C# Performance Task

Code -

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections;

namespace Day\_14\_test

{

//class 1 contain basic info

class basic\_d

{

public string g\_name, g\_city, g\_state, g\_cast, g\_mobile, g\_country;

public void input(string d\_name, string d\_city, string d\_state, string d\_cast,

string d\_mobile, string d\_country)

{

g\_name = d\_name;

g\_city = d\_city;

g\_state = d\_state;

g\_cast = d\_cast;

g\_mobile = d\_mobile;

g\_country = d\_country;

}

}

//class 2 contain qualification info

class edu\_d : basic\_d

{

public string g\_c\_name, g\_branch, g\_sub, g\_sem, g\_e\_c\_city;

public void edu\_input(string e\_c\_name, string e\_branch, string e\_sub, string e\_sem, string e\_c\_city)

{

g\_c\_name = e\_c\_name;

g\_branch = e\_branch;

g\_sub = e\_sub;

g\_sem = e\_sem;

g\_e\_c\_city = e\_c\_city;

}

}

//class 3 contain Marks

class mark\_d : edu\_d

{

public int g\_total;

public SortedList<string, int> g\_mark;

public void mark\_total(SortedList<string, int> mark)

{

int total = 0;

foreach (var i in mark.Keys)

{

total = total + mark[i];

}

g\_total = total;

g\_mark = mark;

}

}

//class 4 grade logic and printing info

class display\_d : mark\_d

{

public string g\_garde;

public void display()

{

Console.WriteLine("\nYour Name is : " + g\_name +

"\nCollege Name : " + g\_c\_name +

"\nTotal is : " + g\_total

); ;

if (g\_total > 250)

{

g\_garde = "Grade A";

Console.WriteLine("\nGrade A");

}

else if (g\_total > 150 && g\_total < 250)

{

g\_garde = "Grade B";

Console.WriteLine("\nGrade B");

}

else if (g\_total > 100 && g\_total < 150)

{

g\_garde = "Grade C";

Console.WriteLine("\nGrade C");

}

else

{

g\_garde = "Fail";

Console.WriteLine("\nFail");

}

}

}

//class 5 - printing marksheet

class marksheet\_d : display\_d

{

public void display\_marksheet()

{

Console.Write("\nName : " + g\_name +

"\nCollege : " + g\_c\_name +

"\nSubject : " + g\_mobile +

"\nBranch : " + g\_branch

);

int sno = 1;

Console.WriteLine("\n\nSno\t|\tSubject Name\t|\tMarks");

foreach (var ii in g\_mark.Keys)

{

Console.WriteLine(sno + "\t|\t" + ii + "\t\t|\t" + g\_mark[ii]);

sno++;

}

Console.WriteLine("\n\n");

Console.WriteLine("Total is : " + g\_total);

Console.WriteLine("Grade is : " + g\_garde);

Console.WriteLine("Develped by : Pushpak Fasate");

}

}

class Program

{

//this contain only basic info variable

struct basic

{

public string name;

public string city;

public string state;

public string cast;

public string mobile;

public string country;

};

//this contain only education variable

struct education

{

public string college;

public string branch;

public string subject;

public string sem;

public string c\_city;

};

static void Main(string[] args)

{

basic b;

education e;

marksheet\_d dis = new marksheet\_d();

bool val = true;

Console.WriteLine("COLLEGE STUDENT MARK SYSTEM ");

Console.WriteLine("==============================");

//To run this loop up till user give input 6

while (val)

{

//Giveing option to user

Console.Write("\n[1] Basic Details \n[2] Education Details \n[3] Subject Details" +

"\n[4] Display Total and Grade \n[5] Display Marksheet\n[6]Press 6 for Exit\n\nEnter Your Choice : ");

int ch = int.Parse(Console.ReadLine());

//suppose user give ch value 6 then loop get exit

if (ch == 6)

{

val = false;

}

else

{

//according to the user the swtich case will run

switch (ch)

{

case 1:

//For the Screen - 1

Console.WriteLine("\nBasic Details : ");

Console.Write("Enter Name : ");

b.name = Console.ReadLine();

Console.Write("Enter City : ");

b.city = Console.ReadLine();

Console.Write("Enter State : ");

b.state = Console.ReadLine();

Console.Write("Enter Cast : ");

b.cast = Console.ReadLine();

Console.Write("Enter Mobile : ");

b.mobile = Console.ReadLine();

Console.Write("Enter Country : ");

b.country = Console.ReadLine();

//data can not be null or empty

try

{

if (string.IsNullOrEmpty(b.name) || string.IsNullOrEmpty(b.city) || string.IsNullOrEmpty(b.state)

|| string.IsNullOrEmpty(b.cast) || string.IsNullOrEmpty(b.mobile) || string.IsNullOrEmpty(b.country))

{

throw new Exception("All Field are require");

}

}

catch (Exception bas)

{

Console.WriteLine(bas);

}

dis.input(b.name, b.city, b.state, b.cast, b.mobile, b.country);

break;

case 2:

//For the Screen - 2

Console.WriteLine("\nEducation Details : ");

Console.Write("Enter College Name : ");

e.college = Console.ReadLine();

Console.Write("Enter Branch Name : ");

e.branch = Console.ReadLine();

Console.Write("Enter Subject Name : ");

e.subject = Console.ReadLine();

Console.Write("Enter Semester : ");

e.sem = Console.ReadLine();

Console.Write("College City : ");

e.c\_city = Console.ReadLine();

//data can not be null or empty

try

{

if (string.IsNullOrEmpty(e.college) || string.IsNullOrEmpty(e.branch) || string.IsNullOrEmpty(e.subject)

|| string.IsNullOrEmpty(e.sem) || string.IsNullOrEmpty(e.c\_city))

{

throw new Exception("All Field are require");

}

}

catch (Exception bas)

{

Console.WriteLine(bas);

}

dis.edu\_input(e.college, e.branch, e.subject, e.sem, e.c\_city);

break;

case 3:

//For the Screen - 3

Console.WriteLine("\nSubject Details : ");

SortedList<string, int> allmark = new SortedList<string, int>(5);

int i3 = 1;

for (int i = 0; i < 5; i++)

{

Console.Write("Enter Subject-" + i3 + " : ");

allmark.Add(Console.ReadLine(), int.Parse(Console.ReadLine()));

i3++;

}

//marks can not be in negative value

try

{

foreach (var i in allmark.Keys)

{

if (allmark[i] < 0)

{

throw new Exception("Mark can not be negative");

}

}

}

catch (Exception bas)

{

Console.WriteLine(bas);

}

dis.mark\_total(allmark);

break;

case 4:

//For the Screen - 4

dis.display();

break;

case 5:

//For the Screen - 5

dis.display\_marksheet();

break;

case 6:

//For exit

break;

default:

Console.Write("\n\nInvalid\n\n");

break;

}

}

}

Console.Write("Thank You !");

Console.ReadKey();

}

}

}

Output:







