Visual Basic/C# Programming

(330)

REGIONAL – 2016

Program: Character Stats	______________	(400 points)
TOTAL POINTS	____________	***(400 points)***

Judge/Graders: Please double check and verify all scores and answer keys!

General Instructions

You work for XYZ Games, Inc., a company that develops multiple video games. You have been asked to develop a GUI application that will calculate character statistics.

Your deliverable for this project must include the following.

1. Create a multi-form GUI application.
2. The startup form, Main Form, must resemble Figure 1.
3. The second form, the Hero Form, must resemble Figure 2.
4. The application must adhere to proper naming conventions and include the required methods and functionality as mentioned in the specifications.

NOTES TO CONTESTANT

You will have 90 minutes to complete your work.

Your name and/or school name should *not* appear on any work you submit for grading.

Copy your entire solution/project to the flash drive provided. You must submit your entire solution/project so that the graders may open your project to review the source code. You must ensure that the files required to run your program are present and will execute on the flash drive provided.

Note that the flash drive letter may not be the same when the program is graded as it was when you created the program. It is recommended that you use relative paths rather than absolute paths to ensure that the program will run regardless of the flash drive letter. The graders will **not** compile or alter your source code to correct for this. Submissions that do not contain source code will **not be graded**.

The flash drive that you are provided *must* have three (3) pictures files for you. The files are:

- **archer.png**
- **warrior.png**
- **wizard.png**

Your application will be graded on the following criteria:

Solution and Project

The project is present on the flash drive _____ 5 points
The project is named according to the naming conventions _____ 5 points
Three (3) Character images are added to the project as Resources _____ 5 points

User Interface Design

Main Form

The Main Form resembles Figure 1 _____ 25 points
The name of the form is, “frmMain” _____ 5 points
The form’s caption displays “Contestant #01-2345-6789”, where #01-2345-6789 is the contestant’s properly formatted BPA ID _____ 5 points
The user interface contains a label explaining Heroes _____ 5 points
The user interface contains a button named, “btnHero” _____ 5 points
The user interface contains a button captioned, “Hero” _____ 5 points
The user interface contains a label explaining Villains _____ 5 points
The user interface contains a button named, “btnVillain” _____ 5 points
The user interface contains a button captioned, “Villain” _____ 5 points
The user interface contains a button named, “btnExit” _____ 5 points
The user interface contains a button captioned, “E&xit” _____ 5 points

Hero Form

The Hero Form resembles Figure 2 _____ 25 points
The name of the form is, “frmHero” _____ 5 points
The form’s caption displays “Hero Stats” _____ 5 points
Listbox for hero characters name “lstHero” _____ 5 points
Listbox for hero characters is not pre-populated _____ 5 points
PictureBox is implemented and image is not pre-populated _____ 5 points
Groupbox for skill level is populated with correct skill levels _____ 15 points
Groupbox for character stats is populated with the correct attributes _____ 15 points
A button named, “btnCalculate”, is defined _____ 5 points
A button captioned, “&Calculate”, is defined _____ 5 points
A button name, “btnBack”, is defined _____ 5 points
A button captioned, “&Back to Side Selector”, is defined _____ 5 points

Runtime Checks

Code copied to USB drive and program runs from the USB drive _____ 5 points

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

When the Hero button is clicked, the Hero Form is displayed _____ 25 points
When the Villain button is clicked, a coming soon message is displayed _____ 10 points
When the user attempts to close the application, a confirmation dialog is presented that confirms that the user wishes to exit the application _____ 15 points
User is prompted if they did not select a hero from Hero Listbox _____ 10 points
Application correctly calculates character's individual stats and total XP _____ 25 points

Source Code Review

Code is commented at the top, for each function, and as needed _____ 15 points
Code uses consistent variable naming conventions _____ 10 points

Main Form

An instance of the Hero Form is created and displayed on click of Hero button _____ 20 points

Hero Form

A method called, ResetValues(), is utilized to reset all controls _____ 20 points
A method called, CalcHeroLevels(), is utilized to calculate the appropriate character stats based on the Skill Level selected _____ 20 points
CalcHeroLevels() method displays the strength, speed, armor, magic, and total XP to the appropriate labels _____ 15 points
PictureBox image is assigned via code when Hero Listbox item is selected _____ 20 points

Total: _____ 400 points

Technical Specifications

1. Solution and Project
   - a. Create a Windows Form Application named 330_ContestantNumber, where ContestantNumber is your BPA assigned contestant number. When naming your project, replace dashes (-) with the underscore (_). For example, if your BPA contestant number is 01-2345-6789, then your project name would be 330_01_2345_6789.
2. User Interface
   - a. The user interface to be constructed is shown in Figure 1 and Figure 2. Your application must resemble the prototype (see Figure 1 and Figure 2).
   - b. The Main Form
     - i. The name of the main form should be “frmMain”.
     - ii. The Form’s caption must be set to “Contestant #01-2345-6789”, where 01-2345-6789 is your BPA assigned contestant number (including dashes).
     - iii. User interface elements and default (design-time) property settings
       1. Exit Button
          - a. The name of this button must be cmdExit
          - b. The text property for this button must be set to “E&xit”
       2. Hero Button
          - a. The name of this button must be “btnHero”
          - b. The text property for this button must be set to “&Hero”
       3. Villain Button
          - a. The name of this button must be “btnVillain”
          - b. The text property for this button must be set to “&Villain”
       4. Group Box
          - a. A group box called, “grpSelectSide” must be created
          - b. The group box must have the caption ‘Select a Side’
   - c. The Hero Form
     - i. The name of the main form should be frmHero.
     - ii. The Form’s caption must be set to “Hero Stats”.
     - iii. User interface elements and default (design-time) property settings
       1. Hero Listbox
          - a. The name of this listbox must be “lstHero”
          - b. The list box should be populated with three choices:
            - i. Archer
            - ii. Warrior
            - iii. Wizard
       2. PictureBox
          - a. The name of this picture box must be “picHero”

b. Sizemode should be set to StretchImage

c. The PictureBox image will set via code when a character is selected from the Hero listbox.

3. Skill Level Group Box

a. Group box is populated with radio buttons with the following names and text properties

i. Level 1, radLevel1

ii. Level 10, radLevel10

iii. Level 25, radLevel25

iv. Level 50, radLevel50

v. Level 75, radLevel75

vi. Level 99, radLevel99

b. The group box must have the caption “Select a skill level”

4. Character Stats Group Box

a. Group box is populated with labels for the following:

i. Strength

ii. Speed

iii. Armor

iv. Magic

v. Total XP

b. The group box must have the caption “Character Stats”

5. Back to Side Selector Button

a. The name of this button must be “btnBack”

b. The text property for this button must be set to “&Back to Side Selector”

6. Calculate Button

a. The name of this button must be “btnCalculate”

b. The text property for this button must be set to “&Calculate”

3. Tasks

a. Main Form - Exit Button Click

i. Closes the form. Upon closing, the application must confirm that the form is to be closed and allow the user to cancel the close by responding with “No” to the question, “Do you wish to exit this application?” If the user responds with a “Yes”, then the application is closed.

ii. The message box used to solicit the response must contain the message, “Do you wish to exit this application?” with a caption of “Exit?”

iii. The dialog box used to solicit the response must contain two buttons: Yes and No.

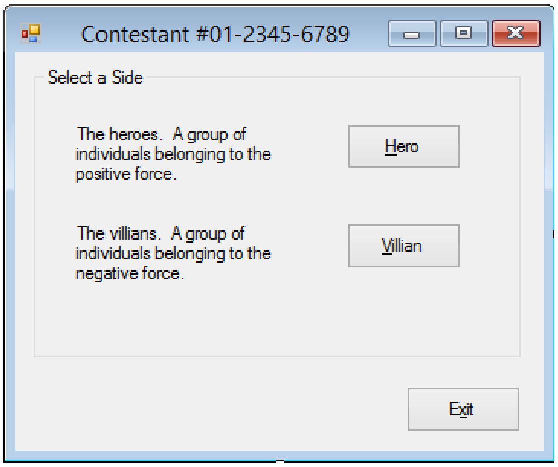
b. Main Form - Hero Button Click
   - i. Open the Hero Form (frmHero)

c. Main Form - Villain Button Click
   - i. Display a message to the user with the following message: “Coming soon: Select a villain” (see Figure 3).

d. Hero Form – Calculate Button Click
   - i. This procedure must validate whether or not a hero has been selected. If the user did not select a hero yet, a message should prompt the user.
   - ii. Upon click, a method named “CalcHeroLevels” is called.
     1. This method will calculate all of the character stats for the selected hero and skill level (see Figure 4 for a table of hero base stats)
     2. If a character is not selected from the Hero Listbox, the user should be prompted to select a character. Stats should not display if a character has not been selected.
     3. The total XP (Experience Points) is calculated by added the strength, speed, armor, and magic attributes.
     4. The stats for each attribute are based on a multiplier in the following list (for example, a Level 10 Warrior would have each attribute multiplied by 2):
        - Level 1 – 1x multiplier
        - Level 10 – 2x multiplier
        - Level 25 – 3x multiplier
        - Level 50 – 4x multiplier
        - Level 75 – 5x multiplier
        - Level 99 – 6x multiplier

e. Hero Form – Back to Side Selector Button
   - i. Closes the Hero form

f. Hero Listbox Selection
   - i. Upon selecting a hero from lstHero, a method called “ResetValues()” must be called.
     1. ResetValues() should reset the skill Level to Level 1 and all character stats should be reset to empty strings.
   - ii. Upon selecting a hero from lstHero, the correct image should populate in the picHero picture box
     - Reminder: the images are on the USB provided to you. If images were not provided, please contact the contest Administrator.
     - The images must be added to the project as a resource.



The heroes. A group of individuals belonging to the positive force.

The villians. A group of individuals belonging to the negative force.

Figure 1

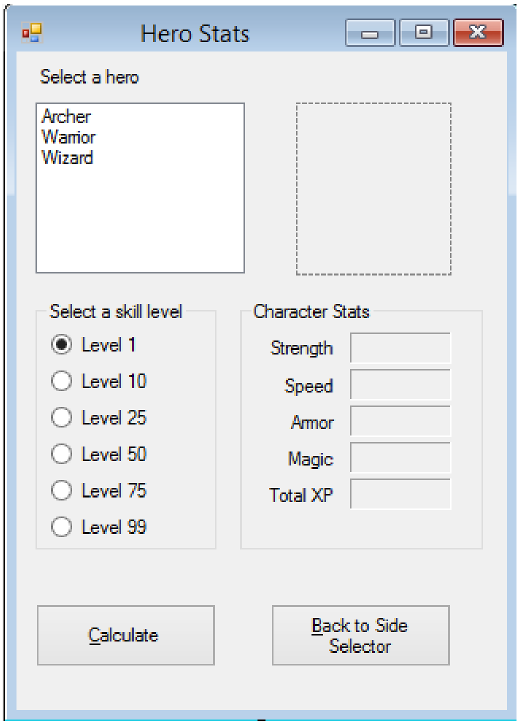
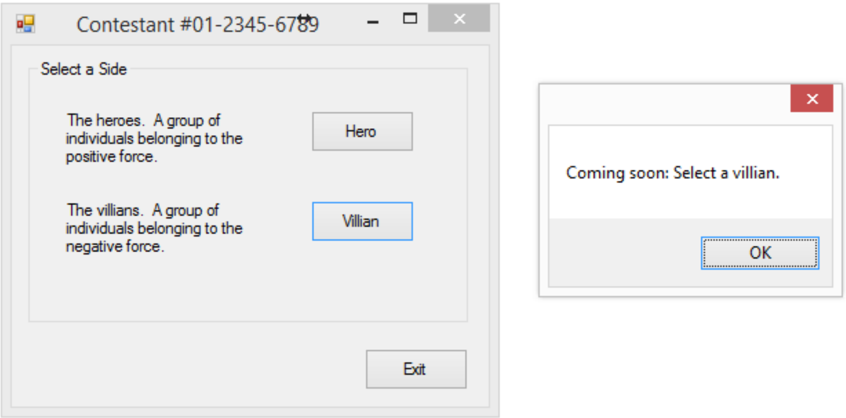


Figure 2



The heroes. A group of individuals belonging to the positive force.

The villians. A group of individuals belonging to the negative force.

Coming soon: Select a villian.

Figure 3

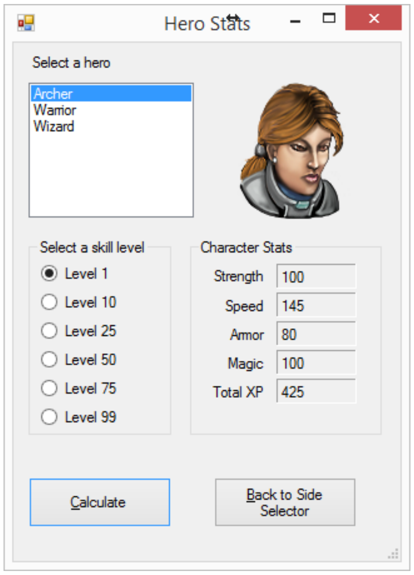
Attribute (Level 1 Stats)	Archer	Warrior	Wizard
Strength	100	150	200
Speed	145	115	400
Armor	80	120	100
Magic	100	90	600

Figure 4 – Hero base stats

Development Standards

- Standard name prefixes must be utilized for forms, controls, and variables.
- All subroutines, functions, and methods must be documented with comments explaining the purpose of the method, the input parameters (if any), and the output (if any).
- The user interface must be similar to Figure 1 and 2.

Example Scenarios



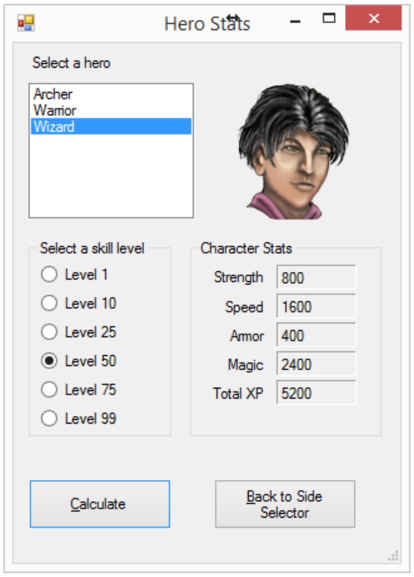
Hero Stats

Select a hero: Archer (selected), Warrior, Wizard

Select a skill level
● Level 1
○ Level 10
○ Level 25
○ Level 50
○ Level 75
○ Level 99

Character Stats
Strength: 100
Speed: 145
Armor: 80
Magic: 100
Total XP: 425

Archer on Level 1



Hero Stats

Select a hero: Archer, Warrior, Wizard (selected)

Select a skill level
○ Level 1
○ Level 10
○ Level 25
● Level 50
○ Level 75
○ Level 99

Character Stats
Strength: 800
Speed: 1600
Armor: 400
Magic: 2400
Total XP: 5200

Warrior on Level 50

****

Visual Basic Source Sample

****

frmMain.vb

```
Public Class frmMain

    Private Sub btnHero_Click(sender As Object, e As EventArgs) Handles btnHero.Click
        ' Create an instance of the HeroForm form.
        Dim frmHero As New frmHero

        ' Show the IndividualForm form.
        frmHero.ShowDialog()
    End Sub

    Private Sub btnVillian_Click(sender As Object, e As EventArgs) Handles btnVillian.Click
        'Message: under construction
        MessageBox.Show("Coming soon: Select a villian.")
    End Sub

    Private Sub btnExit_Click(sender As Object, e As EventArgs) Handles btnExit.Click
        ' Close the MainForm form.
        'Asks for confirmation, if result is yes then close
        If (MessageBox.Show("Do you wish to exit this application?", "Confirm Exit",
MessageBoxButtons.YesNo) = DialogResult.Yes) Then
            Me.Close()
        End If
    End Sub

End Class
```

frmHero.vb

```
Public Class frmHero

    Dim intStrength As Integer
    Dim intSpeed As Integer
    Dim intArmor As Integer
    Dim intMagic As Integer

    Private Sub btnCalculate_Click(sender As Object, e As EventArgs) Handles
btnCalculate.Click
        If lstHero.SelectedIndex <> -1 Then
            ResetValues()
            CalcHeroLevels()
        Else
            MessageBox.Show("You must first select a hero")
        End If


       End Sub
       Private Sub lstHero_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
lstHero.SelectedIndexChanged
        ResetValues()
    End Sub

    Sub CalcHeroLevels()
        If lstHero.SelectedIndex <> -1 Then
            ' Display the totals.
            If radLevel1.Checked = True Then
            ElseIf radLevel10.Checked = True Then
                intStrength = intStrength * 2
                intSpeed = intSpeed * 2
                intArmor = intArmor * 2
                intMagic = intMagic * 2
            ElseIf radLevel25.Checked = True Then
                intStrength = intStrength * 3
                intSpeed = intSpeed * 3
                intArmor = intArmor * 3
                intMagic = intMagic * 3
            ElseIf radLevel50.Checked = True Then
                intStrength = intStrength * 4
                intSpeed = intSpeed * 4
                intArmor = intArmor * 4
                intMagic = intMagic * 4
            ElseIf radLevel75.Checked = True Then
                intStrength = intStrength * 5
                intSpeed = intSpeed * 5
                intArmor = intArmor * 5
                intMagic = intMagic * 5
            ElseIf radLevel99.Checked = True Then
                intStrength = intStrength * 6
                intSpeed = intSpeed * 6
                intArmor = intArmor * 6
                intMagic = intMagic * 6
            Else
                MessageBox.Show("You must select a level")
            End If

            lblStrength.Text = intStrength.ToString()
            lblSpeed.Text = intSpeed.ToString()
            lblArmor.Text = intArmor.ToString()
            lblMagic.Text = intMagic.ToString()
            lblTotal.Text = (intStrength + intSpeed + intArmor + intMagic).ToString()
        Else
            MessageBox.Show("You must select a hero before calculating total XP.")
        End If

    End Sub

    Sub ResetValues()
        If lstHero.SelectedIndex = 0 Then
            picHero.Image = My.Resources.archer
            intStrength = 100

            intSpeed = 145
            intArmor = 80
            intMagic = 100
        ElseIf lstHero.SelectedIndex = 1 Then
            picHero.Image = My.Resources.warrior
            intStrength = 150
            intSpeed = 115
            intArmor = 120
            intMagic = 90
        ElseIf lstHero.SelectedIndex = 2 Then
            picHero.Image = My.Resources.wizard
            intStrength = 200
            intSpeed = 400
            intArmor = 100
            intMagic = 600
        Else

        End If
    End Sub

    Private Sub btnClose_Click(sender As Object, e As EventArgs) Handles btnBack.Click
        Me.Close()
    End Sub

End Class
```

****

C# Source Sample

****

frmMain.cs

```
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 CharacterStats
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void btnHero_Click(object sender, EventArgs e)
        {
```

```
       //Create an instance of the HeroForm form.
            HeroForm frmHero = new HeroForm();

            //Show the IndividualForm form.
            frmHero.ShowDialog();
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void btnVillian_Click(object sender, EventArgs e)
        {
            //Message: under construction
            MessageBox.Show("Coming soon: Select a villian");
        }
    }
}
```

frmHero.cs

```
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 CharacterStats
{
    public partial class HeroForm : Form
    {

        private int intStrength;
        private int intSpeed;
        private int intArmor;
        private int intMagic;
        private int intTotal;

        public HeroForm()
        {
            InitializeComponent();
        }

        private void btnCalculate_Click(object sender, EventArgs e)
        {
            if (lstHero.SelectedIndex != -1)
            {
                ResetValues();
                CalcHeroLevels();
            }
            else
            {
                MessageBox.Show("You must first select a hero");
            }

        }

        private void lstHero_SelectedIndexChanged(object sender, EventArgs e)
        {
            ResetValues();
        }

        void CalcHeroLevels()
        {
            if (lstHero.SelectedIndex != -1)
            {
                //Display the totals.
                if (radLevel1.Checked == true)
                {

                }
                else if (radLevel10.Checked == true)
                {
                    intStrength = intStrength * 2;
                    intSpeed = intSpeed * 2;
                    intArmor = intArmor * 2;
                    intMagic = intMagic * 2;
                }

                else if (radLevel25.Checked == true)
                {
                    intStrength = intStrength * 3;
                    intSpeed = intSpeed * 3;
                    intArmor = intArmor * 3;
                    intMagic = intMagic * 3;
                }

                else if (radLevel50.Checked == true)
                {
                    intStrength = intStrength * 4;
                    intSpeed = intSpeed * 4;
                    intArmor = intArmor * 4;
                    intMagic = intMagic * 4;
                }

                else if (radLevel75.Checked == true)
                {
                    intStrength = intStrength * 5;
                    intSpeed = intSpeed * 5;
                    intArmor = intArmor * 5;
                    intMagic = intMagic * 5;
                }

                else if (radLevel99.Checked == true)
                {
                    intStrength = intStrength * 6;
                    intSpeed = intSpeed * 6;
                    intArmor = intArmor * 6;
                    intMagic = intMagic * 6;
                }

                else
                {
                    MessageBox.Show("You must select a level");
                }

                lblStrength.Text = intStrength.ToString();
                lblSpeed.Text = intSpeed.ToString();
                lblArmor.Text = intArmor.ToString();
                lblMagic.Text = intMagic.ToString();
                lblTotal.Text = (intStrength + intSpeed + intArmor + intMagic).ToString();
            }

            else
                MessageBox.Show("You must select a hero before calculating total XP.");


        }

        void ResetValues()
        {
            if (lstHero.SelectedIndex == 0)
            {
                picHero.Image = CharacterStats.Properties.Resources.archer;
                intStrength = 100;
                intSpeed = 145;
                intArmor = 80;
                intMagic = 100;
            }

            else if (lstHero.SelectedIndex == 1)
            {
                picHero.Image = CharacterStats.Properties.Resources.warrior;
                intStrength = 150;
                intSpeed = 115;
                intArmor = 120;
                intMagic = 90;
            }

            else if (lstHero.SelectedIndex == 2)
            {
                picHero.Image = CharacterStats.Properties.Resources.wizard;
                intStrength = 200;
                intSpeed = 400;
                intArmor = 100;
                intMagic = 600;
            }

            else
            {
            }

        }

        private void btnClose_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```