSC 203*.*
PRE- or COREQUISITE(S): MATH 182*. Four hours each week. Formerly CS 204.*

4 semester hours

Prerequisites: A grade of C or better in CMSC 203.

PRE- or COREQUISITE: MATH 182.

Course Outcomes

#	Upon course completion, a student will be able to:
1.	Demonstrate basic principles of program development and design.
2.	Contrast basic concepts of procedural and object-oriented programming.
3.	Utilize fundamental features of a higher-level language, including event-driven programming, graphical user interface, multi-threading, exceptions, and error-handling.
4.	Implement abstract data types, such as list, stack, queue, priority queue, binary search tree, graph, and heap.
5.	Describe the design and time complexity of algorithms.

Course Materials

Data Structures and Abstractions with Java, Frank M. Carrano and Timothy M. Henry, Pearson, 5 Edition
Textbook and other materials may be purchased through the bookstore

Eclipse and Java JDK can be downloaded from internet.

Grade Basis

Final Examination	30%
Midterm Exam (2)	30%
Programming Assignments/Projects	25%
Labs	15%
Total:	**100%**

Grading Scale:

90 - 100%	A
80 - 89%	B
70 - 79%	C
60 - 69%	D
Below 60%	F

General Class Policies

- You are responsible for all work missed, and for meeting assignment due dates when absent. Please call or email your instructor if you are going to be late or absent.
- You are strongly encouraged to contact your instructor at home by phone or e-mail if you are having difficulties, or have any questions about assignments.
- All assignments are expected to be the result of your own efforts, not the collaboration with others. Plagiarism or turning in an assignment which is essentially identical to that of another student will result in a zero for that assignment, with no opportunity to make up the grade.
- Please include your name and the course information in the submitted assignments.
- There is always a means to submit your assignments on time. Be creative, be persistent, and keep your instructor informed!
- All assignments must be turned in on or before the due dates to receive full credits.
- **Missed Tests, Quizzes, and Discussions**. As a rule: NO MAKEUPS without a doctor's excuse. If the final exam is not taken, the student will receive a grade of F for the course.

Course Outline

Topics
Software Engineering Principles (Design and Verification)
Data Design and Implementation (Built-In, Abstract Data Types, Classes)
Encapsulation, Inheritance and Polymorphism
Collections:
• List ADT (Sorted and Unsorted)
• Vectors
• Stack and Queue ADTs
• Linked Structures
• Recursion
• Binary Search Tree ADT
• Heap and Priority Queue ADTs
• Graph ADTs
Sorting and Searching Algorithms
Java networking using sockets
Multithreading