CheckBox

Author: Keith Alcock 09/24/2008 11:38:00 PM 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. Explain what a CheckBox is and what it's used for.
2. Add a CheckBox to and remove one from a TabPage.
3. Name a CheckBox and assign text to it.
4. React to click events on a CheckBox.
5. Query the checked property of a CheckBox.

Summary: Students add and configure another CheckBox on the Schedule tab of the application for the Computer Integration project. The original Full-day CheckBox proved to be suboptimal, so it is being replaced by a Half-day CheckBox. During replacement, students learn from seeing how the original CheckBox worked. They will use both the Design and Code editors for this task.

IMPLEMENTATION

Learning Context: Students have just added a RadioButton to the program for their Computer Integration activity in a similar way. The task following the current task is to complete the code that displays student class schedules based on the states of all the RadioButtons and the CheckBox.

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 CheckBoxes, how to convert from negative logic to positive (without calling it that), and how to rewire the click handler for the new CheckBox. 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 the CheckBox 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 end looking similar to the example on the web page for the subsequent activity.

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. Code that needs to be edited is already attached to the RadioButton activity. Students do also work with the Designer, but that work is summarized in screen shots.

Attachments

1. CheckBox

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 :** 28.7c Access program and language documentation

- **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 :** 32.C WRITE CODE USING CONDITIONAL STRUCTURES
  - **Indicator :** 32.2c Evaluate Boolean expressions
  - **Indicator :** 32.4c Construct decision statements such as if/else, if, switch case
- **Competency :** 36.C IDENTIFY WAYS TO INPUT AND OUTPUT INFORMATION
  - **Indicator :** 36.1c Provide user with means to input data on a console and/or GUI

Assessment/Rubrics:

By the end of the class period, students should have added, named, and titled the CheckBox for Half-day, converted code to use it, and rewired click handlers to connect it up. Each activity contributes a few criteria to the larger rubric. This activity adds points for controls, logic, and convention as described by the CheckBox rubric.

Rubrics

1. **CheckBox**