KEY

VISUAL BASIC/ C# PROGRAMMING (330)

REGIONAL 2020

Production Portion:

Program 1: Data Storage Conversion _____ (450 points)

TOTAL POINTS _____ ***(450 points)***

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.**

No more than ten (10) minutes orientation
No more than 90 minutes testing time
No more than ten (10) minutes wrap-up

Data Storage Conversion

You are to create a Visual Basic or C# program that can convert all data types between Byte and Terbit. This program will have the user to enter the quantity through a textbox and select the original data type through the use of radio buttons. The user then must select the desired data type to convert to using radio buttons. The user will then press the convert button to display the answer. The user will then be able to change the quantity and/or the original data type and/or the desired data type. The conversion of data is based off of the number 1024. The following can be used for the conversion equations: 1 Byte = 1024^0 Bytes, 1 Kilobyte = 1024^1 Bytes, 1 Megabyte = 1024^2 Bytes, 1 Gigabyte = 1024^3 Bytes and 1 Terabyte = 1024^4 Bytes.

Steps:

1. Solution and Project
    a. Create a Visual Basic Windows Form Application named "VB_REG_ContestantNumber" or "CS_REG_ContestantNumber" (depending on which language you are using). The ContestantNumber is your BPA assigned contestant number. When naming your project, replace dashes (-) with the underscore (_). For example, if you BPA contestant number is 98-7654-3210, then your project name would be VB_REG_98_7654_3210.
2. User Interface
    a. The user interface to be constructed as shown in Figure 1. Your application must be visually identical to the prototype shown in Figure 1.
    b. The form's caption must be set to "Data Storage Conversion"
    c. The program is required to use two group boxes.
    d. The first group box
        i. The caption will be "Convert From"
        ii. One label to prompt the user to enter the quantity.
        iii. One textbox to take in the user input.
        iv. There will need to be five radio buttons for the user to select between Byte, Kilobit, Megabit, Gigabit and Terabit.
    e. The second group box:
        i. The caption will be "Convert To"
        ii. One label prompting the user to select the data size.
        iii. There will need to be five radio buttons for the user to select between Byte, Kilobit, Megabit, Gigabit and Terabit.
        iv. One button with the text of Convert
        v. One textbox to display the converted data amount
    f. Below the two group boxes, there will be a single button for exit.

3. Program Execution
    a. The program should user validate all input and create an error message if any of the user's input is missing without stopping the program's execution. See Figure 2.
    b. The program should validate that the value in the textbox located in the "Convert From" section contains a number.
    c. With all the given input, the program should calculate and display the conversion to the textbox within the "Convert To" group box.
        i. All output can be rounded to the 15th decimal place.
        ii. See Figure 3
    d. The program should exit when the user clicks the "Exit" button.
4. The Source Code
    a. To keep all the company's programs consistent, the following naming conventions must be used:
        i. The following prefixes should be used in naming the following:
            1. Forms – frm, Group_Boxes – grp, Labels – lbl, Textboxes – txt, Radio_Buttons – rb and Buttons – btn.
        ii. No single letter variables are to be used in the program.
    b. The program must be commented throughout, especially all functions and methods.

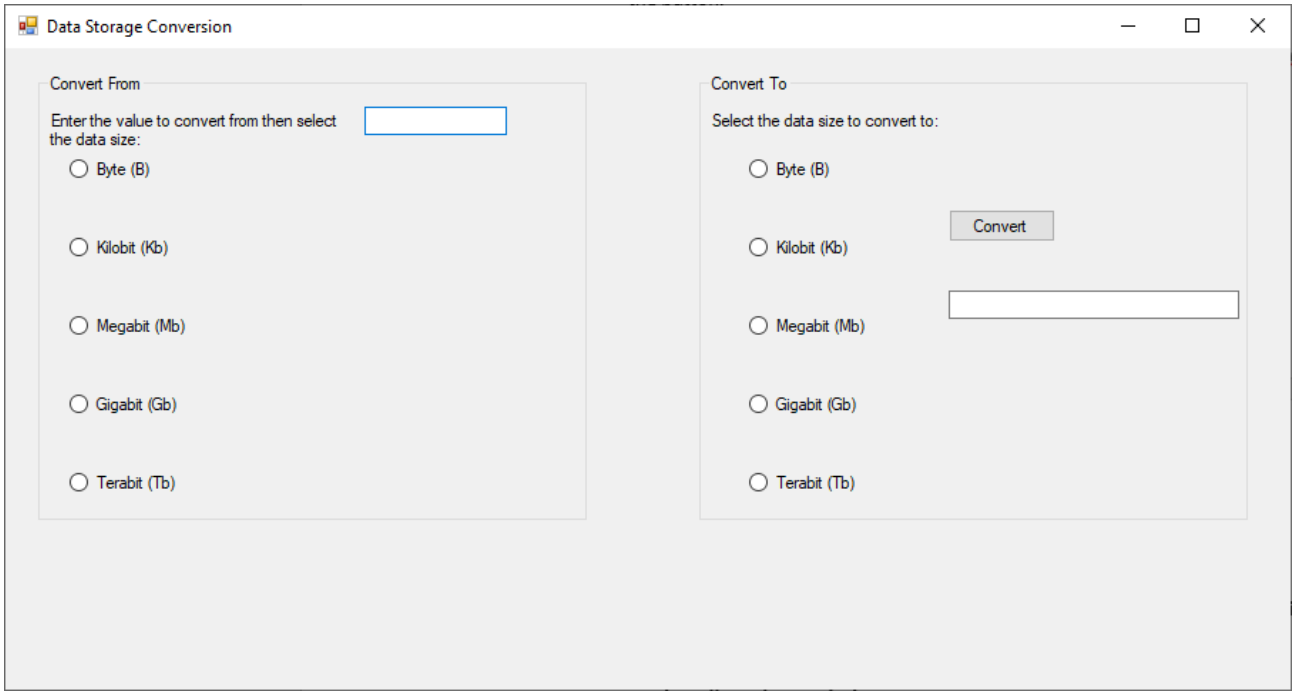

Figure 1

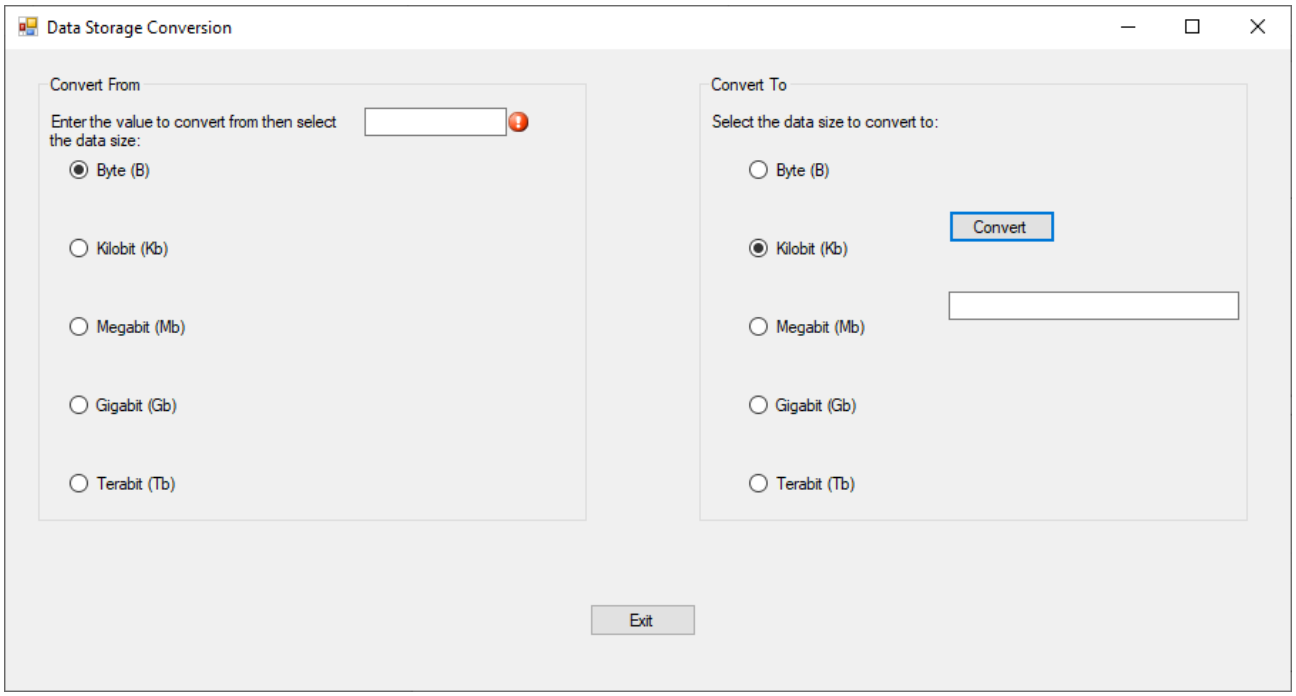

Figure 2

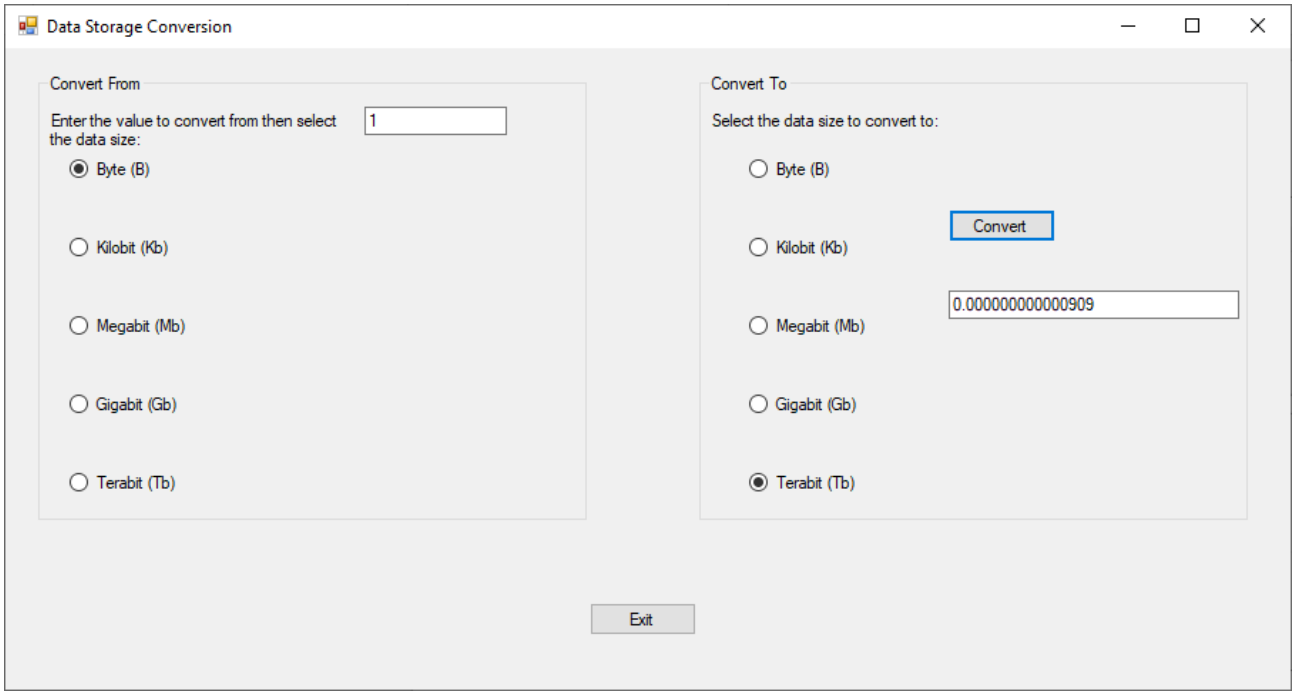

Figure 3

Your application will be graded on the following criteria:

Solution and Project

The project is present on the flash drive	____ 10 pts
The project is named according to the naming conventions	____ 10 pts

Program Execution

Code copied to USB drive and program runs from USB	____ 30 pts

If the program does not execute, then the remaining items in this section receive a score of zero.

Form has the correct caption (Data Storage Conversion)	____ 15 pts
The form is divided using two group boxes with the correct captions (10 per)	____ 20 pts
The radio buttons are evenly spaced vertically	____ 30 pts
Exit button is present and works correctly	____ 15 pts
Error message occur if any/all of the input is missing (20 points per input)	____ 60 pts
Error message occurs if a number is not entered in the convert from textbox	____ 20
Program calculates and displays the correct conversion	____ 60 pts

Source

Proper prefixes were used correctly	____ 40 pts
Code is commented throughout	____ 30 pts
Code exists to handle all missing inputs	_____30 pts
Code exists to validate the input entered into the textbox is a number	_____ 20 pts
Program calculates from smaller to higher conversions	____ 30 pts
Program calculates from higher to smaller	____ 30 pts

Total Points _____/ 450 points

VB SAMPLE CODE:

```
Imports System
Imports System.Collections.Generic
Imports System.ComponentModel
Imports System.Data
Imports System.Drawing
Imports System.Linq
Imports System.Text
Imports System.Threading.Tasks
Imports System.Windows.Forms

Namespace Data_Storage
   Public Partial Class frmDataConversion
      Inherits Form

      Public Sub New()
         InitializeComponent()
      End Sub

      Private Function ValidateFrom() As Boolean
         Dim bStatus As Boolean = True
         Dim parsedValue As Integer

         If txtFrom.Text = "" Then
            errorProvider1.SetError(txtFrom, "Please enter a value.")
            bStatus = False
         Else

            If Not Integer.TryParse(txtFrom.Text, parsedValue) Then
               errorProvider1.SetError(txtFrom, "Please enter an integer value")
               bStatus = False
            Else
               errorProvider1.SetError(txtFrom, "")
            End If
         End If

         If rbFromB.Checked = False AndAlso rbFromKb.Checked = False AndAlso
rbFromMb.Checked = False AndAlso rbFromGb.Checked = False AndAlso rbFromTb.Checked =
False Then
            errorProvider3.SetError(rbFromKb, "Please select a from data type.")
            bStatus = False
         Else
            errorProvider3.SetError(rbFromKb, "")
         End If

         If rbToB.Checked = False AndAlso rbToKb.Checked = False AndAlso rbToMb.Checked =
False AndAlso rbToGb.Checked = False AndAlso rbToTb.Checked = False Then
            errorProvider2.SetError(lblDirections, "Select a data type to convert to.")
```

```
            bStatus = False
        Else
            errorProvider2.SetError(lblDirections, "")
        End If

        Return bStatus
    End Function

    Private Sub btnToConvert_Click(ByVal sender As Object, ByVal e As EventArgs)
        Dim bValidFrom As Boolean = ValidateFrom()
        Dim fromVal As Integer
        Dim cvToBytes As Double = 0
        Dim cvToNew As Double = 0
        Dim cvToString As String

        If bValidFrom Then
            fromVal = Int32.Parse(txtFrom.Text)

            If rbFromB.Checked Then
                cvToBytes = fromVal
            ElseIf rbFromKb.Checked Then
                cvToBytes = 1024 * fromVal
            ElseIf rbFromMb.Checked Then
                cvToBytes = Math.Pow(1024, 2) * fromVal
            ElseIf rbFromGb.Checked Then
                cvToBytes = Math.Pow(1024, 3) * fromVal
            ElseIf rbFromTb.Checked Then
                cvToBytes = Math.Pow(1024, 4) * fromVal
            End If

            If rbToB.Checked Then
                cvToNew = cvToBytes
            ElseIf rbToKb.Checked Then
                cvToNew = cvToBytes / Math.Pow(1024, 1)
            ElseIf rbToMb.Checked Then
                cvToNew = cvToBytes / Math.Pow(1024, 2)
            ElseIf rbToGb.Checked Then
                cvToNew = cvToBytes / Math.Pow(1024, 3)
            ElseIf rbToTb.Checked Then
                cvToNew = cvToBytes / Math.Pow(1024, 4)
            End If

            txtConvert.Text = cvToNew.ToString("N15")
        Else
            MessageBox.Show("Input error, please correct and retry.")
        End If
    End Sub

    Private Sub btnExit_Click(ByVal sender As Object, ByVal e As EventArgs)
```

```
            Application.[Exit]()
        End Sub
    End Class
End Namespace
```

C# SAMPLE CODE:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Data_Storage
{
   public partial class frmDataConversion : Form
   {
      public frmDataConversion()
      {
         InitializeComponent();
      }



      private bool ValidateFrom()
      {
         bool bStatus = true;
         int parsedValue;
         if (txtFrom.Text == "")
         {
            errorProvider1.SetError(txtFrom, "Please enter a value.");
            bStatus = false;
         }
         else
         {
            if (!int.TryParse(txtFrom.Text, out parsedValue))
            {
               errorProvider1.SetError(txtFrom, "Please enter an integer value");
               bStatus = false;
            }
            else
            {
               errorProvider1.SetError(txtFrom, "");

            }
         }
         if (rbFromB.Checked == false && rbFromKb.Checked == false &&
            rbFromMb.Checked == false && rbFromGb.Checked == false &&
            rbFromTb.Checked == false )
         {
            errorProvider3.SetError(rbFromKb, "Please select a from data type.");
```

```
            bStatus = false;
        }
        else
        {
            errorProvider3.SetError(rbFromKb, "");

        }

        if (rbToB.Checked == false && rbToKb.Checked == false &&
            rbToMb.Checked == false && rbToGb.Checked == false &&
            rbToTb.Checked == false )
        {
            errorProvider2.SetError(lblDirections, "Select a data type to convert to.");
            bStatus = false;
        }



        else
        {
            errorProvider2.SetError(lblDirections, "");
        }
        return bStatus;
    }

    private void btnToConvert_Click(object sender, EventArgs e)
    {
        bool bValidFrom = ValidateFrom();
        int fromVal;
        double cvToBytes = 0;
        double cvToNew = 0;
        string cvToString;
        if (bValidFrom)
        {
            fromVal = Int32.Parse(txtFrom.Text);
            if (rbFromB.Checked)
            {
                cvToBytes = fromVal;
            }
            else if (rbFromKb.Checked)
            {
                cvToBytes = 1024 * fromVal;
            }
            else if (rbFromMb.Checked)
            {
                cvToBytes = Math.Pow(1024, 2) * fromVal;
            }
            else if (rbFromGb.Checked)
            {
                cvToBytes = Math.Pow(1024, 3) * fromVal;
```

```
            }
            else if (rbFromTb.Checked)
            {
               cvToBytes = Math.Pow(1024, 4) * fromVal;
            }

            if (rbToB.Checked)
            {
               cvToNew = cvToBytes;
            }
            else if (rbToKb.Checked)
            {
               cvToNew =  cvToBytes/ Math.Pow(1024, 1);
            }
            else if (rbToMb.Checked)
            {
               cvToNew = cvToBytes / Math.Pow(1024, 2)  ;
            }
            else if (rbToGb.Checked)
            {
               cvToNew = cvToBytes / Math.Pow(1024, 3) ;
            }
            else if (rbToTb.Checked)
            {
               cvToNew = cvToBytes / Math.Pow(1024, 4) ;
            }



            //cvToString = String.Format("{0:0,0.00}", cvToNew);
            //txtConvert.Text = cvToString;
            txtConvert.Text = cvToNew.ToString("N15");

         }
         else
         {
            MessageBox.Show("Input error, please correct and retry.");
         }
      }

      private void btnExit_Click(object sender, EventArgs e)
      {
         Application.Exit();
      }
   }
}
```