Module name	CW number and title
Fundamentals of Programming	**Coursework 1**

CW weighting	CW checks the learning outcomes
40%	3. understand and implement fundamental data structures using classes 4. understand and implement the concept of persistence in relation to objects
Submission deadline	
Apr 4, 2013	
CW format (individual/group)	
Individual	

CW Task

"SuperDrive" is a driving school operating in your city. The school runs a series of classes for its students as well as an internal exam to prepare the students for the actual test held by Governmental Road Police Department. The exam is run in the FIFO (first in first out) manner so that students are assigned to the test according to the time they entered the school – the earlier a student has entered the school the earlier he/she will take the exam.

The manager of SuperDrive wants you to create an Information System to simplify the operations of the school namely students administration and selection of candidates for internal exam. You are given a database designed to hold all required information (see superdrive.mdb file). The database has the following structure:

Table st_student – holds students' information

st_id	Artificial Id for the table (autonumber)
st_name	Student's name
st_address	Student's address
st_dob	Student's date of birth
st_entry_date	Entry date to the school

Table ex_exam – holds information on internal exams

ex_id	Artificial Id for the table (autonumber)
ex_date	Exam date
ex_capacity	How many students can attend the exam

As a result of a series of interviews with employees of the school you have identified the following requirements:

1. Manage students personal information
    - Display all students registered in the school
    - Enable addition/editing of students' records
    - Enable deletion of students' records
    - Persist students information in the database
2. Facilitate students management
    - Allow sorting the students' records by Name and Entry date
    - Allow fast searching for a student by exact name
3. Manage exams
    - Enable creation of new exams (no need for edit and delete functionality)
    - Persist exam information in database
4. Prepare list of students for exams
    - A user should be able to generate a list of participating students for an exam. Students' records should be placed into queue data structure according to the entry date (FIFO) and then the required number of students (determined by Capacity attribute of the exam) should be de-queued and shown on the form. The user should be notified if the number of students registered is less than the exam capacity (e.g. via a message box). You are **NOT** required to persist this information as a part of exam persistence.

Important notes: you are allowed to use standard .Net mechanisms for sorting, searching as well as use Queue class. However in order to gain distinctive mark you should provide your own implementation of at least one of the algorithms or your own implementation of queue data structure.
Be sure to use object-oriented approach as well as comment you code properly.

Format for submission of work

Your code (whole solution with all projects) should be submitted to Registrar's Office on a CD/DVD disc alongside with filled CA1 form not later than the deadline stated above. You are not required to produce any kind of written report as a part of this coursework
Coursework must be submitted electronically via University's Intranet.
This is your responsibility to put CW through the Turn-it-in anti plagiarism software before submission.

Your name should not appear on the cover page or anywhere else. Put your ID number on the cover page and on every other page (follow a template).

CW Assessment Criteria

Component	Weight (marks)
General requirements	**15**
The program compiles	3
Comments present	5
Classes created for business logic and DAL	7
Required functionality	**65**
Display all students registered in the school	10
Enable addition/editing of students' records	5
Enable deletion of students' records	5
Persist students information in the database	10
Allow sorting the students records by Name and Entry date	5
Allow fast searching for a student by exact name	5
Enable creation of new exams	5
Persist exam information in database	5
Implementation of exam queue	15
Advanced functionality	**20**
Custom sort/search	20