

**Computer Application Lesson 1 to Prepare for
UIL Computer Application Contest**

Lesson Plan Title: Simplifying the IF/IIf Function

Goal of Lesson: To have students understand the ins and out of the IF/IIf function.

Grade Level/Course: BIM (9-12)

TEKS Addressed:

(4C,D) Compile information from primary and secondary sources in systematic ways using available technology (ELAR 110.42,43,44)

(1H) Compile written ideas and representations, interpret empirical data into reports, summaries, or other formats, and draw conclusions. (ELAR 110.46)

(5C) Summarize information from text through the use of outlines, study guides, or learning logs (ELAR 110.48)

(4G,7A) Follow oral and written directions. (Bed 120.22)

Skill in use of database features, formulas and functions

(4A, 3A, 6A) Perform calculations involving money, time , space, materials, and data. (MktEd 124.12,24)

(4G,7A) Follow oral and written instructions. (Bed 120.22,42)

(2E) Use computer technologies to record, locate, analyze, present, and exchange information. (TechEd 124.47)

§130.272. Principles of Information Technology (One-Half to One Credit).

(8) The student applies spreadsheet technology. The student is expected to:

- (A) identify the terminology associated with spreadsheet software and its functions;
- (B) format and organize numerical content to perform mathematical processes such as addition, subtraction, multiplication, and division; percentages and decimals; and order of operations principle;
- (C) employ both student-created formulas and preprogrammed functions to produce documents such as budget, payroll, statistical tables, and personal checkbook register;
- (D) create and analyze spreadsheets incorporating advanced features such as lookup tables, nested IF statements, subtotals, cell protection conditional formatting, charts, and graphs

§130.114. Business Information Management I (One to Two Credits)

(11) The student applies spreadsheet technology. The student is expected to:

- (A) perform mathematical processes, including:
 - (v) prediction of patterns of data; and
- (B) formulate and produce solutions to a variety of business problems, including:
 - (i) budget, personal, and business;

§130.115. Business Information Management II (One to Two Credits).

(11) The student designs solutions to mathematical business problems using spreadsheet technology. The student is expected to:

- (A) recognize and apply lookup tables, built-in functions, macros, and advanced charts and graphs;

Computer Application Lesson 1- continued

Overview of Lesson: Lesson explains systematic approach to using the IF function in Excel starting with steps to draw a schematic of the logical operation and building to actually entering the function.

Materials Needed: Office 2007 or 2010 with Excel, computer, printer, paper, and pencil.

Procedures and Activities: This is a hands-on operation that guides students from the point of analyzing the logic in an IF statement to writing the statement.

Independent Practice: After completion of this lesson plan, suggest that students replicate this activity in Access using the IIf function and expression builder. Create a list of additional suggested statements that students may analyze and resolve into their own IF function.

Assessment: The Student Independent Exercise can be graded to determine if they have successfully understood the process of using IF functions and embedded IF functions. If students are able to added a field as suggested in Exercise D of the Independent Exercise, they are excelling in this arena

Students can then be asked to design IF statement particulars to be resolved by other students.

If students successfully solve additional IF statements they have achieved the goal. If they are able to concoct their own statements to be resolved into an IF statement, they are excelling.

Simplifying the IF/IIf Function

- I. This exercise is to explain the ins and outs of the IF/IIf function. This function takes a bit more effort to use as it involves logical decisions and actions based on those decisions.

A. Create an Excel Spreadsheet to match the one below.

	A	B	C	D	E
1	Name	Age	adolescent or teenager	adolescent, teenager, or adult	preadolescent, adolescent, teenager, or adult
2	Jim	5			
3	Joan	6			
4	Kavya	8			
5	Tran	12			
6	Joel	13			
7	Chin	16			
8	Padma	20			
9	Frank	28			

- B. Give the cells in the **Age** column a name so that they can be referenced by name rather than the column/row indication for each cell.

1. Select all the cells in Column B.
2. Select the Formula Ribbon.
3. Locate the Defined Names block and click the Define Name icon.
4. This will give you a window that names your range of cells, Age and shows you the scope of the range.

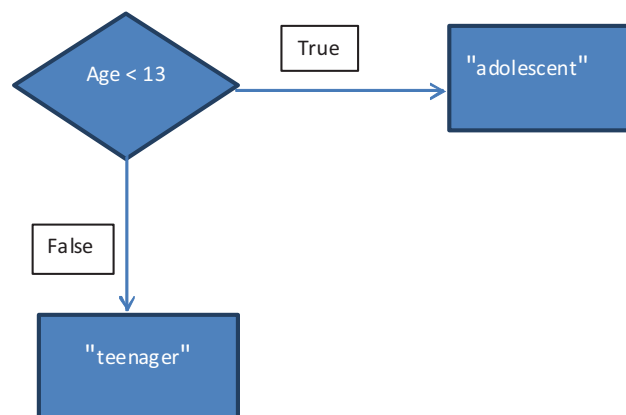
- C. Now, when you are in a cell and want to use a cell in Column B in a formula, you simply call the cell **Age**. As you fill down, the formula references the cell on the correct row for each replicated formula.

- II. We're going to use an exercise to fill in the cells in Column C with "adolescent" if the **Age** of the person on that row is less than 13, or it will insert "teenager" if the **Age** of the person is 13 or greater. Now, it would be easy enough to just glance down the column and insert the appropriate text manually, but this would be impossible if you had 10,000 records, so we need to learn to use formulas to do this globally for "real-life" files with large number of records.

- A. We'll use the IF or IIf function to do this operation, but before we start typing, let's reduce the operation to a visual representation in the form of a basic flow chart representation.

- B. Note that we use a diamond shape for a decision block, so we'll enter our question into that block. Rectangles are used for the operation blocks, so this representation shows one decision with the value-if-true operation to the right and the value-if-false operation below the decision block. This is the basic structure to be used for every decision/value-if-true/value-if-false operation.

- C. Our decision is to test to see if **Age** is less than 13, so we'll simply scribble that into the diamond. The value-if-true is "adolescent" so that drops into the box to the right, and the value-if-false is "teenager", so that goes in the lower box.



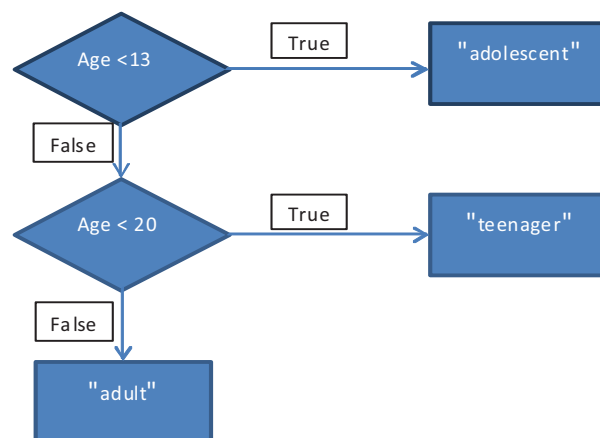
- D. When we have this drawn this pictorially, we can start typing our IF function directly into a Cell C2 of the spreadsheet. Starting at the top of the diagram, type = because that is the first symbol in a cell for all Excel functions, then add IF() (or if you're doing this in Access, use IIf without =). This is the start and end of the function. Now place your cursor inside the parentheses, and type your decision data as shown in the decision block =IF(Age<13). Now add a comma and the value-if-true block which should have the following appearance, =IF(Age<13,"adolescent"). Now add a comma and the value-if-false block and hit enter. Your function should be the following IF statement. (Note that you have the right number of parentheses because we put the open and closing parentheses in before we fleshed out the parameters.)

=IF(Age<13,"adolescent","teenager").

- E. Select Cell C2 with your newly entered function and fill down thru Cell C9 by putting your cursor in the lower left corner of the cell until it forms a cross (+) and double clicking. Now your spreadsheet should have the following appearance.

	A	B	C	D	E
1	Name	Age	adolescent or teenager	adolescent, teenager, or adult	preadolescent, adolescent, teenager, or adult
2	Jim	5	adolescent		
3	Joan	6	adolescent		
4	Kavya	8	adolescent		
5	Tran	12	adolescent		
6	Joel	13	teenager		
7	Chin	16	teenager		
8	Padma	20	teenager		
9	Frank	28	teenager		

- III. Note the spreadsheet will need some correcting to make it more accurate. For example, Padma and Frank who are 20 and 28 respectively are not teenagers. We will create another IF function for Cell D2 that adds an "embedded" IF to our original IF function, replacing the value-if-false parameter. First, however, let's modify our diagram. To do this, we leave the original decision block and value-if-true block in place. We will replace the value-if-false block with a decision block that tests to see if **Age** is less than 20 and adds a value-if-true block to the right with "teenager" because those who are 13 or greater and less than 20 are teenagers. The value-if-false block then becomes "adult" to indicate that anyone 20 or over is not a teenage, but an adult.

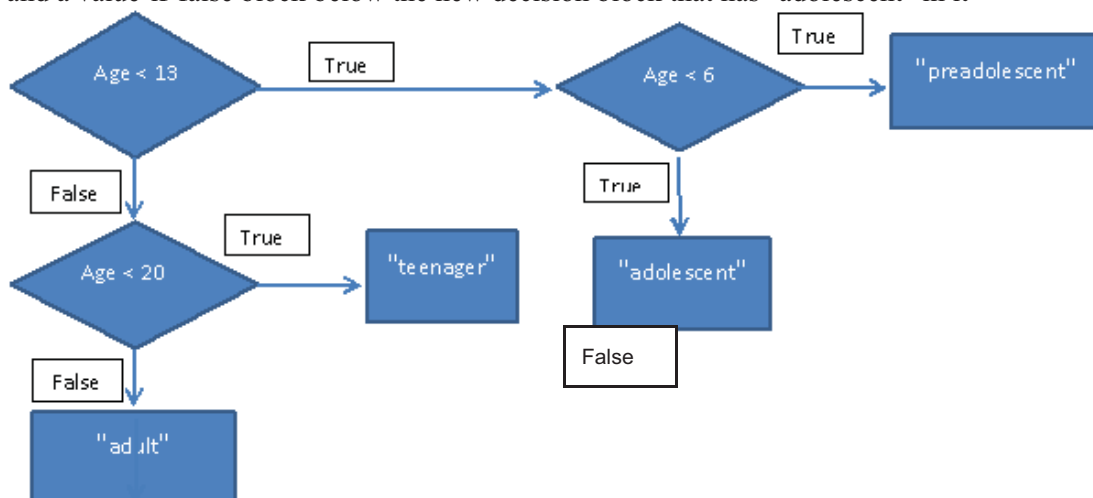


- A. Now, we need to enter an IF statement in Cell D2 that reflects the diagram. To do this we copy the formula from Cell C2 (Ctrl C) as it is displayed in the formula bar, click the check mark to the left of the formula bar, then paste it into the formula bar of Cell D2 (Ctrl V), and then we'll replace the entire value-if false parameter to match our diagram. Select the "teenager" parameter and delete it. We're going to replace it with another IF function.
- B. With your cursor in the place of the deleted parameter, start typing by entering IF() so we now have the following formula: =IF(Age<13,"adolescent",IF()). Note that we now have two closing parentheses which will be the correct number when we complete our function. Now put your cursor between the parentheses, and enter the decision block data (Age<20) followed by a comma, then the value-if-true data ("teenager" followed by a comma and the value-if-false data ("adult"). Your corrected function will have the following appearance, and this formula can be filled down from Cell D2 into Cells D3 thru D9. (The bold part is the newly added segment):
 =IF(Age<13,"adolescent",**IF(Age<20,"teenager","adult")**)
- C. Now your spreadsheet should have the following appearance.

	A	B	C	D	E
1	Name	Age	adolescent or teenager	adolescent, teenager, or adult	preadolescent, adolescent, teenager, or adult
2	Jim	5	adolescent	adolescent	
3	Joan	6	adolescent	adolescent	
4	Kavya	8	adolescent	adolescent	
5	Tran	12	adolescent	adolescent	
6	Joel	13	teenager	teenager	
7	Chin	16	teenager	teenager	
8	Padma	20	teenager	adult	
9	Frank	28	teenager	adult	

IV. Note the spreadsheet still needs an additional correction to make it accurate. For example, Jim is a pre-adolescent, so we need to add another embedded IF to replace the value-if-true block with another decision/value-if-true/value-if-false set of blocks to determine whether the Age is less than 6 and if so, label these as "pre-adolescent" and otherwise use "adolescent" in the value-if-false block.

- A. Now, basically, we will have an IF diagram whose value-if-true and value-if-false "legs" are also IF functions that are called embedded IF statements. Each IF is shown with decision, a value-if-true, and a value-if-false block.
- B. We're replacing the value-if-true block with a decision block that tests to see if the **Age** is less than 6. We then add a value-if-true block to the right of the decision block that has "preadolescent" in it and a value-if-false block below the new decision block that has "adolescent" in it



- C. Now we type the function with the two embedded IF statements into Cell E2 in your spreadsheet using the same method. Select Cell D2 and copy the function (Ctrl C) from the formula bar. Click the check mark to the left of the formula bar. Select Cell E2 and paste the function into it (Ctrl V). Delete the value-if-true parameter, and in its place type IF(). With your cursor between the parentheses, type the data from the new decision block data (Age<6) followed by a comma, then the data from the new value-if-true block ("preadolescent") followed by a comma and value-if-false data ("adolescent"). Now your spreadsheet should have the following appearance.

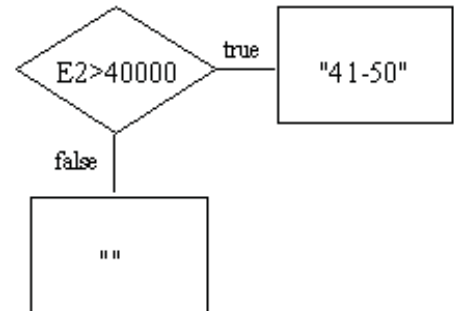
	A	B	C	D	E
1	Name	Age	adolescent or teenager	adolescent, teenager, or adult	preadolescent, adolescent, teenager, or adult
2	Jim	5	adolescent	adolescent	preadolescent
3	Joan	6	adolescent	adolescent	adolescent
4	Kavya	8	adolescent	adolescent	adolescent
5	Tran	12	adolescent	adolescent	adolescent
6	Joel	13	teenager	teenager	teenager
7	Chin	16	teenager	teenager	teenager
8	Padma	20	teenager	adult	adult
9	Frank	28	teenager	adult	adult

- D. With this approach to IF statements, the logic of the statement will be resolved in the diagram, and typing the statement in will be infinitely simplified, and the correct parentheses will be in place.

V. Independent Practice

- A. Page 7 Student Worksheet should be copied for each student in the class.
- B. Students should complete Exercises A thru D.
- C. Answer Keys for the exercises are the following:
- D. Exercise A

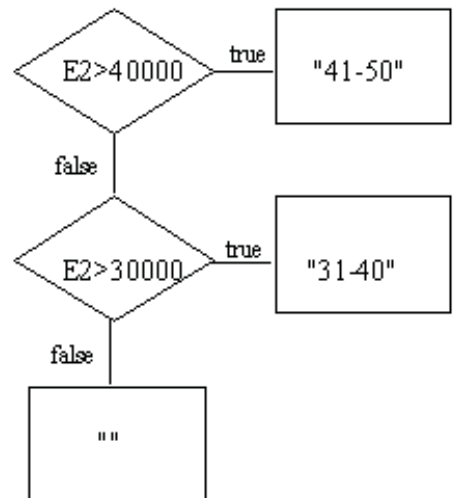
1. Diagram



2. =IF(E2>40000,"41-50", "")

E. Exercise B.....

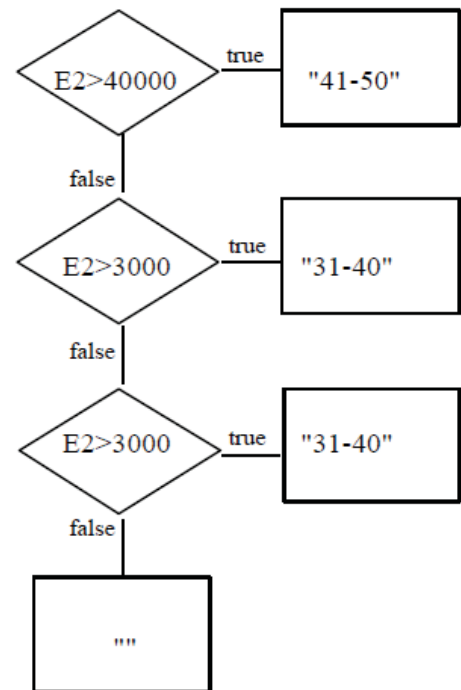
1. Diagram



2. =IF(E2>40000,"41-50",IF(E2>30000,"31-40",""))

F. Exercise C.....

1. Diagram



2. =IF(E2>40000,"41-50",IF(E2>30000,"31-40",IF(E2>20000,"21-30","under 30")))

3. Exercise D

- Student's description of what he/she is trying to accomplish.
- Student's diagram
- Student's IF statement with embedded IF in both value-if-true and value-if-false.

G. The worksheet should have the following appearance plus student's title and results in Column J.

	A	B	C	D	E	F	G	H	I
1	First Name	Last Name	Years	Starting Salary	Current Salary	Avg. % of Increase	Salary over 40,000	Between 30,000 & 40,000	All Salary Brackets
2	Ima	Cook	1	28,600	29,800	4%			21-30
3	Chris	Cross	7	24,100	51,300	16%	41-50	41-50	41-50
4	Molly	Fye	3	26,200	31,200	6%		31-40	31-40
5	Maura	Less	9	22,700	40,000	8%		31-40	31-40
6	Mel	Loewe	2	27,300	28,900	3%			21-30
7	Lou	Pole	11	20,800	42,800	10%	41-50	41-50	41-50
8	Mike	Raffone	14	19,650	50,600	11%	41-50	41-50	41-50
9	Art	Tillery	6	25,000	35,800	7%		31-40	31-40

H. Grading - total of 100 points (deduct for typos if you choose)

- Add two points for each correctly shaped block with the appropriate entry for Exercises A thru C. 30 pts
- Add 5 points for correct values in Column G 5 pts
- Add 5 points for correct values in Column H 10 pts
- Add 5 points for correct values in Column I 15 pts
- Add 10 points for each correct formula for Exercises A thru C 15 pts
- Add 10 points for correct diagram for Exercise D 10 pts
- Add 10 points for correct data and formula for Exercise D 15 pts

VI. **Student Independent Exercise-** Students are to complete Exercises A thru D after entering the following worksheet in Excel.

	A	B	C	D	E	F
1	First Name	Last Name	Years	Starting Salary	Current Salary	Avg. % of Increase
2	Ima	Cook	1	28,600	29,800	4%
3	Chris	Cross	7	24,100	51,300	16%
4	Molly	Fye	3	26,200	31,200	6%
5	Maura	Less	9	22,700	44,600	11%
6	Mel	Loewe	2	27,300	28,900	3%
7	Lou	Pole	11	20,800	42,800	10%
8	Mike	Raffone	14	19,650	50,600	11%
9	Art	Tillery	6	25,000	35,800	7%

A. Exercise A

1. In Cell G1, add the following label: **Salary over 40,000.**
2. On a blank page of paper write the title, Exercise A.
3. Create a diagram of the following on a piece of paper using the appropriate decision blocks, value-if-true blocks, and value-if-false blocks: If the value in Cell E2 is greater than 40000, the value-if-true is the label "41-50" and the value-if-false is a null character ("").
4. Using the diagram, write an IF function under your diagram starting with equal sign (=).
5. Enter your function in Cell G2 and fill down thru Cell G9.

B. Exercise B

1. In Cell H1, add the following label: **Between 30,000 and 40,000.**
2. On a blank page of paper write the title, Exercise B.
3. Create a diagram of the following on a piece of paper using the appropriate decision blocks, value-if-true blocks, and value-if-false blocks: If the value in Cell E2 is greater than 40,000, value-if-true is "41-50". Otherwise if the value in Cell E2 is greater than 30,000, the value-if-true is "31-40" and the value-if-false is a null character.
4. Using the diagram, write an IF function under your diagram starting with equal sign (=).
5. Enter your function in Cell H2 and fill down thru Cell H9.

C. Exercise C

1. In Cell I1, add the following label: **All Salary Brackets.**
2. On a blank page of paper write the title, Exercise C.
3. Create a diagram of the following on a piece of paper using the appropriate decision blocks, value-if-true blocks, and value-if-false blocks: If the value in Cell E2 is greater than 40,000, value-if-true is "41-50". Otherwise, if the value in Cell E2 is greater than 30,000, the value-if-true is "31-40" and the value-if-false tests to see if Cell E2 is >20000 returning "21-30" if the condition is met and a null character if the condition is not met.
4. Using the diagram, write an IF function under your diagram starting with equal sign (=).
5. Enter your function in Cell I2 and fill down thru Cell I9.

D. Exercise D

1. In Cell J1 have the students to create their own column heading.
2. On a blank page of paper write the title, Exercise D and describe what their exercise is trying to accomplish.
3. Have the students create a diagram with a decision block describing an IF condition with the value-if-true and value-if-false blocks also describing If condition. This would create an IF statement with embedded if for value-if-true and value-if-false.
4. Have students write their IF and embedded IF function under the diagram.
5. Have them enter their function in Cell J2 and fill down thru J9.

E. Print the worksheet.

**Computer Application Lesson 2 to Prepare for
UIL Computer Application Contest**

Lesson Plan Title: Understanding How Dates and Time Work

Goal of Lesson: To have students understand how computers deal with dates and time, so that they will understand how Excel and Access use dates and times in calculations.

Grade Level/Course: BIM (9-12)

TEKS Addressed:

(4C,D) Compile information from primary and secondary sources in systematic ways using available technology (ELAR 110.42,43,44)

(1H) Compile written ideas and representations, interpret empirical data into reports, summaries, or other formats, and draw conclusions. (ELAR 110.46)

(5C) Summarize information from text through the use of outlines, study guides, or learning logs (ELAR 110.48)

(4G,7A) Follow oral and written directions. (Bed 120.22)

Skill in use of database features, formulas and functions

(4A, 3A, 6A) Perform calculations involving money, time, space, materials, and data. (MktEd 124.12,24)

(4G,7A) Follow oral and written instructions. (Bed 120.22,42)

(2E) Use computer technologies to record, locate, analyze, present, and exchange information. (TechEd 124.47)

§130.272. Principles of Information Technology (One-Half to One Credit).

(8) The student applies spreadsheet technology. The student is expected to:

- (A) identify the terminology associated with spreadsheet software and its functions;
- (B) format and organize numerical content to perform mathematical processes such as addition, subtraction, multiplication, and division; percentages and decimals; and order of operations principle;
- (C) employ both student-created formulas and preprogrammed functions to produce documents such as budget, payroll, statistical tables, and personal checkbook register;

§130.114. Business Information Management I (One to Two Credits).

(11) The student applies spreadsheet technology. The student is expected to:

- (A) perform mathematical processes, including:
 - (i) addition, subtraction, multiplication, and division;
 - (ii) percentages and decimals;
 - (iii) order of operations principle;
 - (v) prediction of patterns of data; and
- (B) formulate and produce solutions to a variety of business problems, including:
 - (i) budget, personal, and business

§130.115. Business Information Management II (One to Two Credits).

(11) The student designs solutions to mathematical business problems using spreadsheet technology.

The student is expected to:

- (A) recognize and apply lookup tables, built-in functions, macros, and advanced charts and graphs;

Computer Application Lesson 2-continued

Overview of Lesson: This lesson will present the use of dates and time in Excel and in Access so that students will understand how they are kept in memory so they can be used in calculations. It will further instruct how to calculate using dates.

Materials Needed: Computer with Excel and paper with pencil.

Procedures and Activities: The instructional part of this lesson plan will be a step-by-step tutorial on dates and times in the computer using Excel. It will be followed by an exercise that uses date and time in an Excel calculation and in an Access calculation.

Independent Practice: Students will be asked to complete a worksheet that uses dates and time in an Excel worksheet or in an Access query.

Assessment: The Student Worksheet can be graded to determine if each student has successfully understood how dates are handled in computers, how to format dates, and how to calculate using dates. If students are able to complete the exercises satisfactorily, they understand the use and manipulation of dates. If their grade is high, they excel.

Date & Time in Computers

- I. This lesson attempts to explain the underlying method that computers use in order to deal with dates and times.
 - A. Everything in a computer is kept digitally, each letter has a digital representation, and a number is simply kept as a numeric field.
 - B. Time & date are kept in a single field in the computer as a number.
 1. The value to the left of the decimal represents the date, and it is kept as the number of days since 1/1/1900.
 2. The value to the right of the decimal represents time, and it is notated as a percentage of a 24-hour day. We'll come back to this later.
 - C. Enter the following on an Excel worksheet.

	A	B	C	D	E
1	Name	Birthdate	Number	Short Date Format	Time Only Format
2	Jack N. Debocks	1/1/1900			
3	Luke B. Foruleap	2/7/1942			
4	Ivan Chu	9/30/2000			
5	Maura DeSame	1/1/2011			

- D. In Cell C2, enter =B2, fill down thru Cell C5.
- E. Select Cells C2 thru C5 and fill across thru Cells E2:E5.
- F. Format each column as follows.
 1. Format Column B as Custom format: m/d/yyyy h:mm AM/PM
 2. Format Column C as a number with 2 decimals.
 3. Format Column D as a short date, m/d/yyyy.
 4. Format Column E as Time, m:hh AM/PM.
 5. Now your worksheet should have the following appearance:

	A	B	C	D	E
1	Name	Birthdate	Number	Short Date Format	Time Only Format
2	Jack N. Debocks	1/1/1900 12:00 AM	1.00	1/1/1900	12:00 AM
3	Luke B. Foruleap	2/7/1942 12:00 AM	15379.00	2/7/1942	12:00 AM
4	Ivan Chu	9/30/2000 12:00 AM	36799.00	9/30/2000	12:00 AM
5	Maura DeSame	1/1/2011 12:00 AM	40544.00	1/1/2011	12:00 AM

6. What we have learned is the same values that we put in Cells B2 thru B5 and replicated across thru Cells E2 thru E5 can be represented in different ways by just selecting a specific format in Excel to be date & time, a number, only the date, and only the time. They all still have the identical data in the field, the difference is just in the selection for formatting the field.
7. Now select Cells B2 thru E5 and format as a number with 2 decimals, and you will see that they are identical values all the way across the row. Then click Ctrl Z to undo this reformatting.

- G. **Date:** The date 1/1/1900 for computers is the "first date," and as we can see that date is held in the computer as the number 1, and all other dates are held as the number of days since that "first date."
1. It turns out that 2/7/1942 is 15,379 days past 1/1/1900.
 2. Then 9/30/2000 is 36,799 days after 1/1/1900.
 3. And last, 1/1/2011 40,544 days past 1/1/1900.
 4. What the computer actually uses in memory to do calculation with dates is the value that is shown in Column B. Note that the decimals are all zero because we didn't specify a time of day for each birth date.
 5. We can see the same date in Column C formatted as a short date just ignores whatever is to the right of the decimal.
- H. **Time:** Time is held in the computer in the same field as date. It is shown as the percentage of a 24-hour day. We specified 12:00 AM which is the zero hour for the day, so all of our formulas that show times show 12:00 AM, and Column B shows all values with zero in the decimals positions.
1. Change your date in Cells B3 thru B5 to show 6:00 AM, 12:00 PM, and 6:00 PM.
 2. This should change all the cells in Columns C thru E for these Rows 3 thru 5.
 3. Now when we look at the Number format for Cell C3, we see 15379.25.
 - a. 6:00 AM is equal to 6 divided by 24 or $\frac{1}{4}$ of a day; hence, the decimals value for 6:00 AM becomes .25.
 - b. Likewise 12:00 PM is fully 12 hours of a 24-hour period or $\frac{1}{2}$ of the day; hence, the decimal value becomes .50.
 - c. With 6:00 PM, we are 18 hours thru the day, so $18/24$ is $\frac{3}{4}$ of a day; hence the decimal value becomes .75.
 4. Now your worksheet should have the following appearance:

	A	B	C	D	E
1	Name	Birthdate	Number	Date Only Format	Time Only Format
2	Jack N. Debocks	1/1/1900 12:00 AM	1.00	1/1/1900	12:00 AM
3	Luke B. Foruleap	2/7/1942 6:00 AM	15379.25	2/7/1942	6:00 AM
4	Ivan Chu	9/30/2000 12:00 PM	36799.50	9/30/2000	12:00 PM
5	Maura DeSame	1/1/2011 6:00 PM	40544.75	1/1/2011	6:00 PM

- I. So far all we have learned is how the computer "sees" what we see as date and time, but knowing that the computer is looking simply at a number with decimals, we can see now that the computer can easily calculate using that number.
- II. Do the following calculations using dates.
- A. Add a label in Cell F1 that says **Date + 1**.
 - B. In Cell F2, enter the formula to add 1 to the date that is in Cell C2, and fill down thru Cell F5. The formula should be =C2+1. This should be formatted as a number with 2 decimals.
 - C. Add a label in Cell G1 that says **Date + 1 Short Date**.
 - D. Insert the formula =F2 in Cell G2 and replicate thru Cell G5, and format as Short Date.
 - E. What we learn is to add 1 to a date, adds one day. It doesn't change the month, the year, or the time, but simply adds one day to the date.

- F. Calculate how old each person will be as of 12/31/2011 on our worksheet.
1. Add a label in Cell H1 that says **Age**. The column should be formatted as a number.
 2. In Cell H2, subtract Cell D2 from 12/31/2011.
 - a. To tell Excel that 12/31/2011 is a date, we have to enclose it in quotation marks in a formula. If we entered 12/31/2011, the computer would do the math, take 12, divide it by 31, then divide the result by 2011, & then subtract the value from Cell D2!!!
 - b. Thus our formula is the following: ="12/31/2011"-D2.
 - c. Fill this formula down thru Cell H5.
 3. That really doesn't tell us how old a person is, does it? That tells us how many days there are between 12/31/2011 and the person's birth date.
 4. To turn this into years, we need to divide by 365.25. Which is the number of days in an average year. Every 4th year there is an extra day, so in 4 years, we have 1461 days using this calculation: $(365 * 3 + 366)$. When we divide this by 4, we find there are 365.25 days in a calendar year.
 - a. In the same formula in Cell H2, enclose the calculation in parentheses so that the subtraction will occur before any other calculation, and divide this by 365.25.
 - b. The formula should now say the following: ="(12/31/2011"-D2)/365.25
 - c. Fill down thru Cell H5.
 5. Note that the result gives you numbers with decimals. This shows the person's age in years and percent of next year.
 - a. This number includes decimals and if we viewed it with no decimals, it would round one of the values up. For an age, you cannot use part of a year. A student cannot get a driver's license until he/she is 16 years old, not 15 and 11 months and 30 days.
 - b. To remedy this, we have to add an INT function in front of our formula in Cell H2 and fill down thru Cell H5. This INT function should have the formula we've created as the parameter: =INT(("12/31/2011"-D2)/365.25).
 - c. Now all the Age values are whole numbers with 0-.00 in the decimal positions.

III. Do the following calculations using time.

- A. In Cell I2, use a formula to eliminate the date from a Date/Time field. There are several ways to do this. Here are 2 ways.
1. In Cell I1, add the label **Time without a Date**. Format as a number.
 2. In Cell I2, use the following formula and fill down: =C2-INT(C2)
 - a. This effectively subtracts the INT portion of C2 from the number with decimals.
 - b. Effectively, we found that Cell C3 showed 15380.25, so if we take that value (15380.25) and subtract the integer portion of this cell by truncating the decimals using the INT function, (15380.00), we get .25 which is the time represented in the number.
 3. In Cell J1, add the label **Time Without Date** using the MOD function. Format as a number.
 - a. Enter the formula =MOD(C2,1) in Cell J2 and fill down.
 - b. The MOD function returns the decimals remaining after dividing the first parameter by the second parameter.
 - c. Since dividing by 1 doesn't change the value, this simply truncates the integer value in the field.
 4. With Columns I and J both shown as numbers with 2 decimals, they should be equal.
 5. Now format both columns as time: h:mm AM/PM, and they should match Column E.

- B. Add 8 hours to each time in Column J.
1. In Cell K1, add the label **Time + 8 Hours**.
 2. In Cell K2, add 8 hours to Cell J2.
 - a. Since this is a Date/Time value, adding 8 would add 8 days. The time remains the same.
 - b. We need to describe 8 hours out of 24 hours. Do this by dividing 8 by 24 and adding that to Cell J2 and filling down.
 - c. The formula should be the following: $=J2+(8/24)$
 3. Now we show that 8 hours after 12:00 AM is 8:00 AM.
 4. Format Column K as numbers with 2 decimals.
 - a. Note that the last entry is 1.08. Adding 8 hours to 6:00 PM takes you past midnight and into the next day, so we now have the time as .08, but the date is now 1 or 1/1/1900.
 - b. Click Ctrl Z to undo the formatting.
 5. Now format Column K as a date in the Custom format m/dd/yyyy h:mm AM/PM.
 - a. Now we show the first three as having the date 1/01/1900 and the last one as 1/01/1900.
 - b. Click Ctrl Z to undo the formatting change.
- C. Your spreadsheet should now have the following appearance for the columns past E.

	A	F	G	H	I	J	K
	Name	Date + 1	Date + 1 - Short Date	Age	Time without a Date	Time without a Date using MOD	Time + 8 Hours
1							
2	Jack N. Debocks	2.00	1/2/1900	111.00	12:00 AM	12:00 AM	8:00 AM
3	Luke B. Foruleap	15380.25	2/8/1942	69.00	6:00 AM	6:00 AM	2:00 PM
4	Ivan Chu	36800.50	10/1/2000	11.00	12:00 PM	12:00 PM	8:00 PM
5	Maura DeSame	40545.75	1/2/2011	0.00	6:00 PM	6:00 PM	2:00 AM

IV. Independent Exercises.

- A. All students should have a copy of page 6, **Student Worksheet**, and a computer with Excel and blank paper for printing.
- B. Grade the printout of the Excel worksheet with the following criteria. Ignore font size and margins. If there is a typo in Columns A thru C or Cell F1, you may have to view their Excel file to see if their answers are correct or compute them manually.
- C. Calculate a grade based on the following for a total of 100 points. (Optionally, take off for typos.):
 1. Format for **Age** column matches key. 5 pts
 2. Format for **Years of Employ** column matches key. 5 pts
 3. Format for **1st Qtr Shift Start Time** column matches key. 5 pts
 4. Format for **2nd Qtr Shift Start Time** column matches key. 5 pts
 5. Format for **3rd Qtr Shift Start Time** column matches key 5 pts
 6. Format for **4th Qtr Shift Start Time** matches key 5 pts
 7. Format for **Expected Retirement** column matches key 5 pts
 8. Format for **Age at Retirement** column matches key 5 pts
 9. All values in **Age** are correct 10 pts
 10. All values in **Years of Employ** are correct 10 pts
 11. All values in **1st Qtr Shift Start Time** are correct. 10 pts
 12. All values in **2nd Qtr Shift Start Time** are correct 5 pts
 13. All values in **3rd Qtr Shift Start Time** are correct. 5 pts
 14. All values in **4th Qtr Shift Start Time** are correct 5 pts
 15. **Expected Retirement** in Cells J2 thru J6 are correct. 10 pts
 16. **Age at Retirement** in Cells K2 thru K6 match key. 5 pts
- D. The following is the answer key for Exercises A thru C.

	A	B	C	D	E	F	G	H	I	J	K
1	Name	Birthdate	Start Date	Age	Years of Employ	1st Qtr. Shift Start Time	2nd Qtr. Shift Start Time	3rd Qtr. Shift Start Time	4th Qtr. Shift Start Time	Expected Retirement	Age at Retirement
2	Ira Peate	9/28/1972	9/28/1994	38.00	15.00	1/1/2010 8:00 AM	2/1/2010 1:00 AM	3/3/2010 6:00 PM	4/3/2010 11:00 AM	12/31/2034	62.00
3	I. D. Clair	7/4/1978	7/3/2000	32.00	10.00	1/1/2010 1:00 PM	2/1/2010 6:00 AM	3/3/2010 11:00 PM	4/3/2010 4:00 PM	12/30/2040	62.00
4	Rob Banks	8/14/1981	8/14/2009	29.00	7.00	1/1/2010 6:00 PM	2/1/2010 11:00 AM	3/4/2010 4:00 AM	4/3/2010 9:00 PM	12/31/2043	62.00
5	Sue Case	2/15/1986	2/15/2008	24.00	2.00	1/1/2010 11:00 PM	2/1/2010 4:00 PM	3/4/2010 9:00 AM	4/4/2010 2:00 AM	12/30/2048	62.00
6	Tad Pohl	4/23/1988	4/23/2010	22.00	0.00	1/2/2010 4:00 AM	2/1/2010 9:00 PM	3/4/2010 2:00 PM	4/4/2010 7:00 AM	12/31/2050	62.00

V. **Student Worksheet** - Students are to complete Exercises A thru C after entering the following worksheet in Excel, then print their worksheet.

	A	B	C	D	E	F	G	H	I	J	K
1	Name	Birthdate	Start Date	Age	Years of Employ	1st Qtr. Shift Start Time	2nd Qtr. Shift Start Time	3rd Qtr. Shift Start Time	4th Qtr. Shift Start Time	Expected Retirement	Age at Retirement
2	Ira Peate	9/28/1972	9/28/1994			1/1/2010 8:00 AM					
3	I. D. Clair	7/4/1978	7/3/2000								
4	Rob Banks	8/14/1981	8/14/2003								
5	Sue Case	2/15/1986	2/15/2008								
6	Tad Pohl	4/23/1988	4/23/2010								

- A. Exercise A. These calculations should calculate the number of full years as whole numbers.
 1. In Cell D2, calculate the Age of each person by subtracting their **Birth date** from 12/31/2010, dividing the result by 365.25 and using the INT function with the entire calculate as the parameter. Format the column as a number with 2 decimals.
 2. Calculate the **Years of Employ** for each person by subtracting their **Start Date** from 12/31/2010, dividing the result by 365.25 and using the INT function with the entire calculate as the parameter. Format the column as a number with 2 decimals.
- B. Exercise B. Do the following calculation and format Cells F2 thru I2 as Custom dates in the format shown above.
 1. In Cell F3, add 5 hours to the Date/Time shown in Cell F2 and fill down thru Cell F6.
 2. In Cell G2, add 30.5 days plus 5 hours to the Date/Time value shown in Cell F2 and fill down thru Cell G6.
 3. Select Cells G2 thru G6 and replicate into Cell H2 thru H6 and I2 thru I6.
- C. Exercise C. Calculate Retirement columns using the following specifications.
 1. In Cell J2, calculate the difference between 62 and Age in Cell D2.
 - a. Multiply the result times 365.25.
 - b. Add the result to 12/31/2010 and format as a short date.
 - c. Fill down thru Cell J6.
 2. In Cell K2, calculate the difference between J2 and B2.
 - a. Divide the result by 365.25.
 - b. Truncate all decimal values.
 - c. Format as a number with 2 decimal places.
- D. Print your worksheet in landscape orientation on one page, modifying type font and margins as necessary.

**Computer Application Lesson 3 to Prepare for
UIL Computer Application Contest**

Lesson Plan Title: Using Update Queries and the Expression Builder

Goal of Lesson: To have students understand the ins and outs of using an update query to update a field in a table with calculations or functions using the expression builder.

Grade Level/Course: BIM (9-12)

TEKS Addressed:

(4C,D) Compile information from primary and secondary sources in systematic ways using available technology (ELAR 110.42,43,44)

(1H) Compile written ideas and representations, interpret empirical data into reports, summaries, or other formats, and draw conclusions. (ELAR 110.46)

(5C) Summarize information from text through the use of outlines, study guides, or learning logs (ELAR 110.48)

(4G,7A) Follow oral and written directions. (Bed 120.22)

Skill in use of database features, formulas and functions

(4A, 3A, 6A) Perform calculations involving money, time, space, materials, and data. (MktEd 124.12,24)

(4G,7A) Follow oral and written instructions. (Bed 120.22,42)

(2E) Use computer technologies to record, locate, analyze, present, and exchange information. (TechEd 124.47)

§130.272. Principles of Information Technology (One-Half to One Credit).

(9) The student applies database technology. The student is expected to:

(A) identify the terminology associated with database software and its functions;

(B) create, populate, edit, maintain, and save database files;

(D) perform data management procedures such as locating, sorting, searching, querying, organizing, and outputting data;

(F) produce organized reports with calculated figures.

§130.114. Business Information Management I (One to Two Credits)

(6) The student identifies database software to create databases that facilitate business decision making. The student is expected to:

(A) explain the principles of data analysis;

(B) explain the nature of tools that can be used to access information in the database system;

(C) choose appropriate software;

(D) define fields and type of data;

(E) enter database structure;

(G) analyze company's data requirements;

(H) design a database to meet business requirements; and

(7) The student applies data entry techniques to enter information in databases and is expected to:

(A) access information in the database system;

(B) build data in a data warehouse;

(C) create a meaningful data set;

(D) enter data into databases, tables, and forms;

(E) edit data in databases, tables, and forms;

(8) Student use commands to retrieve data and create reports from databases and are expected to:

(B) formulate queries;

(C) create and print reports; and

(D) manipulate data in the database management system.

Computer Application Lesson 3-continued

Overview of Lesson: Lesson explains a step-by-step approach to creating an update query in the design view and then using the expression builder to design appropriate calculations that will update the values in given fields.

Materials Needed: Office 2007 or 2010 with Access, an appropriate computer and printer, and blank paper.

Procedures and Activities: This is a hands-on operation that guides students through building a database, creating a simple table, entering data, building an update query to operate on fields that were not initialized to particular values, and working through the expression builder in a step-by-step format to build calculations that will add values to fields in a table.

Independent Practice: After completion of this lesson plan, there is an independent exercise for students to complete on their own that will replicate the process of building a database with a table with some of the fields initialized. Then the students will be asked to create an update query that will be used in conjunction with the expression builder to create formulas to be used in populating the fields in the table that were not initialized and are based on calculations of fields that were initialized to values.

Assessment: Students are graded on the Student Worksheet which they are asked to print and turn in. The grading grid gives a fairly accurate view of each student's level of mastery for creating an update query and using the expression builder to complete calculations. Those with a high grade certainly have a high level of mastery. Those with lower grades can be given the entire exercise from the lesson segment thru the Student Worksheet to learn the material.

Creating Update Queries Using the Expression Builder

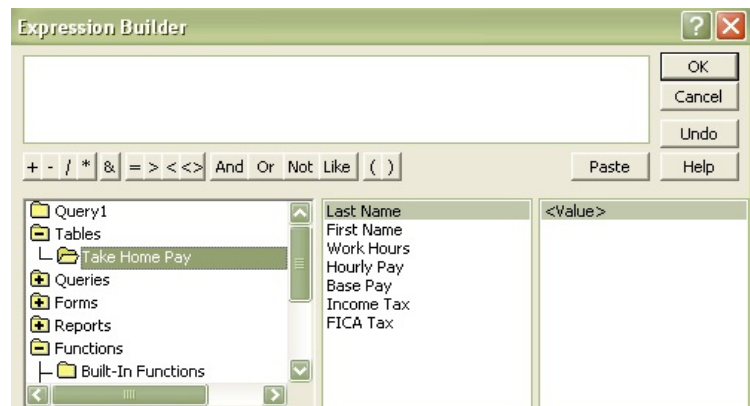
- I. Create a blank database in Access named **UpdQry-ExpBldr**, and create a new table in design view named **Take Home Pay**, with the following specifications:

	Field Name	Data Type	Description
🔑	ID	AutoNumber	Primary key
	Last Name	Text	15 Characters
	First Name	Text	15 Characters
	Work Hours	Number	Integer, General Number, 2 Decimal places
	Hourly Pay	Currency	2 Decimal places
	Base Pay	Currency	2 Decimal Places
	Income Tax	Currency	2 Decimal Places
	FICA Tax	Currency	2 Decimal Places
	Net Pay	Currency	2 Decimal Places

- A. Populate the **Take Home Pay** table with the following data:

ID	Last Name	First Name	Work Hours	Hourly Pay
1	Pepper	Bella	40	\$10.75
2	Hommy	Sal	35	\$9.50
3	Anya	Les	26	\$8.75
4	Marri	Cal A.	18	\$10.25
5	Gweenie	Lynn	38	\$12.25

- B. An update query is a type of query that uses functions and calculations to change data in fields of a table. This does NOT create a subset of a table as the Select Query will do, and it does NOT create a new table as the Make Table Query does, but it simply modifies data in an existing table, and this cannot be "undone."
- C. Create an Update Query using the Expression Builder, to update **Base Pay** using these specifications.
- From the Create ribbon, select Query Design, and then double click the **Take Home Pay** table to add it to the top of the query window.
 - Close the Show Table window.
 - Select Update Query from the Query Type section of the Design ribbon.
 - Double click on **Base Pay** in the table to send it to the first Field box in the lower part of the query.
 - Right click in the **Update To:** box and select **Build...** to open the Expression Builder.
 - The Expression Builder as shown on the right is broken into four sections.



- a. The top section is the workspace for the expression/equation.
 - b. The left column contains a menu of choices.
 - (1) Double click on Tables to see the **Take Home Pay** table.
 - (2) Double click on **Take Home Pay** to see fields of the table.
 - (3) If we had more than one table, they would also appear under Tables.
 - c. Double click on the Functions to view the possible choices, and then double click on Built-In Functions to see the available functions in the right-hand column and the available function groupings in the center column.
7. To build a calculation in the Expression Builder for **Base Pay**, use the following specifications.
- a. Double click Tables in the left column and then double click **Take Home Pay** table.
 - b. In the center column double click **Hourly Pay**, and the Expression Builder will put the field in the box at the top of the window as [Take Home Pay]![Base Pay]. This automatically puts the table name in square brackets followed by an exclamation point and then the select field name in square brackets.
 - c. Enter a multiplication operator (*) or click the * between the top box and the bottom columns of boxes if these are visible (only Office 2007).
 - d. Now double click **Work Hours** in the center column at the bottom to insert that field.
 - e. Click OK to accept the formula and close the Expression Builder and return to the query in the design view.
 - f. The **Update To:** box should now contain the following formula:
 [Take Home Pay]![Hourly Pay]*[Take Home Pay]![Work Hours]
 - g. Save the query as **PayQuery** and select Run. If the Run button has disappeared, reselect the Design Tab.
 - h. The results should appear like those on the right, and you can view them in two ways.
 - (1) On the Design ribbon, select Datasheet View to view the Query results.
 - (2) If the table is open, click the tab for the **Take Home Pay** table, and if it is not open, double click the **Take Home Pay** table on the left of the screen.
8. Create another Update Query in the design view to update both the **Income Tax**, and the **FICA Tax** fields in the **Take Home Pay** table.
- a. Be sure to click on Update as the Query Type on the Design ribbon.
 - b. Double click **Income Tax** and then **FICA Tax** from the table at the top .
 - c. In the **Update To:** box of **Income Tax**, right click and scroll down to select the Expression Builder.
 - (1) Double click on Tables in the lower left column of the Expression Builder to open the list of table in the center column and double click on **Take Home Pay**.
 - (2) Double click **Base Pay** in the center column to put it into the box at the top of the Expression Builder.
 - (3) Type in the multiplication operator (*) to put it in the upper box.
 - (4) Type in .15 after the asterisk to cause **Base Pay** to be multiplied by 15%.
 - (5) Click OK, and the following formula will appear in the Update To box of the query: [Take Home Pay]![Base Pay]*0.15
 - d. Right click in the **Update To:** box of **FICA Tax** and scroll down to select Expression Builder.

Base Pay ▾
\$430.00
\$332.50
\$227.50
\$184.50
\$465.50

- e. Double click on Tables in the lower left column of the Expression Builder to open the list of table in the center column and double click on **Take Home Pay**.
 - (1) Double click **Base Pay** in the center column to put it into the box at the top of the Expression Builder.
 - (2) Type in the multiplication operator (*) to put it in the upper box.
 - (3) Type in .072 after the asterisk to cause **Base Pay** to be multiplied by 7.2%
 - (4) Click OK, and the following formula will appear in the Update To box of the query: [Take Home Pay]![Base Pay]*0.072
 - f. Save the query as **TaxQuery** and run the query.
 - g. The results of this query should be those at the right.
- D. Go to the **Take Home Pay** table and view it in Datasheet View.
1. Delete the values in the **Base Pay**, **Income Tax**, and **FICA Tax** columns.
 2. Create an update query in design view to update several fields in the **Take Home Pay** table with one query.
 - a. Double click **Base Pay**, **Income Tax**, and **FICA Tax** to insert each in a column in the bottom half of the query window and use the Expression Builder to create the following Queries, but DO NOT RUN the query until all fields have been completed. These will have the same formulas as above, and you can build them with the Expression Builder as before or simply type them into the Update To: block. If you choose to type them in the block, the table name and exclamation point are not needed because you are working with only one table, and these are not needed if you are working with multiple tables if they don't have fields with the same name..
 - (1) **Base Pay** is calculated by multiplying **Work Hours** by **Hourly Pay**.
 - (2) **Income Tax** is calculated by multiplying **Base Pay** by 15%.
 - (3) **FICA Tax** is calculated by multiplying **Base Pay** by 7.2%.
 - b. Double click **Net Pay** to insert this field into the fourth column in the bottom half of the query window and use the Expression Builder to create the following calculation, but DO NOT RUN the query until all fields have been completed.
 - (1) Subtract **Income Tax** and **FICA Tax** from **Base Pay** to calculate **Net Pay**
 - (2) This formula will have the following appearance. Note that the table name and exclamation point are not needed when you are working with only one table or with multiple tables if they don't have fields with the same name.

Income Tax ▾	FICA Tax ▾
\$64.50	\$30.96
\$49.88	\$23.94
\$34.13	\$16.38
\$27.68	\$13.28
\$69.83	\$33.52

[Take Home Pay]![Base Pay]-([Take Home Pay]![Income Tax]+[Take Home Pay]![FICA Tax])

- c. Save the query as **AllQueries**, and run it one time.
 - (1) Run the query one time, and view the **Take Home Pay** table, and note that only the **Base Pay** field has been populated.
 - (a) **Base Pay** is a calculation using two fields that are already populated in the table.
 - (b) **Income Tax**, **FICA Tax**, and **Net Pay** cannot be populated until **Base Pay** has values in it, and that doesn't occur until this query has been run for the first time.

- (c) Access calculates all of the fields in a query at one time based on the exact same data. It will not make the calculations separately so the **Income Tax**, **FICA Tax**, and **Net Pay** were being calculated based on a zero value for Base Pay.
- (2) Run the query one more time, and view the **Take Home Pay** table, and note that all the remaining fields have been populated.
 - (a) Note that the **Net Pay** field has the same values as the Base Pay.
 - (b) **Income Tax** and **FICA Tax** are both zero at the start of this query, and all fields are updated in tandem, so effectively, you add value from **Base Pay** into **Net Pay** and then subtract 0 for the **Income Tax** and add 0 from the **FICA Tax** fields.
 - (c) The query must be run one more time to populate the **Net Pay** field after the **Income Tax** and **FICA Tax** fields are populated.
- (3) Run the Query for a third and final time and you will see that all of the fields have been populated.
- 3. The **Take Home Pay** table now contains the following calculated values:

Base Pay ▾	Income Tax ▾	FICA Tax ▾	Net Pay ▾
\$430.00	\$64.50	\$30.96	\$334.54
\$332.50	\$49.88	\$23.94	\$258.69
\$227.50	\$34.13	\$16.38	\$177.00
\$184.50	\$27.68	\$13.28	\$143.54
\$465.50	\$69.83	\$33.52	\$362.16

- 4. This particular query can be rerun over and over, and things will not change. However, there are situations in which the query has a cumulative affect. In this case, if an error has occurred and the query needs to be rerun, the fields in the table all need to be set to their original values, which is usually zero.
- 5. Remember, when updating more than one field you may have to run a query multiple times if a second field being updated relies on the update of the first field.

II. Independent Exercises.

- A. All students should have a copy of Page 7, the Student Worksheet, plus a computer with Access and blank paper for printing.
- B. Grade in printout of the Access table with the following criteria. Ignore font size.
1. Printout in portrait orientation. 5 pts
 2. Printout on one page. 5 pts
 3. Left and right margins 1". 5 pts
 4. Chart shown 3" from top of page (i.e. top margin 3"). 5 pts
 5. Title centered halfway between chart and top of page. 5 pts
 6. Date right justified on right margin halfway between chart and top of page. 5 pts
 7. Page 1 centered below chart 1 ½" from bottom edge of page. 5 pts
 8. Three rows of data shown and no more. 5 pts
 9. Nine columns of data shown and no more. 5 pts
 10. **YearsEmployed** matching key or equal to integer of (12/31/2010-StartDate)/365.25. 20 pts
 11. **CommissionEarned** matching key or equal to (Sales x **Rate**)/100. 20 pts
 12. **MonthlyEarning** matching key or equal to **Base Salary**/12 + **CommissionEarned**. 15 pts

Note: Answer key on the following page with note on grading typos for printed codes.
Deduct 2 points for each typo in the data.

III. Answer Key.


Sales

4/20/2011

ID	FullName	StartDate	YearsEmployed	Base Salary	Sales	Rate	CommisionEarned	MonthlyEaming
1	Cutis E. Counts	1/1/1998	12.0	40,000.00	228,396.92	12.30	28,092.82	31,426.15
2	Dobie Kruei	8/1/2004	6.0	32,000.00	336,827.98	6.30	21,220.16	23,886.83
3	Storm E. See	11/1/2006	4.0	26,000.00	187,329.00	4.75	8,898.13	11,064.79

IV. Student Worksheet. Students are to complete these exercises in Access and print their table.

- A. Exercise A. Create a blank database named **Exercises** with a table named **Sales** with the following specifications.

	Field Name	Data Type	Description
	ID	AutoNumber	Primary key
	FullName	Text	30 characters
	StartDate	Date/Time	Short date
	YearsEmployed	Number	Integer, standard, 1 decimal
	Base Salary	Number	Double, standard, 2 decimals
	Sales	Number	Double, standard, 2 decimals
	Rate	Number	Double, standard, 2 decimals
	CommissionEarned	Number	Double, standard, 2 decimals
	MonthlyEarning	Number	Double, standard, 2 decimals

1. Enter the following data in the **Sales** table. The rest of the fields will be calculated.

ID	FullName	StartDate	YearsEmployed	Base Salary	Sales	Rate
1	Cutis E. Counts	1/1/1998	12.0	40,000.00	228,396.92	12.30
2	Dobie Kruei	8/1/2004	6.0	32,000.00	336,827.98	6.30
3	Storm E. See	11/1/2006	4.0	26,000.00	187,329.00	4.75

2. Create an update query in design view to update the following fields.
- Update **YearsEmployed** using an Int function with all of the following as the parameter.
 - Subtract **StartDate** from 12/31/2010.
 - Divide the result by 365.25.
 - The formula should have the following appearance:
 $\text{Int}((\#12/31/2010\# - [\text{Sales-Upd-ExpBldr}]![\text{StartDate}]) / 365.25)$
 - Update the **CommissionEarned** by multiplying **Sales** times **Rate** and dividing the result by 100.
 - Update **MonthlyEarning** by dividing **Base Salary** by 12 and adding the **CommissionEarned**.
 - Save the query as **UpdQry**, and run the query until all fields are populated.
 - If you have a problem and need to start over, you might need to put 0.00 in the fields that are to be updated with the query.
3. View the table using Print Preview, and make the following changes to page layout.
- Use landscape orientation.
 - Use 3" top and bottom margins.
 - Use 1" left and right margins and size columns so that all fit on one page.
 - Print your table on one page and write your name in the upper right corner.

**Computer Application Lesson 4 to Prepare for
UIL Computer Application Contest**

Lesson Plan Title: Using the field codes to format text and date/time fields during a merge operation to Word from Access (or Excel).

Goal of Lesson: To have students understand the ins and outs of creating a database and table in Access and merging the data from the table into Word using formatting codes so that text and date/time fields are printed in predictable formats.

Grade Level/Course: BIM (9-12)

TEKS Addressed:

(4C,D) Compile information from primary and secondary sources in systematic ways using available technology (ELAR 110.42,43,44)

(1H) Compile written ideas and representations, interpret empirical data into reports, summaries, or other formats, and draw conclusions. (ELAR 110.46)

(5C) Summarize information from text through the use of outlines, study guides, or learning logs (ELAR 110.48)

(4G,7A) Follow oral and written directions. (Bed 120.22)

Skill in use of database features, formulas and functions

(4A, 3A, 6A) Perform calculations involving money, time, space, materials, and data. (MktEd 124.12,24)

(4G,7A) Follow oral and written instructions. (Bed 120.22,42)

(2E) Use computer technologies to record, locate, analyze, present, and exchange information. (TechEd 124.47)

§130.272. Principles of Information Technology (One-Half to One Credit).

(7) The student applies word-processing technology. The student is expected to:

(A) identify the terminology associated with word-processing software and its functions;

(B) improve the touch-system skill using the keyboard and keypad to input data;

(C) edit a variety of text documents using functions such as pagination, appropriate white space, tab settings, and font style, size, and color;

(9) The student applies database technology. The student is expected to:

(A) identify the terminology associated with database software and its functions;

(B) create, populate, edit, maintain, and save database files;

(D) perform data management procedures such as locating, sorting, searching, querying, organizing, and outputting data;

(F) produce organized reports with calculated figures.

§130.114. Business Information Management I (One to Two Credits).

(5) The student applies word-processing technology. The student is expected to:

(B) improve touch-system skills using the keyboard and keypad to input data;

(C) use hardware and software flexibility needed to produce documents to address different computer applications

Computer Application Lesson 4-continued

(6) The student identifies database software to create databases that facilitate business decision making. The student is expected to:

- (A) explain the principles of data analysis;
- (B) explain the nature of tools that can be used to access information in the database system;
- (C) choose appropriate software;
- (D) define fields and type of data;
- (E) enter database structure;
- (G) analyze company's data requirements;
- (H) design a database to meet business requirements; and

(7) The student applies data entry techniques to enter information in databases. The student is expected to:

- (A) access information in the database system;
- (B) build data in a data warehouse;
- (C) create a meaningful data set;
- (D) enter data into databases, tables, and forms;
- (E) edit data in databases, tables, and forms;

(8) The student uses commands to retrieve data and create reports from databases. The student is expected to:

- (A) retrieve data from tables and queries;
- (C) create and print reports; and
- (D) manipulate data in the database management system.

Overview of Lesson: This lesson will present a step-by-step approach to creating a database and table in Access and merging the data in the table into a document that students will create in Word. The primary point of this exercise is to visit the formatting of text and date/time fields during a merge operation from an Access table giving predictable results.

Materials Needed: Students will need a computer with Access, a printer, and the exercise on Pages 5 and 6 of the written instructions.

Procedures and Activities: This is a hands-on operation that guides students thru building a basic database and table and merging directly to Word to create a document that uses fields from the table and formats them in predictable text and date/time formats.

Independent Practice: After completion of this lesson plan, suggest that students work through the Student Exercise on pages 5 and 6 of the instructions to build a database, a table, and merge to Word creating a document in which they supply field codes to format text and date/time fields.

Assessment: The Student Exercise can be graded to determine if they have successfully understood the process of merging data from Access into Word with specific text formatting. If students can then suggest additional ways to format the merged fields to add more complex primary and secondary field codes, they are excelling.

Field Codes for Text, Dates, & Time

- I. This lesson explains how to use field codes or formatting switches, as they are sometimes called, to format data being merged into Word from either Access or Excel. This doesn't seem too important when we're doing a short exercise of merging half a dozen letters. We can manually format each one, but if we had a list like companies in the "real world" for several thousand names for which we are merging data, it would be an impossible task to go thru each letter and correct the formatting.

- A. Create a database in Access named **Field Codes** and create a table named **TextDateTime** with the specifications shown on the right.

Field Name	Data Type	Description
ID	AutoNumber	Primary Key
Last Name	Text	15 characters
First Name	Text	15 characters
Birth Date	Date/Time	Short date
Start Date	Date/Time	Short date
Arrival Time	Date/Time	Medium time

- B. Enter the following data into the **TextDateTime** table.

ID	Last Name	First Name	Birth Date	Start Date	Arrival Time
1	Bellum	Sara	2/6/1979	8/15/2000	6:30 AM
2	Lear	Shanda	11/8/1980	3/1/2004	12:30 PM
3	O'Shea	Rick	7/12/1983	12/15/2007	6:30 PM
4	Carr	Parker	9/23/1985	7/1/2009	12:30 AM

- C. With the **TextDateTime** table open, select the More pull-down menu on the External Data ribbon in the Export block and select Merge it with Microsoft Office Word. (Ctrl, X, G, M) or for some versions select WordMerge in the Export block on the External Data ribbon.

1. Select the radio button to create a new document and then link to it.
2. The document should open in Word.

- D. Enter the following data followed by a colon and tab; then insert the appropriate field name from the database created above using the following directions.

1. On the Mailings ribbon, pull down the menu for the Insert Merge Field from the Write & Insert Fields.
2. Select the appropriate field, such as Last_Name.
3. Add a comma and space after the field name, and pull down the menu again to insert the **First_Name** field.
4. Continue until your document has the appearance below.
5. Now on the Mailings ribbon in the Preview Results block, click Preview Results, and you should see the following.

Employee: Bellum, Sara
 Birth Date: 2/6/1979
 Start Date: 8/15/2000
 Arrival Time 6:30:00 AM

6. Add a field code to change the formatting of the **Last_Name** field to print as all caps using the following specifications.
 - a. Right click on the **Last_Name** field and select Edit Field from the pull-down menu.

- b. From this window, some types of field codes can be selected fairly easily, such as uppercase, lowercase, first capital, and title case.
 - c. Click on Uppercase to make the **Last Name** field print in all caps.
 - d. Now click on Field Codes in the lower left of the Field window. This will show you the merge code with the field code added behind the field name, as the following:
MERGEFIELD Last_Name * Upper
 - e. Click OK in the lower right of the window.
 - f. Your merge code will now have the following appearance:
Employee: «LAST_NAME», «First_Name»
 - g. Now hit Alt-F9 to change the merge code view to shown the MERGEFIELD view. Your Last Name merge field has the following appearance in this format.
{MERGEFIELD Last_Name*Upper}
 - (1) Each field is enclosed in curly brackets in this format.
 - (2) Spaces in field names are replaced with underline characters.
 - (3) The * is notification to Word that this is a field code. (There is a blank space before and after this *, but these are not necessary.)
 - (4) Hit Alt-F9 again to get out of the MERGEFIELD view.
 - h. Click on the Preview Results again, and your name fields will have the following appearance: BELLUM, Sara
 - i. Note that when you preview results, you need to get back into the merge code format before going to the MERGEFIELD format or the updates to fields codes will not show up. If you apply a field code and go directly back to preview results without going thru the merge code format, the same data will appear. You can then right click and select Update Field to show the recent changes.
7. Other frequently used options for formatting text include the following.
- a. * Lower would make the field print in all lowercase letters.
 - b. * FirstCap would make the first letter of the field print as a capital letter. If the field contained several words, only the first word would be capitalized.
 - c. * Caps would capitalize the first letter of every word, including articles, prepositions, and such that are not usually capitalized in a title.
8. More than one option may be used in a field code. For instance, you could use * Lower to make a field all lowercase and then follow that by * Title case. (Codes are separated by a space). This would reset all letters to lowercase and then capitalize the first letter of each words in the phrase.
- E. Apply field codes to the Date and Time fields.
1. The date code is \@ following by characters describing what date format you want.
 2. Characters used in describing a skeleton of a date include the following:
 - a. M - month (note: this may not be lower case or it means minute)
 - b. D or d - day
 - c. Y or y - year
 - d. H or h - hour
 - e. m - minute (note: this may not be capitalized or it means month)
 - f. S or s - second
 - g. AM/PM or am/pm - adds indication of AM or PM immediately after the time.
 3. Using a single M, D, d, Y, y, H, or h will give you single digit months, days, and hours if that is what is represented. If any of these is greater than 9, they will appear as two digits.

4. Using MM, DD, dd, YY, yy, HH, hh, or mm will give you 2 digit months, days, hours, and minutes, and any less than 9 will appear as 01, 02, etc. These can be used with slashes between entities, as MM/DD/YY
5. Using MMM, DDD, ddd will give you the abbreviation for the month or the day of the week, such as Jan, Feb or Mon, Tue, etc. Adding the period, comma, and spaces in the formatted date requires using a quotation mark at the front of the date skeleton. Without the quotation mark indicating the extra symbols are present, the date will stop formatting at the first extra symbol of text entry.
 - a. Use a quotation mark (") at the front of the code, \@ "MMM, DDD, yyyy
 - b. To add a period after the abbreviated month or day, simply add a period in the appropriate place, and to add spaces in the appropriate places, simply insert a space. (Any time you have "extra" symbols or punctuation marks in a field code, the field code data following the \@ or \# or * needs to start with a quotation mark. If you want additional text in a field code, enclose it in single quotation marks.)
6. Finally, using MMMM, DDDD, dddd, YYYY, or yyyy will give you the month or day spelled out (January, Monday) or the 4-digit year.
7. The following exercises use our new knowledge about formatting dates and times.
 - a. Looking at the merge codes for our Word document, right click on the **Birth Date** merge field and select edit field, and then click on Field Codes at the bottom left of the new window.
 - (1) Enter the following field code after the merge field: \@ M/d/y
 - (2) This is the simplest date format. It requires no quotation mark in front of the field code following the \@ as it contains no special characters or text.
 - (3) Click OK, then preview results.
 - (4) **Birth Date** should display as 2/6/79.
 - b. Now with merge codes showing, click ALT F9 to view MERGEFIELD codes.
 - (1) Between "Start_Date" and the curly bracket }, enter the following field code:
 \@ "DDDD, MMMM d, yyyy
 - (2) Click ALT F9, and then Preview Results.
 - (3) **Start Date** should display as Tuesday, August 15, 2000.
 - c. Now view the MERGEFIELD codes.
 - (1) Change the field code for **Start Date** to the following:
 \@ "'Birthday is' DDDD, MMMM d, yyyy
 - (2) **Start Date** should display as Birthday is Tuesday, August 15, 2000.
 - (3) Note that using DDDD makes the day of the week print out entirely; MMMM makes the month print out entirely; the single 'd' makes the date print as a one- or two-digit number; and the yyyy makes the year print as a 4-digit number.
8. Now with merge codes showing, click ALT F9 to view MERGEFIELD codes.
 - a. Add the following code between "Arrival_Time" and the curly bracket }:
 \@hh:mmam/pm Preview the results
 - b. **Arrival Time** should have the following appearance: 06:30AM
 - c. If you want a space between the time and AM, use the following sequence.
 - (1) Add the following code between "Arrival_Time" and the curly bracket }:
 \@ "hh:mm' 'am/pm
 - (2) Now the hours and minutes will have the following appearance: 06:30 AM

II. Independent Exercises.

- A. All students should have a copy of Pages 5 & 6 with the Student Worksheet and Summary of Text and Date/Time field codes, a computer with Access and Word, and blank paper.
- B. Grade in Word with the following criteria. Ignore font size and margins. If there are typos, you may deduct them from the final grade.
- C. Calculate a grade based on the following for a total of 100 points. Grade all records for the same field at the same time. If only one record is correct, student gets only ½ credit. (Take off for typos for all data in the second row of merged data for each pair that does not match key, including upper/lowercase, for all incorrect or missing parts of the printed field codes for which students received credit in the grading. i.e. if the date was almost right, but had a missing space that was not graded, that is a typo.):
 1. Top, left, and right margins shown as 1"..... 5 pts
 2. Columns of data shown vertically left aligned..... 5 pts
 3. All merged names on first row of every pair in appropriate caps/lowercase like key..... 5 pts
 4. Title shown on first row of every pair of merged fields..... 5 pts
 5. **StartDate-Time** merged field on first row approximately matching format of key..... 5 pts
 - a. Period shown after Jan and comma after day..... 5 pts
 - b. Text (at) shown between year and time..... 5 pts
 - c. Time shown as hh:mm with no seconds..... 5 pts
 - d. Time shown as with AM/PM at the end..... 5 pts
 - e. Time shown with space between time and AM/PM..... 5 pts
 6. **ShiftStart-Time** merged field on first row of every pair shown as with minutes and second both shown as two digits. 5 pts
 - a. **ShiftStart-Time** shown as with AM/PM following time. 5 pts
 - b. **ShiftStart-Time** shown as with space between time and AM/PM. 5 pts
 7. Printed field name and code for **FullName** approximately matching key..... 10 pts
 8. Printed field name and code for **Title** approximately matching key..... 5 pts
 9. Printed field name and code for **StartDate-Time** approximately matching key..... 10 pts
 10. Printed field name and code for **ShiftStart-Time** approximately matching key..... 10 pts

III. Answer Key for Scoring.

FullName	Title	StartDate-Time	ShiftStart-Time
E. C. Street	Senior sales manager	Jan. 1, 1998 at 8:00 AM	06:30 AM
E. C. STREET	senior sales manager	1/1/1998 8:00:00 AM	6:30:00 AM
Curt Sanwhey	Salesman advanced	Aug. 1, 2004 at 9:30 AM	12:30 PM
Curt SANWHEY	salesman advanced	8/1/2004 9:30:00 AM	12:30:00 PM
Geoff L. Tavish	Salesman beginning	Nov. 1, 2006 at 6:30 PM	06:30 PM
Geoff L. tavish	salesman beginning	11/1/2006 6:30:00 PM	6:30:00 PM

"FullName" * lower *Caps

"Title" * FirstCap

"StartDateTime" \@ "MMM. d, yyyy" at 'h:mm AM/PM

"ShiftStartTime" \@ "hh:mm AM/PM

III. **Student Worksheet.** Students are to complete the following exercises after entering the following worksheet in Access then print their Word document in landscape orientation.

A. Part A. Create a blank database named **Exercises** with a table named **SalesForce** with the specifications to the right.

1. Enter the following data in the **SalesForce** table.

Field Name	Data Type	Description
ID	AutoNumber	Primary key
FullName	Text	30 characters
Title	Text	25 characters
StartDate-Time	Date/Time	general date
ShiftStart-Time	Date/Time	general date

ID	FullName	Title	StartDate-Time	ShiftStart-Time
1	E. C. STREET	senior sales manager	1/1/1998 8:00 AM	6:30:00 AM
3	Geoff I. tavish	salesman advanced	11/1/2006 6:30 PM	12:30:00 PM
4	Curt SANWHEY	salesman beginning	8/1/2004 9:30 AM	6:30:00 PM

2. Merge the table to Word creating a new document and linking to it.
3. The Word document should have the following specifications.
 - a. Use landscape orientation, and 1" margins all around, and use single spacing for the entire document.
 - b. Enter the following separated by tabs: **FullName**, **Title**, **StartDate-Time**, and **ShiftStart-Time**. These will be headings for the merged data that we will add.
 - c. Merge the appropriate field on the line immediately below each column heading. Then copy the first line of merged data and paste it below the first line so that the document looks like the one seen below. (Note that you can align columns after you have merged data into the fields.

FullName	Title	StartDate-Time	ShiftStart-Time
«FullName»	«Title»	«StartDateTime»	«ShiftStartTime»
«FullName»	«Title»	«StartDateTime»	«ShiftStartTime»

- d. At the end of your document leave two blank lines after the last merged field
 - e. Preview Results for your merged fields. You will see that the same record appears on both lines this is fine.
- B. Part B. Use field codes to format the first row of merged fields.
1. Format the **FullName** field by using a primary and secondary field code.
 - a. From the Merge Code format, click Alt F9 to view MERGEFIELD format.
 - b. For the first **FullName** merge field add the primary field code to make everything lowercase.
 - c. After the primary field code in the **FullName** field, add the secondary field code to indicate capitalization of the first letter of each word. E. C. STREET should now display as E. C. Street.
 2. For the first **Title** merge field add the primary code to print only the first word of the title capitalized, as Senior sales manager
 3. Format the **StartDate-Time** so that it has the following appearance for the first record shown: Jan. 1, 1998 at 08:00AM
- C. After everything is formatted appropriately, complete the following.
1. On the last line below the two blank lines following your merged data, select next Record from the pull down menu on the Mailing ribbon in the Write & Insert Fields block labeled Rules, and insert it on the first line.

- a. Copy everything from the **FullName** label on the first row thru the merged **ShiftStart-Time** merged field and paste to the right of the next Record code.
 - b. Leave two more blank lines after this record.
2. On the next line copy from the next Record code thru the merged **ShiftStart-Time** field and paste it below the two blank lines.
3. Preview Results of all of your data to be sure everything is formatting correctly.
4. Align the columns so that the printout is orderly.
5. View the data in the MERGEFIELD format, and select the field name and field code for each of the 4 fields and copy and paste below the data. These will be graded.
6. Print your data.

IV. Summary of Text and Date/Time field codes.

A. Text may be formatted with the following:

1. *** Upper** - this converts everything in the merged field to all caps.
2. *** Lower** - this converts everything in the merged field to lowercase.
3. *** Caps** - this makes the first letter of each word in the merged field a capital letter. (If other letters in the field are capitalized, these are not changed. i.e. if your field contained SALES manager and was formatted for Caps, it would become SALES Manager. It would make the first letter of each word uppercase, but would not change other letters.)
4. *** FirstCap** - this makes the first letter only of the field a capital. (If other letters in the field are capitalized, these are not changed. i.e. if your field contained SALES mANAgEr and was formatted for FirstCap, it would still be SALES mANAgEr. It would make the first letter of the first word uppercase, but would not change other letters.)

B. Multiple codes may be used to create a specific condition.

1. *** Lower** could be used to make everything in a field lowercase, in which case the field SALES mANAgEr would become sales manager.
2. This can be followed by an additional field code to make the first letter of every word capitalized, and the code would appear *** Lower * Caps**
3. This effectively capitalizes the first letter of every word and makes the remaining letters lowercase.

C. Date & Time codes may be formatted with the following:

1. **\@** - this indicates that the data following is to format a date or time field.
2. Characters used in describing a skeleton of a date include the following:
 - a. **M** - month (note: this may not be lower case or it means minute)
 - b. **D or d** - day
 - c. **Y or y** - year
 - d. **H or h** - hour
 - e. **m** - minute (note: this may not be capitalized or it means month)
 - f. **S or s** - second
 - g. **AM/PM or am/pm** - adds indication of AM or PM after the time.
3. Using a single M, D, d, Y, y, H, or h will give you single digit months, days, and hours if that is what is represented. If any of these is greater than 9, they will appear as two digits.
4. Using MM, DD, dd, YY, yy, HH, hh, or mm will give you 2 digit months, days, hours, and minutes, and any less than 9 will appear as 01, 02, etc. These can be used with dashes or slashes between entities, as MM/DD/YY or MM-DD-YY.
5. Using MMM, DDD, ddd will give you the abbreviation for the month or the day of the week, such as Jan, Feb or Mon, Tue, etc.
6. Use a quotation mark (") at the front of the code, as in \@ "MMM, DDD, yyyy to add a period after the abbreviated month or day or to add extra text enclosed in single quote marks in the formatted field.

Computer Application Lesson 5 to Prepare for

UIL Computer Application Contest

Lesson Plan Title: Using the field codes to format numeric fields during a merge operation to Word from Access (or Excel).

Goal of Lesson: To have students understand the ins and outs of creating a database and table in Access and merging the data from the table into Word using formatting codes so that numeric fields are displayed in predictable numeric formats or in numeric formats with added text, or in text formats.

Grade Level/Course: BIM (9-12)

TEKS Addressed:

(4C,D) Compile information from primary and secondary sources in systematic ways using available technology (ELAR 110.42,43,44)

(1H) Compile written ideas and representations, interpret empirical data into reports, summaries, or other formats, and draw conclusions. (ELAR 110.46)

(5C) Summarize information from text through the use of outlines, study guides, or learning logs (ELAR 110.48)

(4G,7A) Follow oral and written directions. (Bed 120.22)

Skill in use of database features, formulas and functions

(4A, 3A, 6A) Perform calculations involving money, time , space, materials, and data. (MktEd 124.12,24)

(4G,7A) Follow oral and written instructions. (Bed 120.22,42)

(2E) Use computer technologies to record, locate, analyze, present, and exchange information. (TechEd 124.47)

§130.272. Principles of Information Technology (One-Half to One Credit).

(7) The student applies word-processing technology. The student is expected to:

(A) identify the terminology associated with word-processing software and its functions;

(B) improve the touch-system skill using the keyboard and keypad to input data;

(C) edit a variety of text documents using functions such as pagination, appropriate white space, tab settings, and font style, size, and color;

(9) The student applies database technology. The student is expected to:

(A) identify the terminology associated with database software and its functions;

(B) create, populate, edit, maintain, and save database files;

(D) perform data management procedures such as locating, sorting, searching, querying, organizing, and outputting data;

(F) produce organized reports with calculated figures.

§130.114. Business Information Management I (One to Two Credits).

(5) The student applies word-processing technology. The student is expected to:

(B) improve touch-system skills using the keyboard and keypad to input data;

(C) use hardware and software flexibility needed to produce documents to address different computer applications; and

(6) The student identifies database software to create databases that facilitate business decision making. The student is expected to:

(A) explain the principles of data analysis;

(B) explain the nature of tools that can be used to access information in the database system;

(C) choose appropriate software;

Computer Application Lesson 5-continued

- (D) define fields and type of data;
 - (E) enter database structure;
 - (G) analyze company's data requirements;
 - (H) design a database to meet business requirements; and
- (7) The student applies data entry techniques to enter information in databases. The student is expected to:
- (A) access information in the database system;
 - (B) build data in a data warehouse;
 - (C) create a meaningful data set;
 - (D) enter data into databases, tables, and forms;
 - (E) edit data in databases, tables, and forms;
- (8) The student uses commands to retrieve data and create reports from databases. The student is expected to:
- (A) retrieve data from tables and queries;
 - (C) create and print reports; and
 - (D) manipulate data in the database management system.

Overview of Lesson: This lesson will present a step-by-step approach to creating a database and table in Access and merging the data in the table into a document that students will create in Word. The primary point of this exercise is to visit the formatting of numeric fields during a merge operation from an Access table so that the results are displayed as specifically formatted numeric fields, as text fields, or as a combination of a numeric field with embedded text.

Materials Needed: Students will need a computer with Access, a printer, and the exercise on Pages 7-9 of the written instructions.

Procedures and Activities: This is a hands-on operation that guides students thru building a basic database and table and merging directly to Word to create a document that uses numeric fields from the table and formats them in predictable numeric, text, or fields with combined numeric and text formats.

Independent Practice: After completion of this lesson plan, suggest that students work thru the Student Exercise on the last pages 7-9 of the instructions to build a database, a table, and merge to Word creating a document in which they supply field codes to format various numeric fields so that they display as specific number formats, text formats, or a combination of the two formats.

Assessment: The Student Exercise can be graded to determine if they have successfully understood the process of merging from Access to Word and formatting the numeric fields in specific numeric formats or as a text variation to the field. If students can then suggest additional ways to format the merged fields to add more complex primary and secondary field codes, they are excelling.

Field Codes for Numeric Fields

- I. This lesson explains how to use field codes or formatting switches, as they are sometimes called, to format data being merged into Word from either Access or Excel. This doesn't seem too important when we're doing a short exercise of merging half a dozen letters. We can manually format each one, but if we had a list like companies in the "real world" for several thousand names for which we are merging data, it would be an impossible task to go thru each letter and correct the formatting. The most glaring problem here are the numbers that are merged into a document. Even though you may have formatted a number with \$ signs, commas, and 2 decimals, when it is merged into a Word document, it comes in as a raw number with no formatting and often several decimal positions.

- A. Create a database in Access named **Format** and create a table named **Numeric** with the following specifications.

Field Name	Data Type	Description
ID	AutoNumber	Primary key
Last Name	Text	15 characters
First Name	Text	15 characters
Annual Salary	Number	Double, standard, 0 decimals
Monthly Salary	Number	Double, standard, 0 decimals
Starting Year	Number	Integer, fixed, 0 decimals
SS Number	Text	9 characters, input mask for SS Number
Phone Number	Text	10 characters, input mask for phone number

- B. Note that SS Number & Phone Number should store data without symbols in the text.

- C. Enter the following into the **Numeric** table.

ID	Last Name	First Name	Annual Salary	Monthly Salary	Starting Year	SS Number	Phone Number
1	Bellum	Sara	48,500		2000	222-33-4444	(222) 334-4444
2	Lear	Shanda	44,800		2004	333-44-5555	(333) 445-5555
3	O'Shea	Rick	36,750		2007	444-55-6666	(444) 556-6666
4	Carr	Parker	31,000		2009	555-66-7777	(555) 667-7777

- D. Create an update query to update the **Monthly Salary** field.

1. On the Create ribbon, select Query Design (or type Alt, C, QD) to quickly open a query.
2. Add the **Numeric** table to the query.
3. On the Design ribbon, click Update in the Query Type box.
4. Double-click on the **Monthly Salary** field to push it below into a column for updating.
 - a. On the Update To: row, add the following to divide the **Annual Salary** by twelve to derive the **Monthly Salary**: **[Annual Salary]/12**.
 - b. Click on the Run icon in the Design ribbon in the Results box to run the query and update the **Monthly Salary** field.
5. View your **Numeric** table to ensure that all fields have been populated.
6. **Monthly Salary** should have the data to the right.

Monthly Salary
4,042
3,733
3,063
2,583

- E. With the **Numeric** table open, select the More pull-down menu on the External Data ribbon in the Export block and select Merge it with Microsoft Office Word. (Ctrl, X, G, M) Note: in some systems there is a WordMerge button that does this in the Export block.

1. Select the radio button to create a new document and then link to it.
2. The document should open in Word.

- F. Enter the following data followed by a colon and tab; then insert the appropriate field name from the database created above using the following directions.
1. Type in Employee: and a tab.
 2. On the Mailings ribbon, pull down the menu for the Insert Merge Field from the Write & Insert Fields.
 3. Select the appropriate field, such as **Last_Name**, then add a comma and space after the field name, and pull down the menu again to insert the **First_Name** field.
 4. Continue until your document has the appearance below.
 5. Now on the Mailings ribbon in the Preview Results block, click Preview Results, and you should see some of the obvious problems in the number formatting.
 6. Field codes or format switches for fields merged into Word can rectify these problems. These codes are embedded in the merge field code.
 7. These are the basic, acceptable symbols that are used in a format skeleton.

Employee:	«Last_Name», «First_Name»
Annual Salary:	«Annual_Salary»
Monthly Salary:	«Monthly_Salary»
Starting Year:	«Starting_Year»
SS Number:	«SS_Number»
Phone Number:	«Phone_Number»
Employee:	Bellum, Sara
Annual Salary:	48500
Monthly Salary:	4041.6666666666665
Starting Year:	2000
SS Number:	222334444
Phone Number:	2223344444

 - a. 0 - specifies required positions to display. 00.00 would show a 2-digit number with 2 decimals. If field does not have a numeric value, a zero is substituted.
 - b. # - specifies required positions to display. ##.## will show a 2-digit number with 2 decimal places, but if a numeric value is missing, the # is replaced with a blank.
 - c. x - specifies where to stop showing values. If a field has a value 9876 and has the format code \# \$x##, the field would display as \$876. If the value had only been 876, result would have shown all the numbers, as \$876. (The x is used as the last numeric character and everything to the left of the x is removed.)
 - d. . - determines decimal point position.
 - e. , - is a comma separator as is 10000 formatted as \# ###,### would be 10,000.
 - f. - - adds a minus sign for a negative number and leaves a space for a positive number.
 - g. + - adds a plus sign to a positive number, a minus sign for a negative number, and a blank for zero.
 - h. % - adds a percent sign to a number.
- G. To add a field code to the **Annual Salary**, view the merge code format, click Alt F9, and add the following code inside the right bracket: \# \$,#.00 (You must click Alt F9 again before going to preview for the field code to be applied).
1. The merge code should look like this: {MERGEFIELD Annual_Salary \# \$,#.00}
 2. Placement of the \$ sign adjacent to the , (or to a # sign or 0) ensures that the dollar sign will be immediately adjacent to the first digit of the field, no matter how many digits are in the field to the left of the decimal.
 3. The comma ensures that the thousands will be separated with commas.
 4. The decimal places the whole numbers to the left and the decimal values to the right.
 5. The 00 ensures that there will be two decimal positions. If the value in the decimal positions is .6, it will print as .60, and if the value of the decimal positions is zero, it will print as .00. If we had used # signs instead of 0 for the decimal positions, .6 would print at .6 and .00 would not print at all, only the period.

- H. Add the following field code to **Monthly Salary**: \# \$000,000,000.00
1. Your results will show \$000,004,041.67
 2. Note that there is a digit shown at every place that there was a 0 in your skeleton. Too many to the left of the decimal.
 3. Now change your skeleton to show # signs instead of 0's to the left of the decimal.
 4. Now your result shows \$ 4,041.67
 5. There are 5 spaces between the \$ sign and the first digit in the value shown because we had 9 # signs, and our value took up only 4 of them. If the number had been long enough to take up all the positions or longer than that of the # signs to the left of the decimal, the \$ sign would have been immediately adjacent to the first digit.
- I. Now we're going to apply text field codes to numeric fields.
1. The following are possibilities for editing numeric fields.
 - a. * **CardText** (displays as cardinal text, as twenty-one)
 - b. * **CardText** * **Cap** (displays as capitalized cardinal text. There are several possibilities for this secondary field code, including Upper, Lower, FirstCaps, etc.)
 - c. * **DollarText** (displays as cardinal text, twenty-one, and inserts "and" at the decimal place and displays the first two decimals (rounded) as Arabic numerators over 100. (A secondary field code can be added to alter capitalization)
 - d. * **roman** (displays as Roman numerals, as XXXI)
 - e. * **Arabic** (displays as Arabic cardinal numerals, as 21; ArabicDash inserts hyphen before and after number)
 - f. * **Hex** (displays as hexadecimal numbers, 1F)
 - g. * **OrdText** (displays as ordinal text, thirty-first. A secondary field code can be added to specify different capitalization or date format)
 - h. * **Ordinal** (displays as ordinal Arabic numerals, as 31st)
 2. In the MERGEFIELD view, replace the field code for **Annual Salary** with the following: * CardText
 - a. The * tells Word to display the incoming numeric value as text.
 - b. The CardText field says to spell out the numeric value.
 - c. Preview the result which should have the following appearance:
forty-eight thousand five hundred
 3. In the MERGEFIELD view, replace the same field code for **Annual Salary** with the following: * DollarText
 - a. Preview the results.
 - b. This should display as forty-eight thousand five hundred and 00/100, which is the way a value is written out on a check.
 4. In the MERGEFIELD view, replace the field code for **Starting Year** to be the following: * Roman.
 - a. View the results, and the field now is displayed as MM.
 - b. Note that Roman numerals only are available up to 3999. If you try to format a number larger than that, Word will give you an error message on the field.
 5. Now replace the field code in the MERGEFIELD view for **Starting Year** to be * Hex.
 - a. This will display the hexadecimal equivalent of the value.
 - b. The displayed value should show as 7D0.
 - c. This is the value of the **Starting Year** as represented in base 16.
 6. Replace the field code in the MERGEFIELD view for **Starting Year** to be * OrdText.
 - a. The OrdText code present data as an ordinal number, as first, second, third, etc.
 - b. This should display as two thousandth.
 7. Now replace the field code in the MERGEFIELD view for **Starting Year** to be * Ordinal.
 - a. This also displays the number in an ordinal format, but not spelled out.
 - b. This should display as 2000th.

8. Now replace the field code in the MERGEFIELD view for **Starting Year** to be * Arabic.
 - a. This displays a number as an Arabic number, which is what we ordinarily use.
 - b. This should display as 2000.
9. Now replace the field code in MERGEFIELD view for **Starting Year** to be * ArabicDash
 - a. This displays a number as an Arabic number with a dash and blank on each side of the number.
 - b. This should display as - 2000 -.
10. To use multiple codes, you simply put them in an appropriate order.
 - a. Again in the MERGEFIELD view using the **Annual Salary** with the field code set to * DollarText, add a secondary field code to print the entire result in uppercase.
 - (1) Change the field code to be * DollarText * Upper
 - (2) Your result should be FORTY-EIGHT THOUSAND FIVE HUNDRED AND 00/100
 - b. In the same field code replace * Upper in the field code with * FirstCap, and view the field which should appear as Forty-eight thousand five hundred and 00/100.
 - c. In the same field code replace * FirstCap in the field code with * Caps, and view the field which should appear as Forty-Eight Thousand Five Hundred And 00/100 (Note that all words are capitalized, even the preposition "And" is capitalized.)
 - d. This ability of using multiple codes is often used with data fields that are not all consistent in a database. For instance, if some of your names are all caps and some are upper/lowercase, this ability is used to make them all standard.
 - (1) To do this, the first field code should be * lower which will convert all characters in the field to lowercase.
 - (2) The secondary field code can be * Caps to capitalize every word in the field.
- J. This segment modifies the formatting of numeric fields to insert special characters, such as hyphens in a social security number or parentheses and hyphens in a phone number.
 1. In the MERGEFIELD view, add the following field code for the **SS Number** field:


```
\# ###'-##'-####
```

 - a. This operates like a numeric field code because it begins with \#.
 - b. The remaining # signs will each be replaced by a number from the **SS Number** field.
 - c. After the first # signs, there is a '-' which will insert a hyphen in that position of the resulting field.
 - d. After two more digits, there is another '-' which will insert another hyphen at that point in the result.
 - e. The result should be 222-33-4444.
 2. In the MERGEFIELD view, add the following field code for the **Phone Number** field:


```
\# "Cell: (###) '###'-####
```

 - a. Note that \# indicates this is a numeric field code.
 - b. The quotation mark " preceding the skeleton indicates that some "extra" characters or text will be added to this field code, and in this case the extra text is the Cell: preceding the phone number and the blank space following the right bracket. (Note: no closing quotation is necessary.)
 - c. Thus **Cell:** (will appear before the first number in the result.
 - d. The ') " will follow the third number in the field and be followed by a blank space.
 - e. The remaining #'s will be replaced with numbers in the result, and the '-' will stay in the same location in the result as it is in the skeleton.
 - f. The result should be Cell: (222) 334-4444. (Note there is text before the phone number and a space between the right bracket and the next digit
- K. Review all the options for numeric formatting, text formatting, and date and time formatting, as well as text formatting for numeric fields.

II. Independent Exercises - Grading

- A. All students should have a copy of Pages 7 thru 9, Student Worksheet, Summary of Numeric Field Codes, and a computer with Access and Word with blank paper for printing.
- B. Grade in Word with the following criteria. Ignore font size and margins. If there are typos, deduct them from the final grade.
- C. Calculate a grade based on the following for a total of 100 points. Grade all records for the same field at the same time. If only one record is correct, student gets only ½ credit. (Take off for typos particularly for the second row of merged data for each record. Also count typos for columns not being aligned, for a printout not being in landscape, etc.):
 1. FullName on first row of each pair shown with caps/lowercase of key. 5 pts
 2. Sales on first row of each pair shown approximately matching key. 5 pts
 - a. \$ shown immediately before first digit. 5 pts
 - b. Commas shown. 5 pts
 - c. 2 decimals position shown for **Sales** on all records. 5 pts
 3. Rate on first row of each pair matching key with 1 decimal place, space and % sign. 5 pts
 4. CommissionEarned on first row of each pair approximately matching key. 5 pts
 - a. Everything shown in lowercase. 5 pts
 - b. Decimal positions shown as "and 72/100" with appropriate values. 5 pts
 5. Start Year on first row of each pair shown as Roman numerals. 5 pts
 6. **YearsEmployed** on first row of each pair shown as 1 or 2-digit number. 5 pts
 - a. "years" shown following numeric values. 5 pts
 - b. Space shown between numeric value and "years". 5 pts
 7. Format code for **FullName** approximately matching key. 5 pts
 8. Printed field name and code for **Sales** approximately matching key. 5 pts
 9. Printed field name and code for **CommissionRate** approximately matching key. . . . 5 pts
 10. Printed field name and code for **CommissionEarned** approximately matching key. . 5 pts
 11. Printed field name and code for **StartYear** approximately matching key. 5 pts
 12. Printed field name and code for **CommissionRate** approximately matching key. . . . 5 pts
 13. Printed field name and code for **CommissionEarned** approximately matching key. . 5 pts

Note: Answer key is on the following page with note on grading typos for printed codes.

Deduct 2 points for each typo in the data. If there is a space before value for **CommissionRate** or **YearsEmployed**, it is a typo.

III. Answer Key.


FullName	Sales	Rate	CommissionEarned	StartYear	YearsEmployed
E. C. Street	\$228,396.92	9.3%	twenty-one thousand one hundred twenty-six and 72/100	MCMXCVIII	12 years
E. C. STREET	228396.920000000001	9.25	21126.715100000001	1998	12
Geoff L. Tavish	\$187,329.00	4.3%	seven thousand nine hundred sixty-one and 48/100	MMVI	6 years
Geoff L. tavish	187329	4.25	7961.4825000000001	2006	6
Curt Sanwhey	\$336,827.98	6.5%	twenty-one thousand eight hundred ninety-three and 82/100	MMIV	2004 years
Curt SANWHEY	336827.979999999998	6.5	21893.8187	2004	4

[GRADING REMARKS]

"FullName" * lower *Caps	[Spaces are optional before and after * for each instance.]
"Sales" \# \$#,00	[More # signs may be present after the \$ sign, but no more than ###,### may be present or they are typos. Any # signs after decimal are typos]
"CommissionRate" \# #.0%	[Only one # sign may be to the left of the decimal or these are typos. Any # signs after the decimal are typos]
"CommissionEarned" * DollarText	
"StartYear" * Roman	
"YearsEmployed" \# "#' years'	[If this has two # signs in the code, as \# "##' years' it is a typo.]
[NOTE: For all of the above, the quotation marks are optional around the name of the field, and a blank space before or after * or \# is optional. These are NOT graded as typos. Codes may be uppercase, lowercase, or mixed case.]	

IV. **Student Worksheet.** Students are to complete the following exercises after entering the following worksheet in Excel then print their Word document in landscape orientation.

- A. Exercise A. Create a blank database named **Exercises** with a table named **SalesForce-Numeric** with the following specifications.

	Field Name	Data Type	Description
	ID	AutoNumber	Primary Key
	FullName	Text	30 characters
	Sales	Number	Double, standard, 2 decimals
	CommissionRate	Number	Double, standard, 2 decimals
	CommisionEarned	Number	Double, standard, 2 decimals
	StartYear	Number	Long integer, fixed, 0 decimals
	YearsEmployed	Number	Integer, fixed, 0 decimals

1. Enter the following data in the **SalesForce-Numeric** table.

FullName	Sales	CommissionRate	CommisionEarned	StartYear	YearsEmployed
E. C. STREET	228,396.92	9.25		1998	12
Geoff I. tavish	187,329.00	4.25		2006	6
Curt SANWHEY	336,827.98	7.65		2004	4

2. Create an update query in design view to update the following fields.
- Update the **CommisionEarned** field by multiplying **Sales** times **CommissionRate** and dividing the result by 100.
 - Save the query as **UpdQry**, and run the query untill all fields are populated.
3. Open the table in the datasheet view.
4. Merge the table to Word creating a new document and linking to it.
5. The Word document should have the following specifications.
- Use landscape orientation and 1" margins all around.
 - Enter the following as column headings with tabs between them: **FullName**, **Sales**, **CommissionRate**, **CommisionEarned**, **StartYear**, and **YearsEmployed**.
 - Insert the appropriate merge field with tabs between them in the order of the headings.
 - Copy and paste the row with merged data into the row immediately below so that your document in the merge code view has the following appearance. (Note that you can align columns after you have merged data into the fields.

FullName	Sales	CommissionRate	CommisionEarned	StartYear	YearsEmployed
«FullName»	«Sales»	«CommissionRate»	«CommisionEarned»	«StartYear»	«YearsEmployed»
«FullName»	«Sales»	«CommissionRate»	«CommisionEarned»	«StartYear»	«YearsEmployed»

- Preview Results for your merged fields. You will see that the same record is used for both entries and it has peculiar number formatting.

- B. Exercise B. Use field codes to format the first row of merged fields.
1. Format the **FullName** field by using a primary and secondary field code.
 - a. From the Merge Code view of the document, click Alt F9 to see MERGEFIELD view.
 - b. Enter the field code, *lower *Caps for the first **FullName** merge field. This adds a primary field code to make everything lowercase and then adds a secondary field code to capitalize the first letter of each word. (ie. E. C. STREET should now display as E. C. Street.)
 2. Format the **Sales** value so that it appears with the \$ sign adjacent to the first displayed digit, and the field shows commas and 2 decimal positions, as \$228,396.92
 3. Format the **CommissionRate** so that it displays as one digit with 1 decimal position followed by a percent sign, as 9.3%
 4. Format **CommissionEarned** so that it displays in a spelled-out format, as twenty-one thousand one hundred twenty-six and 72/100.
 5. Format the **StartYear** as a Roman numeral.
 6. Format **YearsEmployed** so that it displays as a one- or two- digit value followed by a blank space and the word years, as 12 years.
- C. After everything is formatted appropriately, add the following.
1. Add two blank rows after the merged data.
 2. On the next line, select Next Record from the pull down menu on the Mailing ribbon in the Write & Insert Fields block labeled Rules.
 - a. Copy everything from the first two rows of merged data and paste it to the right of the Next Record code.
 - b. Leave two more blank lines after this record.
 - c. On the next line copy the second pair of lines, including the Next Record code
 - d. You will now have three pairs of merge-data lines.
 3. Preview Results of all of your data to be sure everything is formatting correctly.
 4. Use .25" left and right margins or use smaller type font to make everything fit on the lines.
 5. Vertically align the columns.
 6. Add two blank lines below the last line of merged data.
 7. View the data in the MERGEFIELD view, and select the field name and field code for each of the 6 fields and copy and paste below the data. (i.e. for the **StartYear**, the field should have the following appearance, {MERGEFIELD "Start Year" * Roman}, select only the "Start Year" * Roman to copy and paste below the data. These will be graded.
 8. Print your data.

V. Summary of Numeric field codes.

- A. Numeric fields may be formatted using these symbols in a format skeleton.
 1. 0 - specifies required positions to display, as 00.00 would show a 2-digit number with 2 decimals. If field does not have a numeric value, a zero is substituted.
 2. # - specifies required positions to display, as ##.## but if a numeric value is missing, the # is replaced with a blank.
 3. x - specifies where to stop showing values. If a field has a value 9876 and has the format code as \# \$x##, the field would display as \$876. If the value had only been 876, result would have shown all the numbers, as \$876.
 4. . - determines decimal point position.
 5. , - is a comma separator as is 10000 formatted as \# ###,### would be 10,000.
 6. - - adds a minus sign for a negative number and leaves a space for a positive number.
 7. + - adds a plus sign to a positive number, a minus sign for a negative number, and a blank for zero.
 8. % - adds a percent sign to a number.
- B. Numeric field may be formatted as text with the following codes.
 1. * CardText (displays as cardinal text, as twenty-one)
 2. * CardText * Cap (displays as capitalized cardinal text. There are several possibilities for this secondary field code, including Upper, Lower, FirstCaps, etc.)
 3. * DollarText (displays as cardinal text, twenty-one and inserts "and" at the decimal place and displays the first two decimals (rounded) as Arabic numerators over 100. (A secondary field code can be added to alter capitalization)
 4. * roman (displays as Roman numerals, as XXXI)
 5. * Arabic (displays as Arabic cardinal numerals, as 21; ArabicDash inserts hyphen before and after number)
 6. * Hex (displays as hexadecimal numbers, 1F)
 7. * OrdText (displays as ordinal text, thirty-first. A secondary field code can be added to specify different capitalization or date format)
 8. * Ordinal (displays as ordinal Arabic numerals, as 31st)
 9. Multiple codes may be used to create a specific condition when using text codes to format numeric data.
 - a. The first code is shown after the field name in the MERGEFIELD view.
 - b. The secondary code is added after the primary code.
 - c. This is often used to format a field as dollar text and then to capitalize the first letter of the field, as * DollarText * FirstCap
 10. If additional text is to be inserted in a numeric field, use a quotation mark following the \# code and show any added text within single quotation marks, such as \# "###'-##'-#####" to format a Social Security Number.