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:

1. Read the internal computer clock.
2. Output diagnostic information from their programs.
3. Time an operation internal to a computer program.
4. Recognize a For … Next loop and explain what it does.
5. Explain why they should not store intermediate values in user interface elements.

Summary: Students look in detail into the practice of storing results of intermediate calculations in user interface elements. In doing so they learn how to "instrument" their programs to produce diagnostic output, read the internal computer clock, and figure out what a for next loop does. They measure the time that this storage method requires and compare it to a much quicker alternative. In the end they can explain why the alternative is superior.

IMPLEMENTATION

Learning Context: Students have just completed a number of calculators and paycheck programs. In assessing them, I noticed an inefficient programming practice that should be avoided. Students investigate it to observe the inefficiency for themselves. They can use the measurement technique they learn to ensure that programs they write in the future are as efficient they believe them to be.

Procedure: 1. The activity is written up in detail on a web page which is printed and attached to this lesson plan. Although students can read the instructions on their computers, they probably have questions and comments that interest the entire group. We therefore generally conduct such activities at the front of the room near the screen where everyone can see and hear. Such is the plan for this activity.

2. Work with the students through the background information, instructions, and their questions, and then set them to work timing the two methods of storing intermediate values.

3. The output of this activity is a small table that contains the timing information. The expected duration of storing values in and retrieving them back from the user interface 10,000 times is 2 seconds. Using variables requires approximately 0.01 seconds. The ratio is near 180:1. Outliers indicate potential lack of understanding and should be investigated. The results should be summarized for the class.

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 example output with a 180:1 ratio.

Collaboration: Students will work individually.

Time Allotment: 1 class period. 35 Min. per class.

Author's Comments & Reflections: Reflections will follow in a diary entry.

MATERIALS AND RESOURCES

Instructional Materials: The web page for this activity is attached.

Attachments

1. **Profiling**

Resources:

- Technology resources:
  Internet Explorer, 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 :** 27.C DEMONSTRATE PROGRAM ANALYSIS AND DESIGN
        - **Indicator :** 27.6c Use stepwise refinement to improve design
        - **Indicator :** 27.7c Develop a testing plan
      - **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
      - **Competency :** 29.C TEST AND DEBUG TO VERIFY PROGRAM OPERATION
        - **Indicator :** 29.1c Test individual program modules
      - **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
      - **Competency :** 33.C UTILIZE REPETITION STRUCTURES
        - **Indicator :** 33.1c Identify various types of repetition structures
      - **Competency :** 36.C IDENTIFY WAYS TO INPUT AND OUTPUT INFORMATION
        - **Indicator :** 36.2c Use input/output statements in a program

Assessment/Rubrics: The table that students submit should be checked for accuracy and reasonable values. If need be, students can demonstrate the resulting program. It should not be necessary for the teacher to read and execute each student's code.