Douglas College

CURRICULUM GUIDELINES

A:	Division:	**Instructional**	Date:
B:	Department/ Program Area:	**Computing Science**	New Course ☒ Revision ☐

If Revision, Section(s) Revised:

Date Last Revised:

C:	**CMPT 101**	**D:**	**Introduction to Programming Using Visual Basic**	**E:**	**3**
	Subject & Course No.		Descriptive Title		Semester Credits

F: Calendar Description:

This course introduces students to a programming environment and language. Time is initially spent learning about hardware and software components of a computer system, files and directories/folders management in Windows and the software life cycle. Most of the course is spent on object based programming and fundamental building blocks for programming in Visual Basic 6. Topics cover forms, controls, properties management, events and event handlers, data types including arrays, control structures, functions and procedures.

G: Allocation of Contact Hours to Types of Instruction/Learning Settings

Primary Methods of Instructional Delivery and/or Learning Settings:
Lecture/Laboratory

Number of Contact Hours: (per week / semester for each descriptor)
 Lectures 2 hours per week
 Laboratory 2 hours per week

Number of Weeks per Semester:
14

H: Course Prerequisites:
Math 101 or Math 12 with at least a grade C or an acceptable score on math assessment test

I. Course Corequisites:
None

J. Course for which this Course is a Prerequisite:

CMPT 110

K. Maximum Class Size:
34
Maximum Lab Size: **34**

L: PLEASE INDICATE:

☐ Non-Credit

☐ College Credit Non-Transfer

☒ College Credit Transfer: Requested ☒   Granted ☐

SEE BC TRANSFER GUIDE FOR TRANSFER DETAILS (www.bccat.bc.ca)

M: Course Objectives/Learning Outcomes: At the end of this course the successful student should be able to
-create, locate, move, copy and delete a file or folder in Windows
-create, navigate, and delete folder tree structure in Windows
-understand the process of program design and development
-design a GUI by using objects (forms and controls) and managing properties of controls
-design a program to solve a well-defined problem
-implement a program from a design using structured programming
To implement a program, the student should be able to:
-understand what a form is, how to create it, and be familiar with the properties of a form
-understand what menus are and how to add them to a form
-understand what objects and controls are, and be able to add controls to a form
-understand the properties of a control and explore the events that can occur with a control
-differentiate among the various data types
-distinguish between variables and constants, and understand how to include them in programs
-select appropriate scope for a variable, differentiate between variables that are global to a project and those visible only to a form
-perform number and string manipulations including the use of built-in procedures
-learn how to accept input through input boxes
-format values for output using formatting functions and output boxes
-understand the purpose of procedures and be able to write reusable code in Sub procedures and Function procedures
understand the role of parameters and be able to differentiate between value and reference parameters
-evaluate conditions using the relational operators and combine conditions using logical operators
-understand and code selection logic using IF-THEN and SELECT CASE statements
-understand and code looping routines DO-WHILE, DO-UNTIL, and FOR-NEXT
-understand what arrays are and use one-dimensional arrays to hold data
-pass arrays between procedures

N: Course Content

The course content below applies to both the lecture and the lab as they have equal contact time and are in phase.

1 Introduction to Computers
   1.1 Hardware and Software Components of a Computer System
   1.2 Levels of Languages

2 Windows
   2.1 Files and Folders Management

3 Programming General
   3.1 Program Development Cycle
   3.2 Programming tools - flowcharts, pseudocode

4 Programming in Visual Basic
   4.1 Forms, Controls, Properties, Events
   4.2 Numbers, variables, and constants
   4.3 Strings
   4.4 Input and Output using Text Boxes
   4.5 Built-in Functions - Numeric Functions, String Functions

5. Procedures
   5.1 Sub Procedures
   5.2 Function Procedures
   5.3 Scope of Variables
   5.4 Value and Reference Parameters

6. Decisions
   6.1 Relational and Logical Operators
   6.2 IF-THEN Blocks
   6.3 SELECT CASE Blocks

7. Repetitions
   7.1 DO-WHILE and DO-UNTIL Loops
   7.2 FOR NEXT Loops

8. One-Dimensional Arrays
   8.1 Creating and Accessing Arrays
   8.2 Passing Arrays between Procedures

O: Methods of Instruction

There are three components to the course: lectures, labs, and self directed learning (i.e. programming assignments).

The lecture is used to introduce new material; usually via a sequence of theoretical concepts, then one or more example case studies. The textbook is to be used as an additional source of problems and examples.
The two hour weekly lab parallels the lecture by considering the application of the new material.

Assignments are marked according to correctness of the algorithms and coding style.

P: Textbooks and Materials to be Purchased by Students
   -Schneider, David I, An Introduction to Programming Using Visual Basic 6.0,
   Fourth Edition, Prentice Hall, ISBN 0-13-936428-5
   -Portfolio for Programming Assignments
   -Two 3 1/2" high density diskettes

Q: Evaluation:

Evaluation will be carried out in accordance with Douglas College policy. The instructor will present a written course outline with specific evaluation criteria at the beginning of the semester. Evaluation will be based on some of the following:

weekly labs	**15% - 30%**
assignments (3 to 4)	**15% - 25%**
tests (1 to 2)	**15% - 30%**
lecture attendance, class participation	**0% - 5%**
short quizzes	
final exam	**25% - 40%**

R: Prior Learning Assessment and Recognition: specify whether course is open for PLAR
Not at this time.

Course Designer(s)   Education Council/Curriculum Committee Representative

---

Dean/Director   Registrar