COP 1170 – Introduction to Computer Programming using Visual Basic

Course Justification

This course is the first computer programming course in the Computer Information Systems Associate in Arts degree program; is required in the Computer Programming and Analysis, Database Technology, Internet Services Technology, and Networking Services Technology Associate in Science degree programs; is required in the Computer Programming, and Web Development Specialist College Credit Certificate programs; and is a pre-requisite for the Microsoft Solutions Developer College Credit Certificate Program.

This course is an introductory computer programming course that will familiarize the student with fundamental concepts in computer programming and will prepare the student for the study of advanced topics.

COP 1170 – Introduction to Computer Programming using Visual Basic

Course Description – The Visual Basic Integrated Development Environment is used to develop various computer application programs. The topics of program design, event-driven object-oriented programming, arrays, report generation, and file processing are included. This course may be taken by those not majoring in Computer Information Systems. Knowledge of high school algebra is recommended. Laboratory fee. (3 hr. lecture; 2 hr. lab, 4 credits)

Course Competencies

Competency 1: The student will demonstrate knowledge of computer system architecture by:
   a.   Identifying the function of each of the following items: CPU, RAM, ROM, hard disk, floppy disk, monitor, keyboard, mouse, and printer.
   b.   Describing the differences between programs and data.
   c.   Identifying some of the major functions of computer operating systems such as file and peripheral management and program loading and execution.

Competency 2: The student will apply and synthesize knowledge of User Interface design by:
   a.   Modifying existing user interfaces in sample programming projects.
   b.   Creating a Graphical User Interface incorporating good design principles for a programming project.
   c.   Using at least the following GUI components in assignments: buttons, labels, text boxes, dialogs, picture boxes, check boxes, radio buttons, group boxes and list or combo boxes.

COP 1170 – Introduction to Computer Programming using Visual Basic

Competency 3: The student will demonstrate understanding and application of a modern Integrated Development Environment (IDE) by:
   a.   Describing the major components of an IDE and their functions.
   b.   Using an IDE to examine, create and debug programming projects.

Competency 4: The student will demonstrate analytical knowledge of fundamental computer program components by:
   a.   Explaining the role of each of the following typical program components: controls, classes, objects, properties, methods, functions, procedures, forms, modules and projects.
   b.   Incorporating these components into both existing and new programming projects.

Competency 5: The student will demonstrate analytical knowledge of fundamental computer programming data structures by:
   a.   Explaining the appropriate use of each of the following fundamental data types: Integer, Single, Double, String, and Boolean.
   b.   Explaining the properties of a variable such as its name, value, scope, persistence, and size.
   c.   Appropriately declaring and using variables within programming projects.
   d.   Using explicit type conversions in programming projects.
   e.   Explaining the form and uses of array variables.
   f.   Creating and using array variables within programming projects.
   g.   Implementing a programmer-defined data structure (record) within a programming project.

COP 1170 – Introduction to Computer Programming using Visual Basic

Competency 6: The student will demonstrate analytical knowledge of fundamental computer programming constructs by:

a. Modifying and expanding short programs using control structures and functions.

b. 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).

c. Using function, method and procedure calls within programming projects.

d. Creating programs that respond to user-generated events.

Competency 7: The student will demonstrate analytical knowledge of fundamental computer programming operations by:

a. Using mathematical operators within programming projects

b. Using relational operators within programming projects

c. Using logical (Boolean) operators within programming projects.

d. Using string manipulation functions and methods such as Length, IndexOf, and Substring.

e. Using intrinsic functions and/or methods for type conversion and mathematical operations.

Competency 8: The student will demonstrate analytical knowledge of computer input/output by:

a. Modifying and/or creating programs which read, create, and modify sequential files.

b. Modifying and/or creating programs which produce formatted printed output.

COP 1170 – Introduction to Computer Programming using Visual Basic

Competency 9: The student will demonstrate analytical knowledge of the program development and maintenance processes by:

a. Applying the techniques of functional decomposition to break a programming design problem into smaller pieces.

b. Using diagrams and/or other design documents to illustrate the design of a programming solution.

c. Using diagrams and/or pseudocode to explain the detailed design of a method, procedure or function.

d. Comparing and contrasting source code and executable code.

e. Incorporating adequate and meaningful comments into the source code of programming projects.

Competency 10: The student will demonstrate the ability to synthesize knowledge of fundamental computer programming by designing, implementing, testing and debuging event-driven, object-oriented programs that use each of the following fundamental programming constructs: basic computation, simple I/O, and standard conditional and iterative constructs.