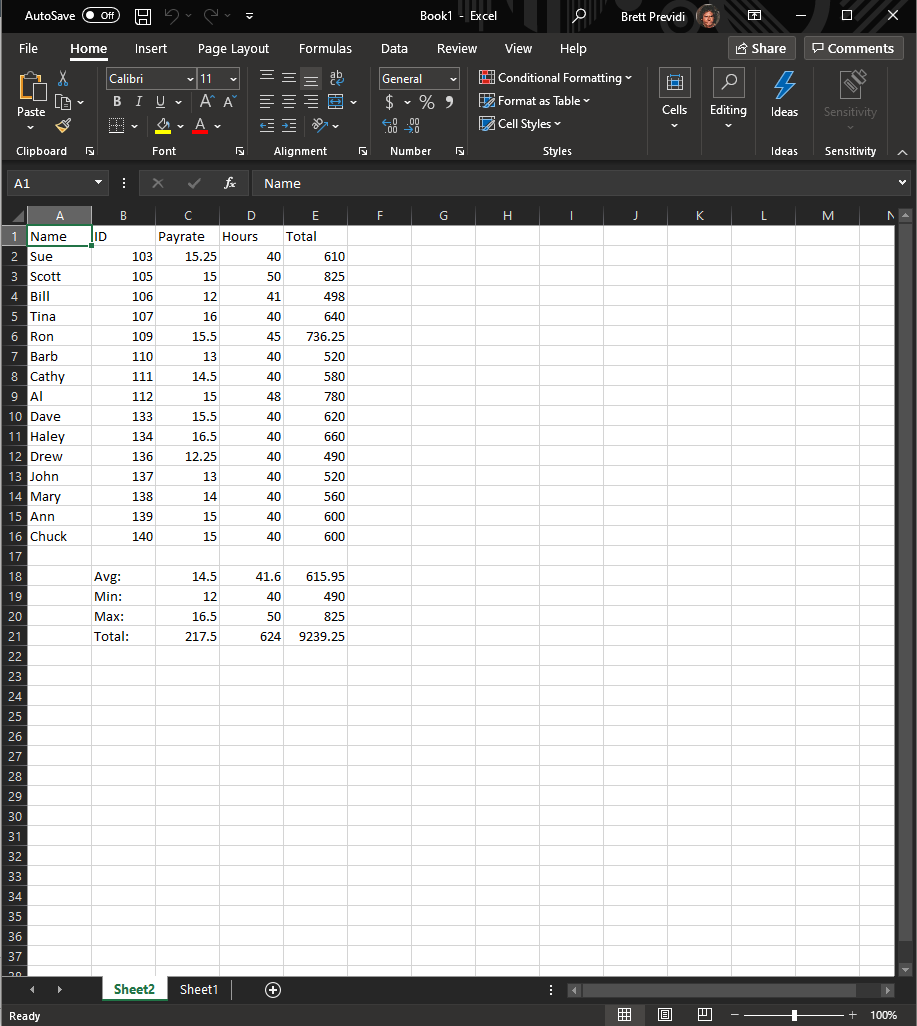
Program Cover Sheet

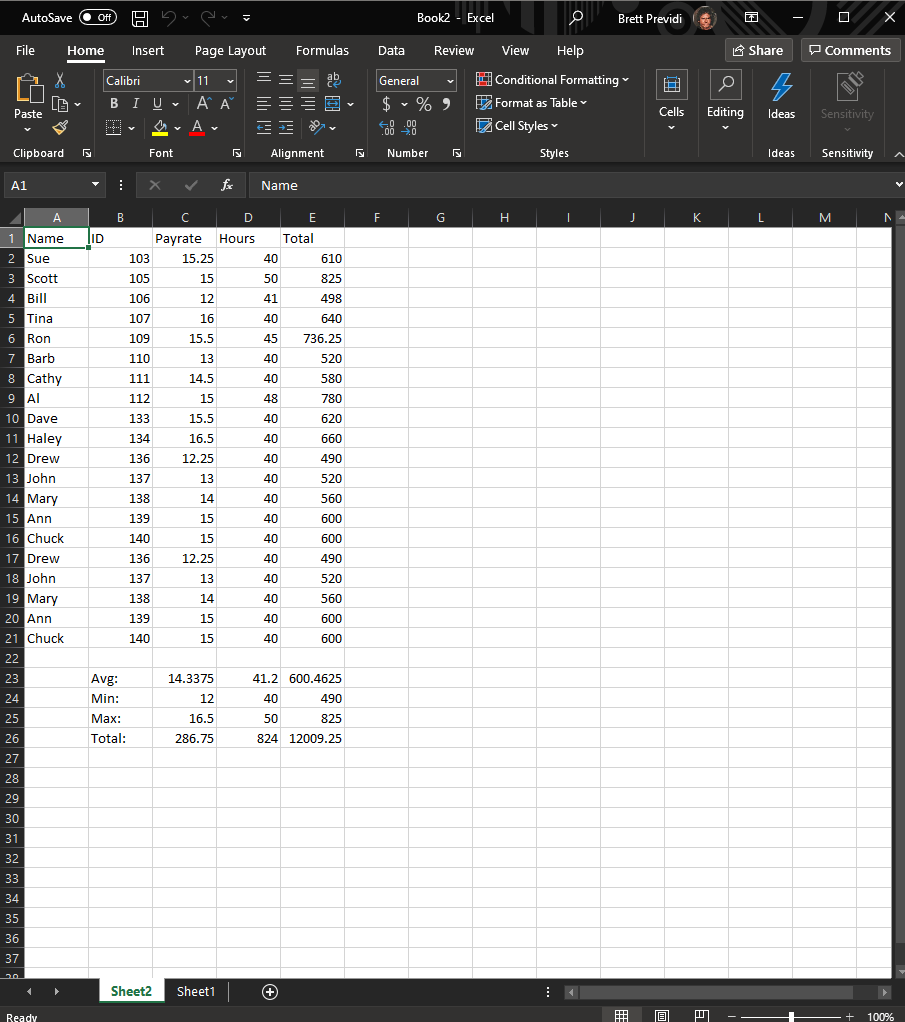
|  |
| --- |
| Name: Brett Previdi |
| Assignment: Assignment 09 |
| List any parts of the assignment that do not work/were not completed: |

|  |
| --- |
| Instructor’s Comments: |
| Grade: |

Program Submission Requirements: (1) all files, zipped and uploaded to Canvas and (2) a completed cover sheet, program execution screenshots and source code printed, **stapled** and turned in during class. Failure to follow the submission requirements will result in points lost on that particular assignment.



Some values repeated to show dynamic changes:



Public Class clsEmpPayroll

'---------------------------

' File Name: clsEmpPayroll

' Project: Assignment 09

' Written By: Brett Previdi

' Written On: 2021 - 04 - 05

' Version 1.0.0.0

'---------------------------

' File Purpose:

' This file contains the class that holds employee information.

' A constructor assigns the 4 variables and their values from the main application.

' A method for calculating total worked hours for an employee is also provided.

'---------------------------

' Program Purpose:

' This program creates a list of employees built using a class holding an employee's

' name, ID, hourly rate, and hours worked per day of the week. The information is passed

' to a new or existing Excel sheet where their name, ID, and payrate is displayed in order.

' Each employee's total hours worked are passed from clas method, and thier weekly pay is

' calculated in Excel. The average, minimum, maximum, and totals for each of the pay related

' variables are calculated and displayed below the employee list in Excel.

'---------------------------

' Global Variable Dictionary

Property strEmployeeName As String 'Holds the employee's name.

Property intEmployeeID As Integer 'Holds the employee's ID number.

Property sngHourlyRate As Single 'Holds the employee's hourly rate.

Property sngHoursWorked As Single() 'Holds an array of hours worked in one week.

'---------------------------

Public Sub New(ByVal newName As String, ByVal newID As Integer, ByVal newRate As Single, ByVal newWorked As Single())

'---------------------------

' Subroutine Name: New(ByVal newName As String, ByVal newID As Integer, ByVal newRate As Single, ByVal newWorked As Single())

' Written By: Brett Previdi

' Written On: 2021 - 04 - 05

'---------------------------

' Subroutine Purpose:

' This serves as the contstructor for objects of the clsEmpPayroll class when

' variables are passed to it. The values passed from the caller are assigned to

' their respective object variables.

'---------------------------

' Local Variable Library

' newName as String 'The passed string containing an employee name from where the object is created to be assigned to the new object.

' newID as Integer 'The passed integer containing an eompoyee ID number from where the object is created to be assigned to the new object.

' newRate as Single 'The passed single containing an employee's hourly rate from where the object is created to be assigned to the new object.

' newWorked as Single() 'The passed array of singles containing all the hours worked by an employee for a week, or 7 days, to be assigned to the new employee object.

'---------------------------

strEmployeeName = newName 'Set the name.

intEmployeeID = newID 'Set the ID.

sngHourlyRate = newRate 'Set the hourly rate.

sngHoursWorked = newWorked 'Set the hours worked.

End Sub

Public Function totalHours()

'---------------------------

' Subroutine Name: totalHours

' Written By: Brett Previdi

' Written On: 2021 - 04 - 05

'---------------------------

' Subroutine Purpose:

' When called, this function combines the hours an employee worked in a week

' and returns the sum. This is done by adding all of the array variables

' to a single variable to return.

'---------------------------

' Local Variable Dictionary

' sngHours as Single 'Holds the sum of the hours worked in a week

'---------------------------

Dim sngHours As Single = 0

For Each day In sngHoursWorked

sngHours += day

Next

Return sngHours

End Function

End Class

Imports Microsoft.Office.Interop 'Bring over the references and objects of Microsoft Excel.

Module mdlEmployeePayroll

'---------------------------

' File Name: mdlEmpPayroll

' Project: Assignment 09

' Written By: Brett Previdi

' Written On: 2021 - 04 - 05

' Version 1.0.0.0

'---------------------------

' File Purpose:

' This is the main file for the program that performs all object creation and communication

' with Microsoft Excel. A list of employee objects are first created. An Excel sheet is either

' created or assigned to the program and the necessary fields are filled in Excel.

'---------------------------

' Program Purpose:

' This program creates a list of employees built using a class holding an employee's

' name, ID, hourly rate, and hours worked per day of the week. The information is passed

' to a new or existing Excel sheet where their name, ID, and payrate is displayed in order.

' Each employee's total hours worked are passed from clas method, and thier weekly pay is

' calculated in Excel. The average, minimum, maximum, and totals for each of the pay related

' variables are calculated and displayed below the employee list in Excel.

'---------------------------

' Global Variable Dictionary

Dim myEmps As List(Of clsEmpPayroll) = New List(Of clsEmpPayroll) 'The list of employees made of constructed employee class objects.

Dim CheckExcel As Object 'Used to check if Excel is already running by tagging an open window to this object.

Dim ExcelDoc As Excel.Application 'New Excel document creation.

Dim intCurrentRow As Integer = 2 'Holds the current Row being manipulated in Excel. Starts at 2 to account for column labels.

'---------------------------

Sub Main()

'---------------------------

' Subroutine Name: Main

' Written By: Brett Previdi

' Written On: 2021 - 04 - 05

'---------------------------

' Subroutine Purpose:

' When the application laods, objects for the list of myEmps will be created.

' The constructor for clsEmpPayroll objects that represent employees will be

' called multiple times passing variables for each entry. The Excel document

' is then created or an existing one is assigned to host the information.

' The employee list is then written to Excel and the necessary calculations

' are dynamically written.

'---------------------------

' No Local Variables

'---------------------------

'Below are all default instances of employees to be added to the payroll system.

myEmps.Add(New clsEmpPayroll("Sue", 103, 15.25, {8, 8, 8, 8, 8, 0, 0}))

myEmps.Add(New clsEmpPayroll("Scott", 105, 15.0, {10, 10, 0, 10, 10, 10, 0}))

myEmps.Add(New clsEmpPayroll("Bill", 106, 12.0, {8, 8, 8, 8, 9, 0, 0}))

myEmps.Add(New clsEmpPayroll("Tina", 107, 16.0, {8, 8, 8, 8, 8, 0, 0}))

myEmps.Add(New clsEmpPayroll("Ron", 109, 15.5, {0, 0, 9, 9, 9, 9, 9}))

myEmps.Add(New clsEmpPayroll("Barb", 110, 13.0, {0, 10, 0, 10, 10, 10, 0}))

myEmps.Add(New clsEmpPayroll("Cathy", 111, 14.5, {8, 8, 8, 8, 8, 0, 0}))

myEmps.Add(New clsEmpPayroll("Al", 112, 15.0, {10, 10, 10, 10, 8, 0, 0}))

myEmps.Add(New clsEmpPayroll("Dave", 133, 15.5, {0, 0, 8, 8, 8, 8, 8}))

myEmps.Add(New clsEmpPayroll("Haley", 134, 16.5, {8, 8, 8, 8, 8, 0, 0}))

myEmps.Add(New clsEmpPayroll("Drew", 136, 12.25, {10, 10, 0, 0, 10, 10, 0}))

myEmps.Add(New clsEmpPayroll("John", 137, 13.0, {8, 8, 8, 8, 8, 0, 0}))

myEmps.Add(New clsEmpPayroll("Mary", 138, 14.0, {8, 8, 8, 8, 8, 0, 0}))

myEmps.Add(New clsEmpPayroll("Ann", 139, 15.0, {0, 0, 0, 10, 10, 10, 10}))

myEmps.Add(New clsEmpPayroll("Chuck", 140, 15.0, {0, 8, 8, 8, 8, 8, 0}))

Try 'See if Excel is already running.

CheckExcel = GetObject(, "Excel.Application")

Catch ex As Exception 'If Excel is running, throw an error and tell the user to close it.

'Excel isn't running, so this is an error that does nothing to create a new document later.

End Try

If CheckExcel Is Nothing Then 'If Excel is not open, create a new instance of Excel.

ExcelDoc = New Excel.Application

ExcelDoc.Visible = True

Else 'If Excel is open, assign the open window to the Excel doc object.

ExcelDoc = CheckExcel

ExcelDoc.Visible = True

End If

ExcelDoc.Workbooks.Add() 'Create a new workbook in the Excel document.

ExcelDoc.Sheets.Add() 'Create a new sheet in the Excel workbook.

ExcelDoc.Cells(1, 1) = "Name" 'Label the first column in the sheet for employee names.

ExcelDoc.Cells(1, 2) = "ID" 'Label the second column for employee IDs.

ExcelDoc.Cells(1, 3) = "Payrate" 'Label the third column for employee payrates.

ExcelDoc.Cells(1, 4) = "Hours" 'Label the fourth column for total employee hours worked.

ExcelDoc.Cells(1, 5) = "Total" 'Label the fifth columnn for the amount of pay employees are to receive.

For Each Emp In myEmps 'Display each employee in the roster.

ExcelDoc.Cells(intCurrentRow, 1) = Emp.strEmployeeName 'Display employee name.

ExcelDoc.Cells(intCurrentRow, 2) = Emp.intEmployeeID 'Display employee ID.

ExcelDoc.Cells(intCurrentRow, 3) = Emp.sngHourlyRate 'Display employee payrate.

ExcelDoc.Cells(intCurrentRow, 4) = Emp.totalHours() 'Call the function to calculate total worked hours in a week then display the result.

ExcelDoc.Cells(intCurrentRow, 5) = "=IF(D" & intCurrentRow & ">40,(C" & intCurrentRow &

"\*40)+(C" & intCurrentRow & "\*1.5)\*(D" & intCurrentRow & "-40),C" & intCurrentRow & "\*D" & intCurrentRow & ")" 'Pass the formula for Excel to calculate earned pay for each employee.

intCurrentRow += 1 'Move to the next row in Excel.

Next

intCurrentRow += 1 'Move down a row to create space below employee list.

ExcelDoc.Cells(intCurrentRow, 2) = "Avg:" 'This block creates a label and formulas to display the averages of...

ExcelDoc.Cells(intCurrentRow, 3) = "=AVERAGE(C2:C" & intCurrentRow - 2 & ")" 'payrate...

ExcelDoc.Cells(intCurrentRow, 4) = "=AVERAGE(D2:D" & intCurrentRow - 2 & ")" 'hours worked...

ExcelDoc.Cells(intCurrentRow, 5) = "=AVERAGE(E2:E" & intCurrentRow - 2 & ")" 'and earned pay between all employees.

intCurrentRow += 1 'Move to the next row in Excel.

ExcelDoc.Cells(intCurrentRow, 2) = "Min:" 'This block creates a label and formulas to display the minimum of...

ExcelDoc.Cells(intCurrentRow, 3) = "=MIN(C2:C" & intCurrentRow - 3 & ")" 'payrate...

ExcelDoc.Cells(intCurrentRow, 4) = "=MIN(D2:D" & intCurrentRow - 3 & ")" 'hours worked...

ExcelDoc.Cells(intCurrentRow, 5) = "=MIN(E2:E" & intCurrentRow - 3 & ")" 'and earned pay between all employees.

intCurrentRow += 1 'Move to the next row in Excel.

ExcelDoc.Cells(intCurrentRow, 2) = "Max:" 'This block creates a label and formulas to display the maximum of...

ExcelDoc.Cells(intCurrentRow, 3) = "=Max(C2:C" & intCurrentRow - 4 & ")" 'payrate...

ExcelDoc.Cells(intCurrentRow, 4) = "=Max(D2:D" & intCurrentRow - 4 & ")" 'hours worked...

ExcelDoc.Cells(intCurrentRow, 5) = "=MAx(E2:E" & intCurrentRow - 4 & ")" 'and earned pay between all employees.

intCurrentRow += 1 'Move to the next row in Excel.

ExcelDoc.Cells(intCurrentRow, 2) = "Total:" 'This block creates a label and formulas to display the sum of...

ExcelDoc.Cells(intCurrentRow, 3) = "=SUM(C2:C" & intCurrentRow - 5 & ")" 'payrate...

ExcelDoc.Cells(intCurrentRow, 4) = "=SUM(D2:D" & intCurrentRow - 5 & ")" 'hours worked...

ExcelDoc.Cells(intCurrentRow, 5) = "=SUM(E2:E" & intCurrentRow - 5 & ")" 'and earned pay between all employees.

End Sub

End Module