If Then Else

Author: Keith Alcock 09/28/2008 10:50:00 AM MDT

VITAL INFORMATION

Subject(s): Careers, Computer Fundamentals 1-2

Topic or Unit of Study: Integrated Unit

Grade/Level: 9-12

Objective: At the conclusion of this lesson students will be able to:

1. Write both the single-line and multi-line If/Then/Else statement.
2. Program two statements on a single line.
3. Nest conditional statements.
4. Optimize code for runtime efficiency using an "early return."
5. Convert a series (think iteration) of If statements into a nested collection (think recursion).

Summary: Students improve the internal logic of their programs by rewriting If/Then/Else statements to be more efficient at runtime. In doing so they complete the objectives listed above. The code provided to students was purposefully written to include these inefficiencies and allow for the documented improvements.

IMPLEMENTATION

Learning Context: Students have just updated a Module for their Computer Integration activity, which involved editing code, so that all functionality is present. This next stage involves editing code for optimization purposes. The If/Then/Else statement isn't completely new to students; they saw it in the Overtime/Regular Paycheck activity. This activity provides a review in addition to covering the topic more thoroughly. . The next tasks involve creating the controls necessary to display student findings about computer integration into their various classes.

Procedure: 1. The activity is written up in fairly fine detail on a web page which is printed and attached to this lesson plan. It specifies where to find the If/Then/Else statements in question, what the syntaxes of the single-line and multi-line variants are, how to write two statements on a single line, what nesting and early return mean, and so on. It does not review the big picture, so the stage should be reset using some of the learning context from the introduction to the integrated unit.

2. Ensure that students can find the web page, ask for questions, and

have students start.

3. Since this is a long, sequential unit, more attention needs to be paid to holding it all together and keeping it synchronized. Visit students early and often to ensure that they are keeping up. It may be a good idea to have students demonstrate their progress beginning five minutes before the bell rings.

Differentiated Instruction: There is little differentiation in instruction and for If/Then/Else statements there is little that can differ in the deliverable.

Sample Student Products: The program should begin looking similar to the example on the web page for this activity and neither looks nor functionality change during the activity. If we were to profile runtime efficiency, however, it would improve..

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.

MATERIALS AND RESOURCES

Instructional Materials: The activity page from the class web site is printed and attached. The code that the lesson refers to has already been attached to earlier lesson plans.

Attachments

1. **If Then Else**

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 :** 27.C DEMONSTRATE PROGRAM ANALYSIS AND DESIGN
        - ■ **Indicator :** 27.6c Use stepwise refinement to improve design
      - **Competency :** 28.C USE SOFTWARE TO CREATE PROGRAMS
        - ■ **Indicator :** 28.1c Enter and modify code using a program editor
      - **Competency :** 29.C TEST AND DEBUG TO VERIFY PROGRAM OPERATION
        - ■ **Indicator :** 29.1c Test individual program modules
      - **Competency :** 28.C USE SOFTWARE TO CREATE PROGRAMS

    - **Indicator :** 28.2c Compile and execute programs
    - **Indicator :** 28.5c Use recognized conventions for naming identifiers and formatting code
    - **Indicator :** 28.6c Employ debugging strategies to eliminate errors
- **Competency :** 32.C WRITE CODE USING CONDITIONAL STRUCTURES
    - **Indicator :** 32.1c Compare values using relational operators (=, >, <, >=, <=, not equal)
    - **Indicator :** 32.2c Evaluate Boolean expressions
    - **Indicator :** 32.4c Construct decision statements such as if/else, if, switch case
    - **Indicator :** 32.5c Use nested decision structures in programs
    - **Indicator :** 32.6c Implement multiple-choice decision statements such as if/else, if, switch case

Assessment/Rubrics:

By the end of the class period, students should have rewritten single-line If/Then/Else statements and written multi-line statements as directed, and then rewritten one multi-line statement independently. Each activity contributes a few criteria to the larger rubric. This activity add points for single-line and multi-line statements as well as the rewrite as described by the If Then Else rubric.

Rubrics

1. **If Then Else**