

## VITAL INFORMATION

---

<b>Subject(s):</b>	Careers, Computer Fundamentals 1-2
<b>Topic or Unit of Study:</b>	Software Development
<b>Grade/Level:</b>	9-12
<b>Objective:</b>	At the conclusion of this lesson students will be able to:  <ol style="list-style-type: none"><li>1. Instantiate objects from existing classes, a teacher object from its teacher class.</li><li>2. Distinguish between an object and a class.</li><li>3. Describe has-a and is-a relationships.</li><li>4. Read and understand existing but unfamiliar code well enough to make minor modifications to it.</li></ol>

<b>Summary:</b>	Students "inherit" a project from elsewhere. It is written with classes, and they need to attach two in order to make it work. They modify their project file to include them and then add the appropriate few lines of code to put them to work. To demonstrate understanding, they personalize the messages that are produced and also add an additional activity. Throughout they learn about object-oriented programming and in the end they have a tool for themselves.
-----------------	--

## IMPLEMENTATION

---

<b>Learning Context:</b>	Students have been struggling with naming files correctly. Rather than having the teacher run a program on his account to determine correct placement of deliverables (as in the grade report lesson), students can run their own program, provided they fill in some missing pieces. To encourage ownership, they customize it, and to put it to immediate use, they add this very activity to the collection. This program also provides a way for students to compare characters that may not be distinguishable, a task they found difficult previously.
--------------------------	--

<b>Procedure:</b>	<ol style="list-style-type: none"><li>1. Introduce the topic referring to the learning context above and information from the web page. Touch lightly on object-oriented topics.</li><li>2. Have students copy the files from the P: drive to their H: drive and follow instructions on adding a few lines of code to make everything work in a personalized way.</li></ol>
-------------------	---

3. Have students include the answer to the one question from the web page when they turn in the announcement of completion into the black box.

<b>Differentiated Instruction:</b>	There is little differentiation in instruction, but an expectation that the products are differentiated.
<b>Sample Student Products:</b>	They should look significantly like the example on the web page.
<b>Collaboration:</b>	Students will work individually.
<b>Time Allotment:</b>	1 class period. 45 Min. per class.
<b>Author's Comments &amp; Reflections:</b>	Reflections will follow in a diary entry.

## **MATERIALS AND RESOURCES**

---

**Instructional Materials:** web page for the activity (attached), the Visual Basic files to modify

**Attachments**

1. <a href="#">File Name Tester</a>
-------------------------------------

**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 :** 29.C TEST AND DEBUG TO VERIFY PROGRAM OPERATION
      - **Indicator :** 29.1c Test individual program modules
    - **Competency :** 36.C IDENTIFY WAYS TO INPUT AND OUTPUT INFORMATION
      - **Indicator :** 36.4c Output text with formatting
    - **Competency :** 38.C EMPLOY OBJECT-ORIENTED PROGRAMMING TECHNIQUES
      - **Indicator :** 38.1c Make a distinction between an object and a class
      - **Indicator :** 38.2c Describe relationships between classes such as is-a, has-a
      - **Indicator :** 38.3c Instantiate objects from existing classes
      - **Indicator :** 38.10c Write a client program that includes user-defined objects

**Assessment/Rubrics:**

**Rubrics**

1. <a href="#">File Name Tester</a>
-------------------------------------