# PCS4 Exam – June 2016

**Date: 21 June, 2017**
**Time: 8.45 – 11.15 h (150 minutes).**

**ADMITTED RESOURCES:**

- You are allowed to use everything on paper (books, notes, etc.) and on your laptop, but only what you bring in: you are not allowed to borrow something from someone else.
- During the exam, it is not allowed to use the network. You should make the exam yourself: so, no communication with MSDN or google for help and no communication with other students, like using facebook, e-mail, skype, Dropbox, gsm or whatever.

**THE APPLICATION**

For an upcoming charity event, a friend has made an application that simulates participants competing against each other in a running contest. However, the application is not finished yet. Of course, the application could have more functionalities but in this exam we consider only a part of it.


The image on the right displays the application when it is running. When the user clicks on the 'Start' button, the participants will start moving towards the finish-line. When a participant finished, he/she will be added to the result listbox in the right lower corner. With the 'Reset' button, the participants will be placed at the start again.
The 'Display stats' button allows the user to see the winners of previous races as well as how many times they won.
Next, there are two sort buttons, that allow the user to sort all participants in certain ways. The results should be visible in the listbox on the right lower corner. Last but not least, the 'Display youngest' button should display the youngest participant (for example in a MessageBox).
An example would be that a user lets the participants race four times:

- click 'Start'
- wait for the participants to finish
- click 'Reset'

Then when the user clicks 'Display stats', the system displays statistical information of these four races.

**PRELIMINARY REMARKS:**

- In this exam, you will receive three (partly) implemented classes, as seen in the class overview below: Participant, Race and RaceForm.
    - Whenever you see a region with the text '`This code is implemented for you`', this is code from your friend. You should not need to make changes. If you do decide to change code, mention this as follows above the ***region***:
      // Code changed by: <your name>
        - Of course, you may make use of the methods that are already implemented.

- This race specifically hosts exactly **six** participants, each with a unique name. As you can see in the RaceForm class, this is done in the method `InitializeParticipants()`. If you like you may change their properties, but you will not be rewarded points.

## Assignment 1: Racing and reaching the finish-line (5 + 25 pts).

In this assignment, you will implement the code to: make the participants race, and add information of the finished participants in the listbox lbInfo. The 'Participant' class already contains information, including: Name, Age, Hobby, X (How far the participant ran on the track).

**a.** Currently, if you press that 'Start' button, the participants will not move. Implement the Update method so that the 'Participant' will move horizontally with every update by increasing the 'X' property with its 'Speed' property. When the participant crosses the finish-line, the Boolean property 'isFinished' should be set to true.
**Make the Update method work; Move the participant towards the finish-line unless the participant reached it.**

**b.** When a participant finished the race, the RaceForm should be notified. The information about the participant should be shown in the listbox lbInfo.
**Make this feature working by using events.**
*(Solutions that do not use events will not receive points)*

## Assignment 2: Linked list (25 pts).

When pressing the 'Display stats' button, the user should see who has won and how many times. For this purpose, you should implement a linked list of items. Every item has information about a participant name and how many time he/she won. If a race is ended, the winner should be registered in the linked list. If a winner is not part of the linked list yet, add a new item about this winner. However, if the participant is already part of the linked list, increment the amount of times that this participant won.
Make it also possible to return the information from the items in the linked list.
*In other words; in this linked list, we only want information about the participants that won at least one time.*
**Implement the linked list in which you store the required information and use it in the application. By clicking the 'Display stats' button the user gets information on the screen about the items from the linked list. Make it work.**

## Assignment 3: Sorting (10 + 15 pts).

As mentioned in the introduction the user is also supposed to be able to sort the list of **all** participants. It should be possible to do this in two ways, by:

- Name (alphabetically)
- Age (from high to low) and then Hobby (alphabetically)

Use two different ways of sorting from the following three ways: `IComparable`, `Comparison` or `IComparer`.
Both buttons 'Sort by Name' and 'Sort by Age then Hobby' should sort the list in the proper way and displays the participants in the listbox lbInfo.

**a.** **Sort the list of participants by Name. Implement the eventHandler of the 'Sort by Name' button**

**b.** **Sort the list of participants by Age, then Hobby. Implement the eventHandler of the 'Sort by Age then Hobby' button**

## Assignment 4: Recursion (20 pts).

In the Race class, there must be a method that returns a youngest participant. You should implement this method by using recursion. The 'Display youngest' button should display information about a youngest participant.

**Add a recursive method that finds a youngest participant. Implement the eventHandler of the 'Display youngest' button.**

**End of exam.**