**Program**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace C# 3

{

class Program

{

static void Main(string[] args)

{

ApplicationUtilities.DisplayApplicationInformation();

ApplicationUtilities.DisplayDivider("Start Program");

ApplicationUtilities.DisplayDivider("Prompt for Employee Information and create first employee");

Employee employee1 = new Employee();

employee1.FirstName = InputUtilities.getStringInputValue("First Name");

employee1.LastName = InputUtilities.getStringInputValue("Last Name");

employee1.Gender = InputUtilities.getCharInputValue("Gender");

employee1.Dependents = InputUtilities.getIntegerInputValue("Dependants");

employee1.AnnualSalary = InputUtilities.getDoubleInputValue("Annual Salary");

Console.WriteLine(employee1.ToString());

ApplicationUtilities.PauseExecution();

ApplicationUtilities.DisplayDivider("Number of Employee Object(s):" + Employee.NumberEmployees);

ApplicationUtilities.PauseExecution();

//Updated weekly pay

ApplicationUtilities.DisplayDivider("Get Weekly Pay");

employee1.AnnualSalary = InputUtilities.getDoubleInputValue("Updated Annual Salary");

string thename = employee1.FirstName + " " + employee1.LastName;

Console.WriteLine(thename + " modified weekly pay: " + employee1.CalculateWeeklyPay(employee1.AnnualSalary).ToString("C2"));

ApplicationUtilities.PauseExecution();

//display employee inforamtion

Employee employee2 = new Employee("Sue","Smith", 'F', 3,52000);

Console.WriteLine(employee2.ToString());

ApplicationUtilities.TerminateApplication();

}

}

}

**Application Utilities**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace C# 3B

{

public class ApplicationUtilities

{

public static void DisplayApplicationInformation()

{

Console.WriteLine("Welcome the Basic Employee Program");

Console.WriteLine("CIS247a, Week 3 Lab");

Console.WriteLine("Name: Kevin Nguyen");

Console.WriteLine("This program accepts user input as a string, then makes the \nappropriate data conversion and assigns the value to Employee objects");

Console.WriteLine();

}

public static void DisplayDivider(string outputTitle)

{

Console.WriteLine("\n\*\*\*\*\*\*\*\*\* " + outputTitle + " \*\*\*\*\*\*\*\*\*\n");

}

public static void TerminateApplication()

{

DisplayDivider("Program Termination");

Console.Write("Thank you. Press any key to terminate the program...");

Console.ReadLine();

}

public static void PauseExecution()

{

Console.Write("\nProgram paused, press any key to continue...");

Console.ReadLine();

Console.WriteLine();

}

}

}

**Input Utilities**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace C# 3C

{

public class InputUtilities

{

public static string GetInput(string inputType)

{

string strInput = String.Empty;

Console.Write("Enter the " + inputType + ": ");

strInput = Console.ReadLine();

return strInput;

}

public static string getStringInputValue(string inputType)

{

string value = String.Empty;

bool valid = false;

string inputString = String.Empty;

do

{

inputString = GetInput(inputType);

if (!String.IsNullOrEmpty(inputString))

{

value = inputString;

valid = true;

}

else

{

value = "Invalid input";

valid = false;

}

if (!valid)

Console.WriteLine("Invalid " + inputType + " try again!");

} while (!valid);

return value;

}

public static int getIntegerInputValue(string inputType)

{

bool valid = false;

int value = 0;

string inputString = String.Empty;

do

{

inputString = GetInput(inputType);

if (!(String.IsNullOrEmpty(inputString)))

{

valid = Int32.TryParse(inputString, out value);

}

if (!valid)

Console.WriteLine("Invalid " + inputType + " try again!");

} while (!valid);

return value;

}

public static double getDoubleInputValue(string inputType)

{

bool valid = false;

double value = 0;

string inputString = String.Empty;

do

{

inputString = GetInput(inputType);

if (!(String.IsNullOrEmpty(inputString)))

{

valid = Double.TryParse(inputString, out value);

}

if (!valid)

Console.WriteLine("Invalid " + inputType + " try again!");

} while (!valid);

return value;

}

public static char getCharInputValue(string inputType)

{

bool valid = false;

char value = 'u';

string inputString = String.Empty;

do

{

inputString = GetInput(inputType);

if (!(String.IsNullOrEmpty(inputString)))

{

valid = Char.TryParse(inputString, out value);

}

if (!valid)

Console.WriteLine("Invalid " + inputType + " try again!");

} while (!valid);

return value;

}

}

}

**Employee**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace C# 3D

{

public class Employee

{

private const int MIN\_DEPENDENTS = 0;

private const int MAX\_DEPENDENTS = 10;

private const double MIN\_SALARY = 20000;

private const double MAX\_SALARY = 100000;

private const string DEFAULT\_NAME = "Not Given";

private const char DEFAULT\_GENDER = 'U';

// the variables are declared

private string firstName = DEFAULT\_NAME,lastName = DEFAULT\_NAME;

private char gender = DEFAULT\_GENDER;

private int dependents = MIN\_DEPENDENTS ;

private double annualSalary = MIN\_SALARY ;

private static int numEmployees = 0;

// default constructor - you don't expect the user to put anything in.

public Employee()

{

firstName = DEFAULT\_NAME;

lastName = DEFAULT\_NAME;

gender = DEFAULT\_GENDER;

dependents = MIN\_DEPENDENTS;

annualSalary = MIN\_SALARY;

}

// implement the overloaded constructor

public Employee(string fn, string ln, char gen, int dep, double ans)

{

firstName = fn;

lastName = ln;

gender = gen;

dependents = dep;

annualSalary = ans;

}

// we calculate the weekly pay (52 weeks in a year)

public double CalculateWeeklyPay()

{

return annualSalary / 52;

}

public double CalculateWeeklyPay(double modifiedSalary)

{

return modifiedSalary / 52;

}

// place the override tostring

public override string ToString()

{

string empvar = "============= Employee Information =============";

empvar += "\nName:" + firstName + " " + lastName;

empvar += "\nGender: " + Char.ToUpper(gender);

empvar += "\nDependants: " + dependents;

empvar += "\nAnnualSalary: " + annualSalary.ToString("C2");

empvar += "\nWeeklyPay: " + CalculateWeeklyPay().ToString("C2");

return empvar;

}

public string FirstName

{

get { return firstName; }

set

{

if (value.Trim() == String.Empty || value.Trim() == "")

firstName = DEFAULT\_NAME;

else

firstName = value;

}

}

public string LastName

{

get { return lastName; }

set

{

if (value.Trim() == String.Empty || value.Trim() == "")

lastName = DEFAULT\_NAME;

else

lastName = value;

}

}

public char Gender

{

get { return gender; }

set

{

if (Char.ToUpper(value) == 'F' || Char.ToUpper(value) == 'M')

gender = value;

else

gender = DEFAULT\_GENDER;

}

}

public int Dependents

{

get { return dependents; }

set

{

if (value < MIN\_DEPENDENTS)

dependents = MIN\_DEPENDENTS;

else if (value > MAX\_DEPENDENTS)

dependents = MAX\_DEPENDENTS;

else

dependents = value;

}

}

public double AnnualSalary

{

get { return annualSalary; }

set

{

if (value < MIN\_SALARY)

annualSalary = MIN\_SALARY;

else if (value > MAX\_SALARY)

annualSalary = MAX\_SALARY;

else

annualSalary = value;

}

}

public static int NumberEmployees

{

get

{

return numEmployees += 1;

}

}

~Employee() { }

}

}

