CENTRAL TEXAS COLLEGE  
COSC 2336  
PROGRAMMING FUNDAMENTALS III  
Semester Hours Credit: 3

INSTRUCTOR:__________________  
OFFICE HOURS:__________________

I. INTRODUCTION

A. Further applications of programming techniques, introducing the fundamental concepts of data structures and algorithms. Topics include data structures (including stacks, queues, linked lists, hash tables, trees, and graphs), searching, sorting, recursion, and algorithmic analysis. Programs will be implemented in an appropriate object-oriented language. (This course is included in the Field of Study Curriculum for Computer Science.)

B. This course serves as a required or elective course on various degree plans. Curriculum plans for degrees and certificates are listed in the current Central Texas College catalog.

C. This course, in conjunction with the other courses listed on the degree plan, serves as preparation for careers in computer programming.

D. This course may be delivered as a traditional lecture/lab, non-self-paced online, or blended course.

E. Prerequisites: COSC 1337 Programming Fundamentals II, which cannot be taken concurrently.

II. LEARNING OUTCOMES

Upon successful completion of this course, the student will be able to:

A. Design and develop programs that implement basic data structures, including stacks, queues, linked lists, hash tables, trees, and graphs. (C1, C7, C8, C16, C18, F1, F3, F8)

B. Apply recursive techniques and algorithms to solve problems. (C5, C6, C8, F2, F11)

C. Implement searching and sorting algorithms. (C1, C7, C8, C16, C18, F1, F3, F8)

D. Understand algorithm efficiency, Big-O notation, and why it should be considered in programming. (C1, C8, C16, C18, F1, F3, F8)
E. Analyze and select appropriate data structures to implement a solution to a problem. (C1, C7, C8, C16, C18, F1, F3, F8)
F. Design and implement data structures using classes and incorporating object-oriented concepts. (C1, C8, C16, C18, F1, F3, F8)
G. Demonstrate best practices of software development including testing, validation, and documentation. (C5, C6, C8, F2, F11)

III. INSTRUCTIONAL MATERIALS

A. The instructional materials identified for this course are viewable through www.ctcd.edu/books

B. One USB storage device or personal cloud-based storage.

IV. COURSE REQUIREMENTS

A. Attend both lecture and lab or in the case of online delivery, be actively engaged in Blackboard and maintain constant progress.

B. Be prepared to participate in discussion, team projects/assignments and take unannounced assessments relating to the lecture materials.

C. Complete all exams/assessments.

D. Submit all assignments on time.

V. ASSESSMENTS

A. Student content mastery will be evaluated in the following areas:
   • Assessments (midterm exam, quizzes, projects, etc.)
   • Final Assessment (final exam and/or semester project, participation)

B. Scheduled and unscheduled quizzes will be given at the discretion of the instructor.

C. Exams/assessments may be composed of both subjective and objective questions plus computer output.

D. A student must take all exams/assessments. No make-up exams/assessments will be given. Both online and on campus students who know in advance that they will be absent due to school sponsored trips, military duty or orders, or any other valid reason, must arrange to take an early exam/assessment. Unexpected absences due to illness or other extenuating circumstances will require the student to see the instructor about make-up work in lieu of the missed exam/assessment.
E. Students with unexcused absences will be given a zero for any missed work.

VI. SEMESTER GRADE COMPUTATIONS

<table>
<thead>
<tr>
<th>Course Requirements</th>
<th>Points</th>
<th>Points</th>
<th>Grade</th>
<th>Quality Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments</td>
<td>300</td>
<td>900</td>
<td>900-1000</td>
<td></td>
</tr>
<tr>
<td>Assessments</td>
<td>300</td>
<td>800</td>
<td>800 - 899</td>
<td></td>
</tr>
<tr>
<td>Final Assessment</td>
<td>400</td>
<td>700</td>
<td>700 - 799</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>1000</td>
<td>600</td>
<td>600 - 699</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>0 - 599</td>
<td></td>
</tr>
</tbody>
</table>

VII. NOTES AND ADDITIONAL INSTRUCTIONS FROM THE INSTRUCTOR

A. Information on the following Academic Policies, as described in the CTC Course Catalog will be followed:
   1. Withdrawals
   2. Grading
   3. Class Attendance and Course Progress
   4. Scholastic Honesty

B. Cell Phones and Pagers: Students will silence cell phones and mobile devices while in the classroom or lab.

C. Americans with Disabilities Act (ADA): Disability Support Services provide services to students who have appropriate documentation of a disability. Students requiring accommodations for class are responsible for contacting the Office of Disability Support Services (DSS) located on the central campus. This service is available to all students, regardless of location. Review the website at www.ctcd.edu/disability-support for further information. Reasonable accommodations will be given in accordance with the federal and state laws through the DSS office.

D. Instructor Discretion: The instructor reserves the right of final decision in course requirements and may make changes to the course outline and/or assignments as needed.

E. Civility: Individuals are expected to be aware of what a constructive educational experience is and be respectful of those participating in a learning environment. Failure to do so can result in disciplinary action up to and including expulsion.
VIII. COURSE OUTLINE

A. Lesson Zero: Introduce course requirements and objectives. Introduce students to the laboratory and use of the microcomputer.

1. Learning Outcomes: Upon successful completion of this unit the student will be able to:
   a. Demonstrate knowledge of course requirements as defined in the syllabus and reviewed by the instructor
   b. Enter, save, and execute a program using MS Visual C++
   c. Describe advanced programming techniques

2. Learning Activities:
   a. Instructor will cover the syllabus. (C5, F1)
   b. The instructor will review the lab procedures for the microcomputer lab. (C5, C6, C8, C9, C19, F1, F5, F11)
   c. The student will write a program using the materials presented (C1, C5, C6, C16, C17, C18, C19, F1, F2, F3, F5, F9, F11, F13, F17)

3. Lesson Outline: Follow the learning activities

B. Lesson One: Developing Efficient Algorithms

1. Learning Outcomes: Upon successful completion of this unit the student will be able to:
   a. Demonstrate knowledge on how to estimate algorithm efficiency using the Big O notation.
   b. Demonstrate knowledge on how to explain growth rates and why constants and non-dominating terms can be ignored in the estimation.
   d. Demonstrate knowledge on how to determine the complexity of various types of algorithms.
   e. Demonstrate knowledge on how to describe common growth functions (constant, logarithmic, log-linear, quadratic, cubic, exponential).
   f. Demonstrate knowledge on how to design various types of efficient algorithms using dynamic programming techniques.

2. Learning Activities:
   a. The instructor will demonstrate how to estimate algorithm efficiency using the Big O notation. (C5, C6, C8, F1, F5, F11, F13)
   b. The instructor will discuss and demonstrate how to explain growth rates and why constants and non-dominating terms can be ignored in the estimation. (C5, C6, C8, F1, F5, F11, F13)
   c. The instructor will demonstrate how to determine the complexity of various types of algorithms. (C5, C6, C8, F1, F5, F11, F13)
d. The instructor will demonstrate how to describe common growth functions (constant, logarithmic, log-linear, quadratic, cubic, exponential). (C5, C6, C8, F1, F5, F11, F13)
e. The instructor will demonstrate how to design various types of efficient algorithms using dynamic programming techniques. (C5, C6, C8, F1, F5, F11, F13)
f. The students will write a program using the materials presented. (C1, C3, C8, C17, F2, F3, F9)
g. Debugging of a computer program (C8, C16, C17, F9)

3. Lesson Outline: Follow the learning activities

C. Lesson Two: Sorting

1. Learning Outcomes: Upon successful completion of this unit the student will be able to:
   a. Demonstrate knowledge on how to analyze time complexity of various sorting algorithms.
   b. Demonstrate knowledge on how to design, implement, and analyze insertion, bubble, merge and quick sorts.
   c. Demonstrate knowledge on how to design and implement a binary heap.
   d. Demonstrate knowledge on how to design, implement, and analyze heap sort.
   e. Demonstrate knowledge on how to design, implement, and analyze bucket sort and radix sort.
   f. Demonstrate knowledge on how to design, implement, and analyze external sort for files that have a large amount of data.

2. Learning Activities:
   a. The instructor will demonstrate how to analyze time complexity of various sorting algorithms. (C5, C6, C8, F1, F5, F11, F13)
   b. The instructor will demonstrate how to design, implement, and analyze insertion, bubble, merge and quick sorts. (C5, C6, C8, F1, F5, F11, F13)
   c. The instructor will demonstrate how to design and implement a binary heap. (C5, C6, C8, F1, F5, F11, F13)
   d. The instructor will demonstrate how to design and implement a binary sort and radix sort. (C5, C6, C8, F1, F5, F11, F13)
   e. The instructor will demonstrate how to design, implement, and analyze external sort for files that have a large amount of data. (C5, C6, C8, F1, F5, F11, F13)
   f. The students will write a program using the materials presented. (C1, C3, C8, C17, F2, F3, F9)
   g. Debugging of a computer program (C8, C16, C17, F9)

3. Lesson Outline: Follow the learning activities.
D. Lesson Three: Linked Lists, Queues, and Priority Queues

1. Learning Outcomes: Upon successful completion of this unit the student will be able to:
   a. Demonstrate knowledge on how to create nodes to store elements in a linked list.
   b. Demonstrate knowledge on how to access the nodes in a linked list via pointers.
   c. Demonstrate knowledge on how to add, insert, and remove elements from a linked list.
   d. Demonstrate knowledge on how to add, insert, and remove elements from a linked list.
   e. Demonstrate knowledge on how to traverse elements in various types of containers.
   f. Demonstrate knowledge on variations of linked lists.

2. Learning Activities:
   a. The instructor will demonstrate how to create nodes to store elements in a linked list. (C5, C6, C8, F1, F5, F11, F13)
   b. The instructor will demonstrate how to access the nodes in a linked list via pointers. (C5, C6, C8, F1, F5, F11, F13)
   c. The instructor will demonstrate how to add, insert, and remove elements from a linked list. (C5, C6, C8, F1, F5, F11, F13)
   d. The instructor will demonstrate how to traverse elements in various types of containers. (C5, C6, C8, F1, F5, F11, F13)
   e. The instructor will demonstrate variations of linked lists. (C5, C6, C8, F1, F5, F11, F13)
   f. The students will design a program using the materials and algorithms presented. (C1, C3, C8, C17, F2, F3, F9)
   g. Debugging of a computer program (C8, C16, C17, F9)

3. Lesson Outline:
   a. Follow the learning activities.
   b. Review for and administer Midterm Assessment.

E. Lesson Four: Binary Search Trees

1. Learning Outcomes: Upon successful completion of this unit the student will be able to:
   a. Demonstrate knowledge on how to represent, access and search the elements in a binary search tree.
   b. Demonstrate knowledge on how to insert elements into, delete notes from and traverse a binary tree in in-order, post-order, and preorder.
2. **Learning Activities:**
   a. The instructor will demonstrate how to represent, access and search the elements in a binary search tree. (C5, C6, C8, F1, F5, F11, F13)
   b. The instructor will demonstrate how to insert elements into, delete notes from and traverse a binary tree in in-order, post-order, and preorder. (C5, C6, C8, F1, F5, F11, F13)
   c. The students will write a program using the materials presented. (C1, C3, C8, C17, F2, F3, F9)
   d. Debugging of a computer program (C8, C16, C17, F9)

3. **Lesson Outline:** Follow the learning activities.

F. **Lesson Five: Standard Template Library**

1. **Learning Outcomes:** Upon successful completion of this unit the student will be able to:
   a. Demonstrate knowledge on the relationships among containers, iterators, and algorithms.
   b. Demonstrate knowledge on how to distinguish sequence containers, associative containers, and container adapters.
   c. Demonstrate knowledge on how to use common features of containers.
   d. Demonstrate knowledge on how to store, retrieve, and process elements in sequence and associative containers.
   e. Demonstrate knowledge on how to implement the four types of STL algorithms.

2. **Learning Activities:**
   a. The instructor will demonstrate knowledge on the relationships among containers, iterators, and algorithms. (C5, C6, C8, F1, F5, F11, F13)
   b. The instructor will demonstrate how to distinguish sequence containers, associative containers, and container adapters. (C5, C6, C8, F1, F5, F11, F13)
   c. The instructor will demonstrate how to use common features of containers. (C5, C6, C8, F1, F5, F11, F13)
   d. The instructor will demonstrate how to store, retrieve, and process elements in sequence and associative containers. (C5, C6, C8, F1, F5, F11, F13)
   e. The instructor will demonstrate how to implement the four types of STL algorithms. (C5, C6, C8, F1, F5, F11, F13)
   f. The students will write a program using the materials presented. (C1, C3, C8, C17, F2, F3, F9)
   g. Debugging of a computer program (C8, C16, C17, F9)

3. **Lesson Outline:**
   a. Follow the learning activities.
   b. Review for and administer Final Assessment.