Assignment 6.

Create a class that performs sorting for any class objects, built in types etc

based on sorting algorithm of your choice. Write a driver program to test the

applications. Ensure relevant use of delegates and generic methods wherever

applicable

Consider the following design goal: write a delegate that prints out an

appropriate message for the client depending on the balance in a bank

account. Here are the four messages:

If balance < 0 then print "you are overdrawn"

If balance < 10 then print "your account is very low"

If balance < 100 then print "watch your spending carefully"

Else print "you have over $100 in your account"

This delegate will capture the current value of the nonlocal variable balance

to determine which message to print.

Code:

using System.Collections.Generic;

using System;

class Programme

{

static public void Sort<T>(IList<T> sortArray, Func<T, T, bool> compare)

{

for (int i = 0; i < sortArray.Count - 1; i++)

{

for (int j = i + 1; j < sortArray.Count; j++)

{

if (compare(sortArray[j], sortArray[i]))

{

T temp = sortArray[i];

sortArray[i] = sortArray[j];

sortArray[j] = temp;

}

}

}

}

}

public enum Position

{

Manager = 1,

Dean = 2,

Assistant\_Mana = 6,

Vice\_Dean = 4,

Professor = 5,

Experiment\_Director = 3

}

class University

{

private int id;

private float salary;

private string name;

private Position position;

public int ID

{

get { return id; }

set { id = value; }

}

public float Salary

{

get { return salary; }

set { salary = value; }

}

public string Name

{

get { return name; }

set { name = value; }

}

public Position Position

{

get { return position; }

set { position = value; }

}

public University(int Id, float Salary, string Name, Position position)

{

this.id = Id;

this.salary = Salary;

this.name = Name;

this.position = position;

}

internal static bool CompareSalary(University u1, University u2)

{

return u1.salary < u2.salary;

}

internal static bool ComparePosition(University u1, University u2)

{

return u1.position > u2.position;

}

}

public class TestGenericMethods

{

public static void Main(string[] args)

{

List<University> list = new List<University>(10);

list.Add(new University(5, 200000, "M", Position.Manager));

list.Add(new University(2, 120000, "D", Position.Dean));

list.Add(new University(4, 33000, "Am", Position.Assistant\_Mana));

list.Add(new University(1, 110000, "VC", Position.Vice\_Dean));

list.Add(new University(3, 80000, "Proff", Position.Professor));

list.Add(new University(3, 65000, "Dir", Position.Experiment\_Director));

Programme.Sort<University>(list, University.CompareSalary);

Console.WriteLine("Sorting On Salary : ");

foreach (University u in list)

{

Console.WriteLine($"{u.ID}) {u.Name} , {u.Position} {u.Salary}");

}

Programme.Sort<University>(list, University.ComparePosition);

Console.WriteLine("Sorting On Position : ");

foreach (University u in list)

{

Console.WriteLine($"{u.ID}) {u.Name} , {u.Position} {u.Salary}");

}

}

}

OutPut:

