COURSE OUTLINE

COS101 Course Number	Introduction to Computer Science Course Title
4 Credits	3 Lecture/2 Laboratory Hours: lecture/laboratory/other (specify)

Catalog description:

Introduces both majors and non-majors to the concepts and topics of computer science. Students will develop algorithmic thinking and abstraction using a 3-D animation programming language and MATLAB, a numerical programming tool for scientists and engineers. Students will also explore the various topics of computer science including computer architecture, algorithm analysis, operating systems and programming languages.

Prerequisites: MAT 037, 042 or equivalent

Required texts/other materials:

Thumb or Flash drive of any size

Last revised: Spring 2019

Course coordinator: Donald Reichman

Information resources: http://www.mccc.edu/~reichman

Other learning resources: COS101 tutoring is available in the Learning Center, room LB224 on the second floor of the library on the West Windsor Campus.

Course Competencies/Goals (CC/G):

The student will be able to:

1. Develop solutions to problems using a formal programming language.
2. Decompose problems logically and communicate complex ideas simply.
3. Analyze the fundamental structures and concepts of a formal programming language.
4. Solve problems central to the various fields of study within computer science.

Course-specific General Education Knowledge Goals and Core Skills

General Education Knowledge Goals (GEKG)
Goal 4. Technology. Students will use computer systems or other appropriate forms of technology to achieve educational and personal goals.

MCCC Core Skills (CS)
Goal B. Critical Thinking and Problem-solving. Students will use critical thinking and problem solving skills in analyzing information.
Goal D. Information Literacy. Students will recognize when information is needed and have the knowledge and skills to locate, evaluate, and effectively use information for college level work.
Goal E. Computer Literacy. Students will use computers to access, analyze or present information, solve problems, and communicate with others.

Units of study in detail:

UNIT 1: Programming Using a 3-D Animation Language
Learning Objectives (All use CC/G 1, CC/G 2, CC/G 3, GEKG 4, CS Goal B, CS Goal E)

TOPIC 1: Overview of the Alice 3-D Animation Programming Language

Upon completing this topic a successful student will be able to:

- Describe the basic structures of the Alice programming language.
- Define an Object and explain how it is used in programming.
- Describe interactivity in the context of a computer program.

TOPIC 2: An Overview of Program Design and 3-D Animation

Upon completing this topic a successful student will be able to

- Define a computer animation problem.
- Create an initial scene.
- Build a storyboard that outlines the solution to a desired animation problem.
- Translate the storyboard into an animation program.
- Edit the code to refine the animation.

TOPIC 3: An Introduction to Object Oriented Programming

Upon completing this topic a successful student will be able to

- Distinguish between objects and classes.
- Define and write a world-level method.
- Pass parameters to methods.

TOPIC 4: Class Level Methods and Inheritance

Upon completing this topic a successful student will be able to:

- Build a variety of class level methods.
- Define data encapsulation.
- Build their own classes that encapsulate data and methods.
- Demonstrate the use of class hierarchy and inheritance.

TOPIC 5: Object Visibility and Event Handlers

Upon completing this topic a successful student will be able to:

- Pass parameters to event handlers.

- Demonstrate the use of the “isShowing” and “Opacity” methods.
- Rotate an object around an invisible object.
- Use an invisible target.
- Create a billboard.

TOPIC 6: Functions

Upon completing this topic a successful student will be able to:

- Use built-in functions.
- Create their own functions.
- Pass parameters to functions.
- Distinguish between functions and methods.

TOPIC 7: Execution Control Using *if/else* and Boolean Functions

Upon completing this topic a successful student will be able to:

- Determine the appropriate time to use an *if/else* statement
- Describe the logical control that is exhibited by an *if/else* statement
- Create a Boolean function, a function that returns “true” or “false”
- Use a Boolean function to determine the appropriate method to call
- Create Boolean expressions with multiple conditions
- Create a function with nested *if*/*else* statements

TOPIC 8: Execution Control Using Loops

Upon completing this topic a successful student will be able to:

- Use a counter controlled loop to repeat a call to a method
- Use nested loops to create repeated action within some repeated action
- Implement an event controlled loop
- Implement a *while* loop that uses compound Boolean expressions

Unit 2: Programming Using a Numerical Programming Language

Learning Objectives (All use CC/G 1, CC/G 2, CC/G 3, GEKG 4, CS Goal B, CS Goal E)

New Topic 9: An Introduction to Numerical Programming Using MATLAB

Upon completing this topic a successful student will be able to:

- Describe the syntax, functionality and capabilities of MATLAB to solve engineering and scientific problems.
- Understand and use MATLAB primitive data types, and built-in MATLAB functions to trace MATLAB programs.
- Provide a computer-based programming solution for a simple engineering problem using a numerical processing language such as MATLAB.

Unit 3: Computer Science Concepts

Learning Objectives (All use CC/G 4)

TOPIC 10: Computer Memory and the CPU

Upon completing this topic a successful student will be able to:

- Describe generally how a computer accomplishes execution of a program (CS Goal D).
- Describe the Stored program concept (CS Goal D).
- Convert a binary number to a decimal number (CC/G 4, CS Goal B).
- Convert a decimal number to a binary number (CC/G 4,CS Goal B).
- Convert from octal to decimal and conversely (CC/G 4,CS Goal B).
- Convert from hexadecimal to decimal and conversely (CC/G 4,CS Goal B).
- Describe how the CPU works (CS Goal D).
- Describe the relationship between main memory and the CPU (CS Goal D).

TOPIC 11: Operating Systems

Upon completing this topic a successful student will be able to:

- Describe the various operating system modules (CS Goal D).
- Describe the terms multiprogramming and multiprocessing (CS Goal D).
- Describe how an operating system manages multiprogramming (CS Goal D).
- Describe how time sharing works in a client-server system (CS Goal D).

TOPIC 12: Algorithms and Algorithm Efficiency

Upon completing this topic a successful student will be able to:

- Define an algorithm (CS Goal D).
- Describe how computer science categorizes the running time of algorithms (CS Goal D).

- Explain the Order of a variety of algorithms (CS Goal B).
- Describe the characteristics of an algorithm ((CS Goal D).
- Distinguish between sequential and binary searches (CS Goal B).

TOPIC 13: Programming Languages

Upon Completing this topic a successful student will be able to:

- Distinguish between an assembler and a compiler (CS Goal D).
- Distinguish between an interpreter and a compiler (CS Goal D).
- Describe the early high-level programming languages (HLL) (CS Goal D).
- Describe the role of C in the development of HLL (CS Goal D).
- Describe the general principles of Object Oriented Languages (OOL) (CS Goal D).
- Explain why JAVA was created and distinguish it from C++ (CS Goal D).

Evaluation of student learning:
Specific methods for evaluating a students progress through the course is up to the discretion of the instructor. Below is a suggested format which balances the lab and lecture components.

Exams: Three one-hour exams = 30% of the grade and One comprehensive final exam = 20% of the grade. The exam questions will ask the student to use the structures of the Alice programming language to create and animate scenes. Students will also describe the outcome from segments of Alice code. The last one-hour exam will ask the student to solve problems and analyze topics from a variety of computer science topics.

Labs: Students will design, implement and test programs using the Alice Programming Language. Their projects will ask them to storyboard an animation, and then create the code to implement the animation. They will do 12 projects of increasing complexity throughout the semester plus one comprehensive final project. The average of the 13 lab projects = 50% of the grade

Academic Integrity Statement: Mercer County Community College is committed to Academic Integrity—the honest, fair and continuing pursuit of knowledge, free from fraud or deception. This implies that students are expected to be responsible for their own work and that faculty and academic support services staff members will take reasonable precautions to prevent the opportunity for academic dishonesty.

See http://www.mccc.edu/admissions_policies_integrity.shtml for a complete explanation of policies and procedures regarding academic integrity and academic integrity violations.