WF Ass1

Completion and Hand up Tuesday 3rd November
Must Be Completed as GUI Apps.

Part A.

Question 1.

- Using **the C# 'Help' system** find out about **random numbers**, write a small application simulating Lotto, this allows a user to generate a 'Lotto Line' and the method will display 6 random numbers between 1 and 49. You **may ignore** duplicate numbers etc in the "Lotto Line".

- Modify the method in Q1 to display the numbers generated in **ascending** order and to compensate for duplicate numbers.

Question 2.

The following table shows the average seasonal rainfall in millimetres in Ireland in the years 2002 through 2005 :

	Year			
Season	**2016**	**2017**	**2018**	**2019**
Spring	150	163	147	138
Summer	100	89	88	87
Autumn	157	97	96	94
Winter	184	133	129	117

- Design an appropriate data structure to hold this data. Create a suitable method for data entry and display.

- Create Command buttons which will display : Driest Year, Wettest Year, Driest Season, and Wettest Season.

- Design an appropriate interface which will allow a user to choose any year (including all years) and any season (including all seasons) and which when clicked which will display the rainfall for the chosen combination.

Question 3.

There is a 3 race card at leopardstown and the runners are as shown below :

2: 20 ::::: Blue Jay, Fireside and Not Again
2: 50 ::::: Summers Night, Coriolanus, Blue Rinse, Silver Shadow and SLK
3: 30 ::::: Purple Rain, Last Ditch, Forty Fives and Too Double

Represent the data as a 2-D Jagged Array. Create a method with List Runners Buttons for each of the three races which will display the runners for the chosen race.

Part B.

There is an class named hseEmployee made up of the following data :

empName : String
empNumber : int *(emp number for new employees is assigned automatically in increments of 100)*
empType :String
empYrsService :int
empSalary :double

The private member variable empNumber is read only (as was noted it is assigned automatically in increments of 100).

- Create this class.

- Create a **non-parameterized constructor** for this Class and a **toString** method which will print the following details for a hseEmployee object. The statement to generate it is : ***printDetails(employee1);***

Emp Name : A. N. Other
Emp Number : 100
Emp Type : Standard
Emp Yrs Service : 0
Emp Salary : €30,000.00

Points to note : the formatting of the salary in example output shown above.

Please turn page over ……

- This part uses a parametrised constructor which takes input from the user and creates an appropriate hseEmployee object. The test input and output are shown below:

  Input from keyboard

Enter employee name : Tony McCarron
Enter employee type : Porter
Enter employee years of service : 4
Enter employee salary : 34576.7893

Output on screen

Emp Name : Tony McCarron
Emp Number : 200
Emp Type : Porter
Emp Yrs Service : 4
Emp Salary : €34,576.79

The class hseEmployee has a subclass named Doctor whose empName, empType and EmpSalary are shown in the output below. The values shown are the default values for a Doctor object. Note that also the **printDetails method adds the text to the output :**
I can PRESCRIBE for patients!!!

- Create a Doctor object, test and display it using the statement : ***printDetails(myDoctor);***

Sample output for the Doctor object is shown below.

Emp Name : Dr. A. N. Other
Emp Number : 300
Emp Type : Doctor
Emp Yrs Service : 0
Emp Salary : €110,000.00
I can PRESCRIBE for patients!!!

- Create an structure of 10 random hseEmployee objects (Doctor, Porter and Standard employees) and display the details of each of.