

VITAL INFORMATION

Subject(s):	Careers, Computer Fundamentals 1-2
Topic or Unit of Study:	Software Development
Grade/Level:	9-12
Objective:	At the conclusion of this lesson students will be able to: <ol style="list-style-type: none">1. Declare variables in the context of a function definition.2. Program a module that returns a value.3. Construct complex mathematical formulas using composition of simpler formulas.4. Create appropriate test data for a mathematical formula.

Summary: Students update their previously simple calculator to support three new operations: average, next, and harmonic mean. Each of these is designed to reuse functionality already available, but inaccessible in the program. Average uses addition; next, subtraction; and harmonic mean, both multiplication and addition. In order to make use of existing functionality, the program is restructured slightly so that the simple arithmetic operations are contained in functions which can be reused for the more complex calculations.

IMPLEMENTATION

Learning Context: Students have just completed the simple calculator and have previously watched the Shift Happens video. The calculator program gets shifted in this exercise. Based on fictitious feedback from users, additional functionality has been specified. We're putting some of the soft in software and experiencing how small existing modules can be combined in different ways to produce new results.

Procedure:

1. Gather the students at the front of the room for a meeting. They will need to bring pencils and paper, which should be the policy from now on.
2. Display for students the class web page where there is a screen shot of the new calculator with the three new buttons. There are no built-in functions for these calculations, so they must be programmed. The latter two will need to be explained. For example, "2 4 next" results in 6. Harmonic mean is described well in a Wikipedia article. As a group create the equations $\text{average} = (a+b)/2 = \text{Add}(a,b)/2$, $\text{next} = b+(b-a) = b-(-b+a) = b-(a-b) = b-\text{Subtract}(a,b)$, $\text{harmonic} = 2ab/(a+b) = 2*\text{Multiply}(a,b)/\text{Add}(a,b)$. Here a and b are shorthand for Val

(txtFirstNumber) and Val(txtSecondNumber).

3. Show the new function definition for Add from the web page and describe its parts. Have the class figure out the definition of subtract. Multiply will be left as an individual exercise. Next show how the click handler for the add button can be rewritten to make use of the function. Be certain to explain how the variables are passed by position. As a group figure out how the click handler for the average should therefore be written. The subtract and its matching next are left as an individual exercise.

4. Explain to students that these functions, especially the last, are complicated enough to warrant formal testing. When their work is checked they should have a list of five test cases for each new button. Example documentation of a test case is on the class web page.

5. When students feel their work is complete, check the program and the written test cases based on the rubric.

Differentiated Instruction:

There is little differentiation in this assignment. Students may work at their own pace as soon as group discussion is complete.

Sample Student Products:

The web page shows a screen shot of the application, but not functionality. The text to the updated Add function and its two clients is complete code. Other code is either temporarily written on the board or incorporated into each student's individual project.

Collaboration:

Students will work individually.

Time Allotment:

1 class period. 55 Min. per class.

Author's Comments & Reflections:

Reflections will follow in a diary entry.

Re. collaboration: After the initial group discussion, students should work alone on this assignment.

Re. time allotment: Time allotment may have to be adjusted based on how well students seem to understand during the group discussion. The lesson should not be started until all students are finished or nearly finished with the first, simple calculator.

MATERIALS AND RESOURCES

Instructional Materials:

Students should have access to the class web site. A PDF of the page for this lesson is attached.

Attachments

- | |
|---------------------------------------|
| 1. Shifted Calculator |
|---------------------------------------|

Resources:

- Technology resources:
Visual Basic

STANDARDS & ASSESSMENT

Standards:

AZ- Career and Technical Education Programs

- **Level :** Career Preparation (Grades 10 - 12)
- **Program :** Information Technology CIP No. 15.1200
 - **Option :** Software Development - Option C
 - **Competency :** *3.0 DEVELOP APPROPRIATE WORK HABITS FOR SUCCESSFUL EMPLOYMENT IN INFORMATION TECHNOLOGY
 - **Indicator :** 3.3 Complete tasks accurately
 - **Indicator :** 3.4 Complete tasks with minimal supervision
 - **Competency :** *9.0 UTILIZE TECHNOLOGY REQUIRED IN AN INFORMATION TECHNOLOGY WORKPLACE
 - **Indicator :** 9.5 Apply folder and directory management techniques
 - **Competency :** 16.0 PARTICIPATE IN INFORMATION TECHNOLOGY WORK-BASED LEARNING EXPERIENCES
 - **Indicator :** 16.1 Use technology appropriate for a job in information technology
 - **Indicator :** 16.5 Adapt to changes in the workplace
 - **Competency :** 27.C DEMONSTRATE PROGRAM ANALYSIS AND DESIGN
 - **Indicator :** 27.4c Determine input and output
 - **Indicator :** 27.6c Use stepwise refinement to improve design
 - **Indicator :** 27.7c Develop a testing plan
 - **Indicator :** 27.8c Write documentation
 - **Competency :** 28.C USE SOFTWARE TO CREATE PROGRAMS
 - **Indicator :** 28.1c Enter and modify code using a program editor
 - **Indicator :** 28.2c Compile and execute programs
 - **Indicator :** 28.3c Correct syntax errors
 - **Indicator :** 28.5c Use recognized conventions for naming identifiers and formatting code
 - **Indicator :** 28.6c Employ debugging strategies to eliminate errors
 - **Competency :** 29.C TEST AND DEBUG TO VERIFY PROGRAM OPERATION
 - **Indicator :** 29.1c Test individual program modules
 - **Indicator :** 29.2c Identify boundary cases and generate appropriate test data
 - **Competency :** 30.C WRITE CODE TO PERFORM ARITHMETIC CALCULATIONS
 - **Indicator :** 30.1c Identify and use arithmetic operators correctly applying the order of operations with respect to programming
 - **Indicator :** 30.2c Interpret and construct mathematical formulas
 - **Competency :** 31.C EMPLOY MODULARITY IN WRITING PROGRAMS
 - **Indicator :** 31.1c Call standard library functions
 - **Indicator :** 31.2c Utilize parameters to pass data into program modules
 - **Indicator :** 31.4c Write and use modules that return values
 - **Competency :** 34.C USE SIMPLE DATA TYPES AND STRINGS
 - **Indicator :** 34.1c Declare numeric, Boolean, character and string variables

Assessment/Rubrics:

Rubrics

- | |
|------------------------------|
| 1. Shifted Calculator |
|------------------------------|