VISUAL BASIC PROGRAMMING
(44)

State 2012

CONTESTANT ID# _____ START TIME _____ END TIME _____

TOTAL POINTS _____ (630)

Failure to adhere to any of the following rules will result in disqualification:

1. ***Contestant must hand in this test booklet and all printouts. Failure to do so will result in disqualification.***
2. ***No equipment, supplies, or materials other than those specified for this event are allowed in the testing area. No previous BPA tests and/or sample tests or facsimile (handwritten, photocopied, or keyed) are allowed in the testing area.***
3. ***Electronic devices will be monitored according to ACT standards.***

Maximum test time is 90 minutes
(A five-minute warning will be given to ensure all printing is complete.)
NOTE: The administrator should allow for orientation, instructions,
warm-up, checking equipment, etc., before starting testing time.

Do **NOT** open test booklet until instructed to do so.

Scoring the form:

- A Heading is included …… _____(10)
- 3 Descriptive labels 3*@5pts ea* ……. _____(15)
- 2 Text fields 3*@5pts ea*…….…….…..…. _____(10)
- 1 Masked Text Field ……..… _____(05)
- 1 "Save Voter" button ..…..…… ……..……..……..…… _____(05)
- A groupbox encapsulates the textboxes, labels and command button …….. _____(10)
- Label "Programmer: contestant-id*"* in the right corner of form …..……..... _____(10)
- The form looks like the test handout of the form and includes patriotic colors _____(10) _____(75)

Scoring the code:

- Created a solution/project with the contest number…… _____(10)
- The Access DB was placed in the bin folder ……. _____(10)
- Data Designers and data binding were NOT used ……..……. _____(25)
- An OleDb connection was declared and properly initialized utilizing the connection string given ……..……. _____(20)
- An OleDb command was declared and properly initialized ……..…. _____(20)
- OleDb syntax was utilized to accomplish the insert operation (this could vary) _____(25)
- A Try/Catch Block was used when opening the data connection with correct syntax to handle a failure to open the connection……..…. _____(20)
- A Try/Catch Block was used when the record was inserted into the database with correct syntax to handle the condition of a duplicate record ……. _____(20)
- A function was written to edit check the name fields …… _____(30)
- The function was called for the first name field and called again for the last name field *2@15 ea*…….…….. _____(30)
- The function checked for alpha characters and blank fields *2@15 ea*……..... _____(30)
- The function accepted and utilized a parameter from the calling syntax …….. _____(15)
- The SSN field utilized the correct mask …… _____(15)
- All sub routines were documented ……..…… _____(15)
- General documentation at the top of form class was utilized ……. _____(10)
- Standard prefix naming conventions were used on controls and variables.. _____(10) _____(305)

Scoring the Execution (judge must run the program)

- The app returns the proper message if the first name field is blank or contains non-alpha characters *2@ 15 ea* …… _____(30)
- The app returns the proper message if the last name field is blank or contains non-alpha characters *2@ 15 ea* …… _____(30)
- Control is returned to first name field and field data highlighted if the edit check detects an error in the first or last name fields *2@15* ……..... _____(30)
- The SSN field only accepts numeric data …….…..……. _____(15)
- The app returns the proper message with OK and Cancel buttons if a duplicate SSN is entered …… _____(20)
- Control is returned to the first name field and the field data is highlighted if a duplicate SSN is entered and OK button is clicked …… _____(20)
- The form is closed if a duplicate SSN is entered and the Cancel button is clicked …….. _____(20)
- The voter registration record is written to the database if no data entry errors were detected ……..…..……..…... _____(50)

~~_____(15)~~

- Control is returned to the first name field upon a successful entry…….. _____(15)
- All fields are cleared upon a successful entry …… ……. _____(10)
- The tab order is correct ……..…….…… _____(10) _____(250)

TOTAL APPLICATION POINTS **_____(630)**

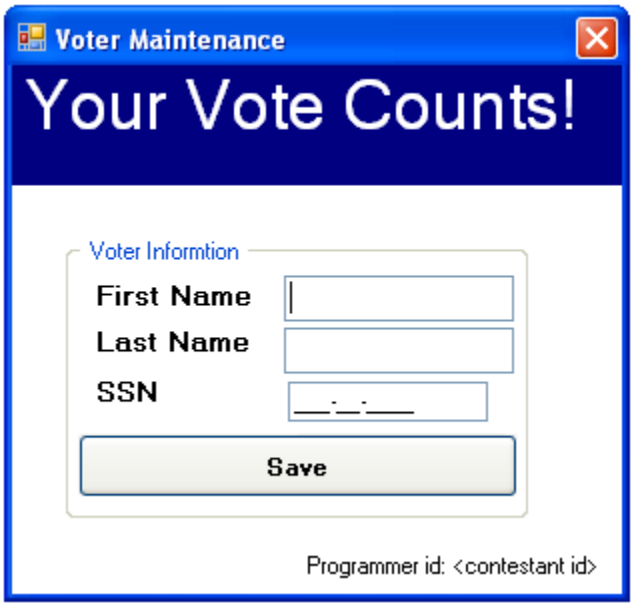
Note to Contestant: The scenario for this year's State event has the same theme as the Regional event, Your Vote Counts. This test will also employ the concept of *prototyping* which is an incomplete version of software to allow users to evaluate developers' proposals for the design of the eventual product and algorithms. The rubric grants points for each minor task completed. The contestant should strive to complete as many of the rubric items as possible. For grading purposes, this application must be written in VS2005 or later

Application:

Your county government would like to integrate a backend data store into the *Your Vote Counts* system. You have been asked to develop a routine to store voter registration information into a database. County officials have requested the following *(Refer to figure 1 to clarify)*

- A Patriotic color scheme must be used
- Include three fields and a button contained within a groupBox: First name, last name, social security number (SSN) and a submit button to save the data
- The head of the IT department for the county has asked that data designers NOT be used. The interface between the UI and the database must be coded. *See technical specs for more information*

Figure 1: Voter information entry screen



TECHNICAL SPECS

Setup the database within the context of the .NET solution

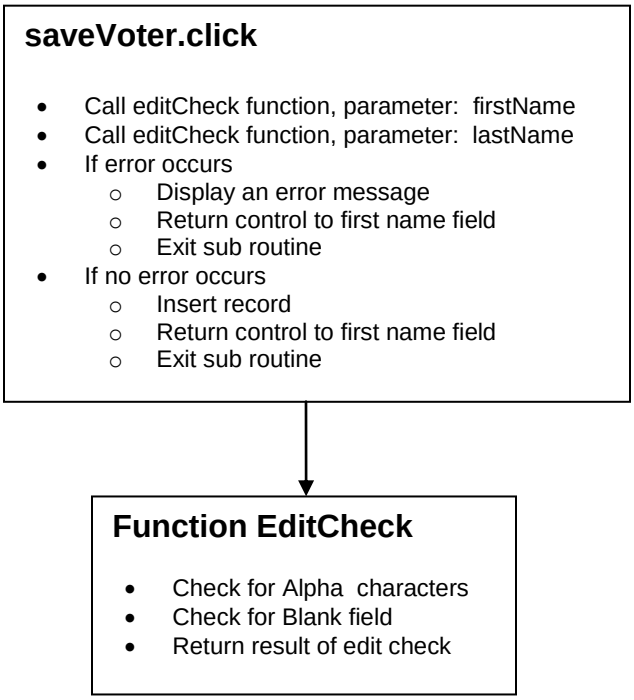
1. Create a solution for the project you have been asked to create given the following specs:
    a. Name the solution slnVbState*ContestantNumber* where contestant number is your contestant number
    b. Name the project: prjVbState*contestantNumber*
2. To keep it simple for this test, copy the Access database into the \bin\debug folder of your project
    a. Use the following connection string: "Provider=Microsoft.ACE.OLEDB.12.0; Data Source = dbVoterInfo.accdb"
    b. **Note: It is not necessary to make a connection to the Access ™ database in Server Explorer since no data designers are to be used for this project**

Write the data entry routine to store voter registration information

1. Create the routine as a prototype to test your data entry algorithms. Since it is a prototype, it need not be written in the context of a complete system
2. **No data designers are to be used**. The routine must be coded using the given connection string, OleDbConnections, OleDbCommands and other OleDb Objects to accomplish the task
3. Use error capture *(try/catch blocks)* routines at the most vulnerable locations.
    a. If an error is captured when the data is *inserted*,
        i. Display an informative message with selections for Ok or Cancel
        ii. If the user selects Cancel, close the form,
        iii. If the user selects Ok, direct the insertion point back to the first name field and select the text
    b. If the database fails to open
        i. Display an informative message
        ii. Close the form
4. Data entered into the first and last name fields must be edited for validity. The IT department manager has requested the following:
    a. Editing must occur when the user clicks the S*ave* button
    b. The first and last name must be populated and must contain only alpha characters
    c. The routine to test for both alpha characters and blank fields must be setup in the same function and called twice, once to check the first name, once to check for the last name *(see the simple logic chart below, figure 2).*
    d. The function must return a value indicating that the field did not pass the edit check. It does not have to indicate if the failure to pass was a blank field or a non-alpha character
    e. If a field does not pass an edit check
        i. Display the message: "Field contains non alpha characters or is blank" in the calling routine.
        ii. Return the insertion point to the first name field and select(high light) the text to allow the user to try again
    f. If a field does pass the edit check
        i. Clear all the fields
        ii. Return the insertion point to the first name field

5. The IT manager requested the following for the SSN *(social security number)* field:
    a. Use a masked text box with the appropriate mask for the SSN field
    b. Do not allow duplicates. If a duplicate entry occurs in the database
        i. Display an informative message with selections for Ok or Cancel
        ii. If the user selects Cancel, close the form,
        iii. If the user selects Ok, direct the insertion point back to the first name field and select the text
    c. Clear all fields and return control to the first name field upon a successful entry
6. Include a label in the lower right hand corner of the form indicating your contestant-id as illustrated in *figure 1*

Figure 2



In-House Protocol

- **Data designers and data binding are NOT permitted. It has been specified that all data connections and data operations must be written in code.**
- Try/Catch blocks must be used to capture potential data errors that could occur when transacting with the database. Inform the user of the error and close the program should an error occur
- Standard name prefixes must be utilized for form controls, coded variables, etc.

- All subroutines must be documented.
- Application documentation at the top of the form must be included
- Code duplication must be minimized
- The form must be in the same format as illustrated in the figures above and possess a professional look.

What to turn in:

- **Your solution/project with Access database copied to the flash drive provided. Your project must be executable from the flash drive. Your connection string must be constructed to find the database on the flash drive. If your solution/project cannot be opened and executed from the flash drive, your score will be zero. The project must be opened in order to grade the code**