**Class frmMain:**

Declare arrays names, letterGrades, numericalGrades

**Sub frmMain\_Load:**

Call LoadDataIntoArrays()

Call DisplayInitialData()

**Sub GradesListBox\_SelectedIndexChanged:**

Call ClearListBoxesAndLabels()

**Sub btnDisplay\_Click:**

Call DisplaySelectedGrades()

**Sub btnClassStatistics\_Click:**

Call DisplayClassStats()

**Sub btnExit\_Click:**

Call ExitApplication()

**Sub frmMain\_FormClosing:**

Call ConfirmExit()

**Sub LoadDataIntoArrays:**

' Load data into arrays as needed

**Sub DisplayInitialData:**

' Populate grades list box with unique letter grades

uniqueGrades = DistinctAndOrderBy(letterGrades)

GradesListBox.Items.AddRange(uniqueGrades)

' Set default selection to index 0

GradesListBox.SelectedIndex = 0

**Sub ClearListBoxesAndLabels:**

' Clear list boxes and labels as needed

StudentsListBox.Items.Clear()

lblStudentNum.Text = ""

lblAverage.Text = ""

lblAverageLetter.Text = ""

**Sub DisplaySelectedGrades:**

' Clear previous data

Call ClearListBoxesAndLabels()

' Get the selected letter grade

selectedGrade = GradesListBox.SelectedItem

' Display students, count, and average for the selected grade

For i = 0 To letterGrades.Length - 1:

If letterGrades[i] = selectedGrade:

StudentsListBox.Items.Add(names[i])

' Sort the StudentsListBox

StudentsListBox.Sorted = True

' Display the number of students

lblStudentNum.Text = StudentsListBox.Items.Count

' Display the average grade for the selected letter grade

averageGrade = CalculateAverageGrade(selectedGrade)

lblAverage.Text = Format(averageGrade, "F2")

' Display the average letter grade for the selected grade

lblAverageLetter.Text = ConvertToLetterGrade(averageGrade)

**Sub ConvertToLetterGrade(grade):**

If grade >= 90:

Return "Average A:"

ElseIf grade >= 80:

Return "Average B:"

ElseIf grade >= 70:

Return "Average C:"

ElseIf grade >= 60:

Return "Average D:"

Else:

Return "Average F:"

**Sub CalculateAverageGrade(selectedGrade):**

totalGrade = 0

count = 0

For i = 0 To letterGrades.Length - 1:

If letterGrades[i] = selectedGrade:

totalGrade += numericalGrades[i]

count += 1

If count > 0:

Return totalGrade / count

Else:

Return 0

**Sub DisplayClassStats:**

' Calculate and display average, highest, and lowest for the entire class

averageClassGrade = numericalGrades.Average()

highestClassGrade = numericalGrades.Max()

lowestClassGrade = numericalGrades.Min()

' Display class statistics in labels

lblGradeAverage.Text = Format(averageClassGrade, "F2")

lblHighestGrade.Text = highestClassGrade

lblLowestGrade.Text = lowestClassGrade

**Sub ExitApplication**:

' Display a confirmation message box

result = MessageBox.Show("Are you sure you want to exit?", "Professor Juarez", MessageBoxButtons.YesNo, MessageBoxIcon.Warning)

' Check the user's response

If result = DialogResult.Yes:

' If the user clicks "Yes," close the form

Me.Close()

**Sub ConfirmExit:**

' Check if the user wants to exit

If MessageBox.Show("Do you want to exit?", "Professor Juarez", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) = DialogResult.No:

' If the user clicks "No," cancel the form closing event

e.Cancel = True