Course Description

COP1332 | Introduction to Visual Basic Programming | 4.00 credits

This course introduces computer science and non-major students to fundamental programming skills using the visual basic integrated development environment. Students will learn program design, the fundamentals of event driven object-oriented programming, arrays, validation of user input, and how to create menu driven programs and multiple form applications.

Course Competencies:

Competency 1: The student will demonstrate knowledge of visual programming by:

1. Creating a visual program to solve a problem
2. Interpreting a series of instructions used in a visual program
3. Identifying the basic structures of the program (sequence, decision, and repetition)

Competency 2: The student will demonstrate knowledge of program design by:

1. Describing the different program development tools
2. Creating the design for a program using the development tools. 3. Modifying the design for an existing program

Competency 3: The student will apply and synthesize knowledge of user interface design by:

1. Describing the difference between a console program and a Graphical User Interface (GUI)
2. Creating a console program
3. Modifying existing user interfaces in sample programming projects
4. Creating a GUI incorporating good design principles for a programming project
5. Using the following GUI components in assignments: buttons, labels, text boxes, dialogs, picture boxes, check boxes, radio buttons, group boxes and list or combo boxes

Competency 4: The student will demonstrate understanding and application of a modern Integrated Development Environment (IDE) by:

1. Describing the major components of an IDE and their functions
2. Examining, creating, and debugging programming projects using an IDE

Competency 5: The student will demonstrate knowledge of fundamental computer program components by:

1. Identifying the fundamental components of computer programs, including variables, data types, control structures, and functions, to establish a foundational understanding of programming concepts and structures
2. Analyzing the relationships and interactions among fundamental program components to comprehend their roles in program execution and data manipulation
3. Applying the knowledge of fundamental program components to create simple programs and solve basic computational problems, demonstrating practical understanding and proficiency

Competency 6: The student will demonstrate knowledge of data types by:

1. Explaining the appropriate use of each of the following fundamental data types: Integer, Single, Double, String, and Boolean
2. Explaining the properties of a variable, such as its name, value, scope, persistence, and size
3. Identifying and using variables appropriately within programming projects
4. Using explicit type conversions in programming projects
5. Explaining the form and uses of array variables
6. Creating and using array variables within programming projects

Competency 7: The student will demonstrate the ability to analyze fundamental computer programming constructs by:

1. Creating programs using control structures and functions
2. Incorporating each of the following programming constructs into programming projects: sequential processing;

counted, pre-test and post-test iteration (for next, do while, loop until) ; and simple and complex selection structures (if, if-else, nested-ifs, select case)

3. Creating programs using function, method, and procedure calls within programming projects
4. Creating programs that respond to user-generated events
5. Validating user input from text boxes
6. Controlling user action during the execution of a program using the Enable/Disable feature of an object

Competency 8: The student will demonstrate the ability to analyze fundamental computer programming operations by:

1. Using mathematical operators within programming projects
2. Using relational operators within programming projects
3. Using logical (Boolean) operators within programming projects
4. Using string manipulation functions and methods
5. Using intrinsic functions and/or methods for type conversion and mathematical operations

Competency 9: The student will demonstrate the ability to analyze advanced user interface design by:

1. Evaluating the effectiveness of advanced user interface design elements, such as navigation, visual hierarchy, and interactive components, to assess their impact on user experience and usability
2. Comparing and contrasting advanced user interface designs across different platforms and applications to identify emerging trends and best practices in the field
3. Critiquing advanced user interface designs based on principles of human-computer interaction and user-centered design, providing constructive feedback and recommendations for improvement

Competency 10: The student will demonstrate the ability to analyze program development and maintenance processes by:

1. Applying functional decomposition techniques to break a programming design problem into smaller pieces
2. Using diagrams and/or other design documents to illustrate the design of a programming solution
3. Using diagrams and/or pseudocode to explain the detailed design of a method, procedure, or function.
4. Comparing and contrasting source code and executable results
5. Documenting code following industry standard practices and procedures

Competency 11: The student will demonstrate the ability to synthesize knowledge of fundamental computer programming by:

1. Designing event-driven, object-oriented programs that use fundamental programming constructs
2. Implementing event-driven, object-oriented programs that use fundamental programming constructs
3. Testing event-driven, object-oriented programs that use fundamental programming constructs
4. Debugging event-driven, object-oriented programs that use fundamental programming constructs

Competency 12: The student will demonstrate industry best practices by:

1. Participating as a member of a collaborative team
2. Researching and reviewing professional sources of information and presenting findings orally, in writing, and/or with a slide presentation

Learning Outcomes:

- Solve problems using critical and creative thinking and scientific reasoning.
- Demonstrate knowledge of ethical thinking and its application to issues in society
- Use computer and emerging technologies effectively