TIK2O0 Final Culminating Activity – Developing your own Software

Due Date and Weighting

Your software is due on the ***last day of classes***. It is worth ***10 marks out of 30*** for your final evaluation (10% overall).

How to hand in your Software

Store all files related to your software in a ***single folder***. When you are satisfied that you have completed your work to the best of your ability, ***copy*** the folder to **I:\In\Nolfi\Tik2o0\Final Culminating Activity**\YourName, where *YourName* stands for your name.

Description

To consolidate and extend the programming concepts that we have learned in TIK2O0, each student will develop his/her own Visual Basic program. The nature of the software is left to the discretion of the student; however, ***all of the following elements must be included:***

- ***at least three*** command buttons
- ***a picture*** for each button (see "**Area Calculator.vbp**" in "**I:\Out\Nolfi\tik2o0\Option Button Examples**")
- ***at least one*** text box
- ***two or more*** option buttons
- ***two or more*** check boxes (see "Object Naming Conventions" in MSDN to discover the recommended prefix for naming check boxes)
- ***one or more*** picture boxes
- ***one or more*** image controls
- processing of ***numeric data*** using ***any or all*** of the numeric operators that we have studied
- processing of ***string data*** using the string concatenation operator (&)
- appropriate use of both ***local variables*** and ***global variables***
- ***at least one*** control from the "Microsoft Windows Common Controls 6.0 (SP6)" library (MSCOMCTL.OCX)

For ***extra credit***, you may also wish to include the following:

- ***the string manipulation functions*** "Left," "Right," "Mid" and "Len" (you will need to do some research for this)
- use the VB menu editor to create ***menus*** (you will need to do some research for this)

Evaluation Criteria

The software that you develop will be judged according to the following criteria:

1. Coding Practices (Style)

(a) The code should be logical, tidy and constructed according to the general guidelines learned throughout the course (i.e. proper indentation, comments for major blocks of code and abstruse code, meaningful identifier names, etc).

(b) The code should be as short as possible. Duplicate code should be eliminated by using general subs (see the "ResetGame" sub in the "Craps" software found in **I:\Out\Nolfi\tik2o0\Craps**).

2. Difficulty of Coding

(a) Programs that are difficult to code will be given more credit than those that are easy to code.

(b) The code should include a large number of the programming techniques learned in TIK2O0.

3. User Interface

The user interface should be attractive, well organized and user-friendly.

4. How Interesting and Useful is your Software?

Programs that are interesting and useful will receive more credit than programs that are not.

How to generate good Software development Ideas

- ☐ Choose a program that is related to one or more of your interests
- ☐ Take into consideration software that you have used
- ☐ Take into consideration examples from this course
- ☐ Discuss your ideas with classmates, friends, acquaintances, relatives, etc

Before you Begin…

- ☐ Please, please ask me for advice before you begin. I need to confirm that the software that you have chosen is appropriate to your skill level and our time constraints.
- ☐ ***Do not begin coding until you have a sound overall plan. Remember that writing a computer program is like "teaching" a computer how to solve a problem. If you do not know how to solve a particular problem, it will not be possible for you to "teach" a computer how to solve it.***

Evaluation Guide

Categories	*Criteria*	*Descriptors*					*Level*	*Average*
		Level 4	Level 3	Level 2	Level 1	Level 0		
Knowledge and Understanding (KU)	**Overall Understanding of Programming Concepts**	Extensive	Good	Moderate	Minimal	Insufficient		
	Appropriate use of Local and Global Variables	Extensive	Good	Moderate	Minimal	Insufficient		
Application (APP)	**Correctness** To what degree is the output correct?	Very High	High	Moderate	Minimal	Insufficient		
	Declaration of Variables To what degree are the variables declared with appropriate data types?	Very High	High	Moderate	Minimal	Insufficient		
	Inclusion of Required Elements To what degree has the student included the required elements?	Very High	High	Moderate	Minimal	Insufficient		
Thinking, Inquiry and Problem Solving (TIPS)	**Algorithm Design and Selection** To what degree has the student used approaches such as solving a specific example of the problem to gain insight into the problem that needs to be solved?	Very High	High	Moderate	Minimal	Insufficient		
	Ability to Write Code Independently To what degree has the student been able to write code without assistance?	Very High	High	Moderate	Minimal	Insufficient		
	Difficulty What is the degree of difficulty required to develop the software project chosen by the student?	Very High	High	Moderate	Minimal	Insufficient		
	Interest and Usefulness To what degree is the software interesting and useful?	Very High	High	Moderate	Minimal	Insufficient		
Communication (COM)	**Indentation of Code** **Insertion of Blank Lines in Strategic Places** (to make code easier to read)	Very Few or no Errors	A Few Minor Errors	Moderate Number of Errors	Large Number of Errors	Very Large Number of Errors		
	Comments • Effectiveness of explaining abstruse (difficult-to-understand) code • Effectiveness of introducing major blocks of code • Avoidance of comments for self-explanatory code • Quality and completeness of "header" comment at the beginning of the code (header comment should include the application name, programmer's name, date, platform, description and limitations)	Very High	High	Moderate	Minimal	Insufficient		
	Descriptiveness of Identifier Names Variables, Objects, Subs, etc **Inclusion of Property Names with Object Names** (**e.g.** 'txtName.Text' instead of 'txtName' alone) **Clarity of Code** How easy is it to understand, modify and debug the code? **Adherence to Naming Conventions** (**e.g.** use "txt" for text boxes, "lbl" for labels, etc.)	Masterful	Good	Adequate	Passable	Insufficient		
	User Interface To what degree is the user interface well designed, logical, attractive and user-friendly?	Very High	High	Moderate	Minimal	Insufficient		
Extra Credit	**Going beyond the Required Elements** To what degree has extra credit been earned by exceeding expectations?	Very High	High	Moderate	Minimal	Not applicable		