ISYS 219 and ISYS 317 - A course objectives and language features ('C' vs. Visual Basic (VB)) comparison

Important Notes:
1.   ISYS 219, ISYS 317 and ISYS 318 should be taught, as a sequence, as closely as possible, and by the SAME instructor.

2.   This sequence should be seen as a progression in offering students an incremental approach to learning key concepts such as:
   a.   Systems Development Life-Cycle (SDLC)
   b.   Program Development Life-Cycle (PDLS)
   c.   Problem Solving Approach
   d.   Programming Techniques [procedural, event-driven (EDP), object-oriented, visual programming (VP)]
   e.   Structured and Modular Programming Methodologies
   f.   Object-Orientation (OO) Programming Methodology
   g.   Client/Server Development (C/S)
   h.   Data Management Principles (DM)

3.   This progression should offer students a path in learning modern programming concepts. This path is synonymous to the OO and Web-based features and capabilities of traditional and modern programming languages. For example, a such simple relationship can be (approximately) described as follows:

FORTRAN, COBOL, BASIC, APL < 'C' < Pascal, Ada, Modula < C++ < Java < Smalltalk, Eiffel

4.   Cliff's 'VB Equation':

VB = (some old) BASIC + (some) 'C' + (some) 'C++' + (some) OO + EDP + VP + DM or C/S

5.   In the table below, a rating between 1 and 5 (1 lowest, 5 highest) is being provided. Low rating means either not available in the language or difficult to emulate; and vice-versa, high rating means either built-incapability or easy to emulate.

6.   The listed language capabilities and features, as covered in the ISYS 219 and ISYS 317 are **grouped** for correlation purposes. An **arithmetic average** is computed for each group, as well as a **final average** for all groups. In addition, a **weight** is added to each group, and a **final weighted average** is computed.

7.   [1] ADT = Abstract Data Type, or user defined data types
8.   [2] Switch, Select (Case type), complex looping
9.   [3] Arrays, Strings, Queues, Stacks, Linked-lists, Trees, Graphs
10.   [4] IDE = Interactive Development Environment.

Teaching Objective	'C'	VB
BASIC FEATURES (G1)		
Basic Program Structure	*4*	*5*
Simple Data Types	*4*	*5*
Basic Program Control	*5*	*5*
Modules, Procedures, Functions	*3*	*4*
Built-in Function	*2*	*5*
Library Support	*5*	*5*
Average **Weight = 30%**	**3.83** **1.15**	**4.83** **1.45**
ADVANCED FEATURES (G2)		
Complex Data Types	*4*	*4*
Extended Data Types (ADT[1])	*5*	*5*
Advanced Program Control[2]	*5*	*4*
Data Structures[3]	*4*	*4*
File Processing	*4*	*4*
Average **Weight = 10%**	**4.40** **0.44**	**4.20** **0.42**
OTHER FEATURES (G3)		
Structured-Modular Programming	*4*	*5*
Procedural Programming	*5*	*5*
Event-Driven Programming	*1*	*5*
Error-Handling	*2*	*4*
Visual Programming	*1*	*5*
Object-Oriented Programming	*2*	*4*

	UoR	BSIS
Multimedia Programming	*1*	*5*
Web Programming	*2*	*3*
Portability	*4*	*0*
Generic Programming, Extensibility	*2*	*2*
Scripting Capability	*3*	*3*
Data Management (Client-Server, Front-end Processing)	*2*	*4*
Average	**2.42**	**3.57**
Weight = 20%	**0.49**	**0.71**

PEDAGOGICAL FEATURES (G4)

	UoR	BSIS
Language Syntax and Constructs	*2*	*4*
Programming Style	*2*	*4*
Thinking Discipline	*3*	*3*
Proper Documentation	*2*	*4*
IDE Support[4]	*4*	*4*
Average	**2.60**	**3.80**
Weight = 20%	**0.52**	**0.76**

INDUSTRY (G5)

	UoR	BSIS
Develops Strong Professional Programming Skills	*5*	*3*
Meets Employer Expectations	*4*	*4*
Basis for Future Programming Skills Development	*4*	*4*
Average	**4.33**	**3.66**
Weight = 20%	**0.87**	**0.73**
Final Average	**3.44/5**	**4.14/5**
Weighted Average	**67.74%**	**84.13%**