NEOSHO COUNTY COMMUNITY COLLEGE
MASTER COURSE SYLLABUS

COURSE IDENTIFICATION

Course Code/Number: CSIS 250

Course Title: Advanced Programming Methods

Division: ☒ Applied Science (AS) □ Liberal Arts (LA) □ Workforce Development (WD)
□ Health Care (HC) □ Lifetime Learning (LL) □ Nursing □ Developmental

Credit Hour(s): Three (3)

Effective Date: Fall 2013

Assessment Goal Per Outcome: 70%

COURSE DESCRIPTION

This course continues the development of discipline in program design, in style and expression, in debugging and testing, especially for larger programs. It will introduce the student to algorithmic analysis, basic aspects of string processing, recursion, internal search/sort methods, Object Oriented programming and simple data structures.

MINIMUM REQUIREMENTS/PREREQUISITES AND/OR COREQUISITES

CSIS-130 Introduction To Computer Information Systems and
CSIS-235 Pascal or CSIS-240 C++ Programming

TEXTS

The official list of textbooks and materials for this course is found on *my*Neosho.

http://www.neosho.edu/ProspectiveStudents/Registration/CourseSyllabi.aspx

GENERAL EDUCATION OUTCOMES

1. Practice Responsible Citizenship through:

- identifying rights and responsibilities of citizenship,
- identifying how human values and perceptions affect and are affected by social diversity,
- identifying and interpreting artistic expression.

2. Live a healthy lifestyle (physical, intellectual, social) through:
   - listing factors associated with a healthy lifestyle and lifetime fitness,
   - identifying the importance of lifetime learning,
   - demonstrating self-discipline, respect for others, and the ability to work collaboratively as a team.
3. Communicate effectively through:
   - developing effective written communication skills,
   - developing effective oral communication and listening skills.
4. Think analytically through:
   - utilizing quantitative information in problem solving,
   - utilizing the principles of systematic inquiry,
   - utilizing various information resources including technology for research and data collection.

COURSE OUTCOMES/COMPETENCIES (as Required)

Upon successful completion of the course the student should be able to:

1. Software Development

Methods and tools for software development

- Describe the general activities in the software life cycle.
- Describe the goals for "quality" software.
- Explain the following terms:
  software requirements,
  software specifications,
  algorithm,
  abstraction.
- Explain the use of "Big-O" to describe the amount of "work" done by an algorithm.
- Write a top-down design for a given problem, either in English or pseudocode.
- Explain why it is desirable to separate logical properties from implementation in computer programs.
- List features that make a program readable and modifiable.
- Show how deskchecking, code walk-throughs, and design and code inspections can improve software quality and reduce software effort.
- Explain what is meant by "abstract data type" and "data encapsulation
- Differentiate between static and dynamic allocation.

2. Stacks

Understand, define and use stacks in a computer program.

- Describe a stack and its operations at a logical level.
- Demonstrate the effect of stack operations on a particular implementation of a stack.
- Implement the stack abstract data type in a static array implementation.
- Explain the difference between static and dynamic allocation of the space in which the stack

elements are stored.

- Compare stack implementations in terms of source code length, use of storage space, and Big-O approximations of the stack operations.
- Determine when a stack is an appropriate data structure for a specific problem.
- Design and implement the solution to a problem for which a stack is an appropriate data structure.
- Show how stacks can be used to evaluate infix expressions.
- Evaluate an expression in prefix or postfix notation.

3. Queues

Understand, define and use Queues in a computer program.

- Describe the structure of a FIFO queue and its operations at a logical level.
- Demonstrate the effect of queue operations on a particular implementation of a FIFO queue.
- Implement the FIFO Queue ADT using a static array-based implementation.
- Implement the FIFO Queue ADT as a linked data structure.
- Compare queue implementation in terms of source code length, use of storage space, and Big-O approximations of the queue operations.
- Design and implement the solution to a problem for which a queue is an appropriate data structure.

4. Linked Lists

Understand, define and use Linked Lists in a computer program.

- Describe a list at a logical level.
- Describe how an ordered list may be implemented as a sequential or linked structure.
- Implement the following list operations for both sequential and linked implementations:
  Create and destroy a list
  Determine whether the list is empty or full
  Insert an element
  Retrieve an element
  Modify an element
  Delete an element
  Print the list elements in order
- Contrast the sequential and linked implementations of a list in terms of the Big-O approximations of their operations.
- Implement a circular linked list.
- Implement a linked list with a header node or a trailer node or both.
- Implement a doubly linked list.
- Discuss design issues for implementing lists with nonunique keys.
- Implement a linked list in an array of records.
- Explain the trade-offs between static and dynamic storage allocation.

5. Recursion

Understand, define and use recursion in a computer program.

- Do the following, given a recursive routine:
  determine whether the routine will halt,

determine the base case(s),
determine the general case(s),
determine what the routine does,
determine whether the routine is correct and, if it is not, correct it.

- Do the following, given a simple recursive problem:
  determine the base case(s),
  determine the general case(s),
  design and code the solution as a recursive procedure or function.
- Verify a recursive routine, according to the Three-Question Method.
- Decide whether a recursive solution is appropriate for a problem.
- Compare and contrast dynamic storage allocation and static storage allocation in relation to using recursion.
- Explain how recursion works internally by showing the contents of the run-time stack.
- Replace a recursive solution with iteration and/or the use of a stack.
- Explain why recursion may or may not be a good choice to implement the solution of a problem.

6. Object Oriented Programming

Fundamentals of Object Oriented Programming.

- Understanding and using Classes and Objects.
- Understanding and using Class Inheritance.

7. Sorting and Searching

Advanced techniques in sorting and searching.

- Design and implement the following sorting algorithms:
  straight selection sort
  bubble sort (two versions)
  merge sort
  quick sort
  heap sort
- Compare the efficiency of the sorting algorithms, in terms of Big-O and of space requirements.
- Discuss other efficiency considerations: sorting small numbers of elements, programmer time, sorting arrays of large data elements.
- Sort on several keys.
- Demonstrate the steps in the algorithms and to implement the following search algorithms:
  sequential search of unordered list
  sequential search of ordered list
  binary search
- Define the following terms:
  hashing
  rehashing
  collisions
  linear probing
  clustering
- Design and implement an appropriate hashing function for an application.
- Design and implement a collision-resolution algorithm for a hash table.

- Discuss the efficiency considerations for the searching and hashing algorithms, in terms of Big-O.

MINIMUM COURSE CONTENT

The following topics must be included in this course. Additional topics may also be included.

A. Language introduction/Review.
B. Verifying, Debugging, and Testing.
C. Data Design.
D. Stacks.
E. FIFO Queues.
F. Linked Lists.
G. Lists Plus.
H. Programming with Recursion.
I. Object Oriented Programming
K. Sorting Algorithms.
L. Searching.

STUDENT REQUIREMENTS AND METHOD OF EVALUATION

INSTRUCTIONAL METHODS

1. Explanation of important concepts by the instructor. Illustration of specific concepts may be made by the use of overhead projector, chalkboard, or audio-visual material.

2. Coding and testing of Pascal programming assignments on a microcomputer.

STUDENT REQUIREMENTS

The final grade in this course will be determined by lab exercises and programming assignments, majors exams, and the final exam using the following percentages.

Lab exercises and programming assignments.........	40%
Major exams......................................	30%
Final exam.......................................	30%

GRADING SCALE

90 - 100% =	A
80 - 89% =	B
70 - 79% =	C
60 - 69% =	D
0 - 59% =	F

ASSESSMENT OF STUDENT GAIN

The purpose of assessing student learning at Neosho County Community College is to ensure the educational purposes of the institution are met and appropriate changes are made in program development and classroom instruction to allow for student success. The instructor(s) of this course will determine the methods of assessment most appropriate and complete an assessment report at the end of the course.

Pre-assessment ideally begins at the beginning of the course when the instructor and the student determine through conference and observation the skill level of the student. Post-assessment to determine gain in competency will be measured at the end of each unit of study and at the end of the course.

Attendance Policy

1. NCCC values interactive learning which promotes student engagement in the learning process. To be actively engaged, the student must be present in the learning environment.

2. Unless students are participating in a school activity or are excused by the instructor, they are expected to attend class. If a student's absences exceed one-eighth of the total course duration, (which equates to one hundred (100) minutes per credit hour in a face-to-face class) the instructor has the right, but is not required, to withdraw a student from the course. Once the student has been dropped for excessive absences, the registrar's office will send a letter to the student, stating that he or she has been dropped. A student may petition the chief academic officer for reinstatement by submitting a letter stating valid reasons for the absences within one week of the registrar's notification. If the student is reinstated into the class, the instructor and the registrar will be notified. Please refer to the Student Handbook/Academic Policies for more information

3. Absences that occur due to students participating in official college activities are excused except in those cases where outside bodies, such as the State Board of Nursing, have requirements for minimum class minutes for each student. Students who are excused will be given reasonable opportunity to make up any missed work or receive substitute assignments from the instructor and should not be penalized for the absence. Proper procedure should be followed in notifying faculty in advance of the student's planned participation in the event. Ultimately it is the student's responsibility to notify the instructor in advance of the planned absence.

ACADEMIC INTEGRITY

NCCC expects every student to demonstrate ethical behavior with regard to academic pursuits. Academic integrity in coursework is a specific requirement. Definitions, examples, and possible consequences for violations of Academic Integrity, as well as the appeals process, can be found in the College Catalog, Student Handbook, and/or Code of Student Conduct and Discipline.

ELECTRONIC DEVICE POLICY

Student cell phones and other personal electronic devices not being used for class activities must not be accessed during class times unless the instructor chooses to waive this policy.

NOTE

Information and statements in this document are subject to change at the discretion of NCCC. Students will be notified of changes and where to find the most current approved documents.

ACCOMMODATIONS

If you are a student with a disability who may need accommodation(s), in compliance with Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act (ADA) of 1990, please notify the Dean of Student Services in the Student Services Office, Sanders Hall, 620-432-0304, on the Chanute Campus, or the Dean for the Ottawa and Online Campuses, 785-248-2798, on the Ottawa Campus as soon as possible. You will need to bring your documentation for review in order to determine reasonable accommodations, and then we can assist you in arranging any necessary accommodations.

NON-DISCRIMINATION POLICY

The following link provides information related to the non-discrimination policy of NCCC, including persons with disabilities. Students are urged to review this policy.

http://www.neosho.edu/Departments/NonDiscrimination.aspx

SEXUAL MISCONDUCT POLICY (TITLE IX)

At NCCC, it is the responsibility of an instructor to help create a safe learning environment in the classroom, including both physical and virtual classrooms. All instructors are considered mandatory reporters at NCCC, therefore any information regarding sexual misconduct that is shared by a student in one-on-one meetings with the instructor must be reported to appropriate personnel at the College. Instructors will keep the information private to the greatest extent possible, but it is not confidential. Generally, climate surveys, classroom writing assignments or discussions, human subjects research, or events such as Take Back the Night events do not provide notice that must be reported to the Coordinator by employees, unless the reporting party clearly indicates that they wish a report to be made.

The following link provides information related to the sexual misconduct policy of NCCC, including resources, reporting options, and student rights. Students are urged to review this policy.

http://www.neosho.edu/TitleIX.aspx

COURSE NOTES