COMP 1020 Winter 2016 Assignment 1

Due Date: Friday, Feb. 5, before midnight

1 MATERIAL COVERED

- Basic objects (*review*) and proper data hiding
- Instance variables and methods
- Static/class variables and methods
- Partially-full arrays (*review*)
- Linear search (*review*) and object comparison

2 NOTES AND INSTRUCTIONS—PLEASE FOLLOW *EXACTLY*

- Follow the programming standards posted on the course website to avoid losing marks.
- You must complete the "Blanket Honesty Declaration" checklist on the course website, before you can submit any assignment.
- To submit the assignment you will upload the required files, as specified for each question, to the Dropbox for Assignment 1 on the course website.
- You will also put the output of your test runs from Dr. Java into .txt files, and upload them to the Dropbox in the same way.
- Name your output files <LastName><FirstName>A1Qn-output.txt.
- Do **not** manipulate your output in any way.
- Complete instructions for handing in your assignment, including an easy way to create the output files, can be found in the "Handin Instructions" document in the Quick Links on the website.
- To be eligible to earn full marks, your Java programs must compile and run upon download, without requiring any modifications.
- **Do not** hand in zipped or otherwise compressed files. Hand in **only and exactly** those files specified in this assignment PDF, in the specified formats. Failure to do so will result in lost marks.

3 QUESTION 1: SIMPLE COURSE CLASS

3.1 GENERAL DESCRIPTION

This is part 1 of a 3-part assignment. For this question, you will write a simple `Course` class that will enable universities to use a rudimentary database for tracking the classes they offer.

3.2 THE COURSE CLASS

Create a new Java file, and name it Course.java. (You do not need to prefix it with your name this time.) Name the class appropriately to match. You have been provided with a main program file, called TestA1Q1.java.

Create five *instance* variables: a String for the course title, a String for the department offering the course, an integer for the program year of the course (1st year course, 2nd year course, etc.), an integer for the term (1, 2 or 3, indicating *Winter*, *Spring* and *Fall* terms) and a String course registration number (CRN) that uniquely identifies the course. All variables should be private.

Additionally, create a *class* variable called `numCourses`, an integer, which will keep track of the number of `Course` objects that have been created. Initialize it to zero.

Create only a single constructor that accepts four parameters: the course title, the department, the term and the program year of the course. This constructor should also assign a **unique** CRN to the `Course`. All CRNs start with "2016", followed by the term (a single digit), and finally a sequence number (3 digits, starting at "001", and incrementing by 1 each time a course is added. Use the `numCourses` variable to do this.).

Write accessor ("get" or "getter") methods for all five instance variables. Do not define any mutators ("set" or "setter" methods).

Define a `toString()` method that returns the CRN, term, title, department, and year, in the format shown in the lines of output in the example below (not including the leading blanks). Note that the term is printed as names, not the numeric value. Declare a constant array of Strings for printing the term names (i.e. don't use if-statements).

3.3 SAMPLE OUTPUT

Here is the output that should be produced when running `TestA1Q1.java`:

```
There are 6 courses available:
        CRN: 20161001: Term: Winter, Advanced YouTube Commenting, Dept: Comp Sci (year 3)
        CRN: 20161002: Term: Winter, Basket Weaving, Dept: Fine Arts (year 1)
        CRN: 20162003: Term: Spring, Intermediate Cereal Cooking, Dept: Culinary Arts (year 2)
        CRN: 20163004: Term: Fall, Potato Trees, Dept: Agriculture (year 2)
        CRN: 20161005: Term: Winter, Paper Airplanes, Dept: Mechanical Engineering (year 4)
        CRN: 20162006: Term: Spring, Zookeeping, Dept: Animal Science (year 1)
End of processing. Programmed by Stew Dent.
```

3.4 HAND-IN

Submit your `Course.java` file (but **not** the main program you were given) and also a text file containing the output of your program. Name your output file as specified in Section 2. Make your output match the appearance of the sample output as closely as possible.

4 QUESTION 2: YOUR OWN LIST OBJECT

4.1 GENERAL DESCRIPTION

This is part 2 of a 3-part assignment. For this question, you will write a simple `CourseListQ2` class that will allow you to group together multiple `Course` objects inside of your own list structure. For now, this list structure will only allow you to add or remove new `Course`s to it, but the third part of this assignment will incorporate more advanced features.

4.2 THE COURSELISTQ2 CLASS

Create a new Java file, and name it CourseListQ2.java. (You do not need to prefix it with your name this time.) Name the class appropriately to match. You have been provided with a main program file, called TestA1Q2.java. You **do not** need to create a copy of your original Question 1 `Course` Java file. It can be used as-is.

Create two *instance* variables: an uninitialized array of `Course`s, and an integer `numCourses` that we will use to specify how many courses are currently in the list.

Define only a default constructor. It should initialize the array of `Course`s to a size of 5 (remember to use constants as appropriate), and set `numCourses` to 0, because our list will initially contain no courses.

Define a void method called `addCourse(Course)`, which takes a single `Course` parameter. It should insert the new `Course` object into the next available position in the `Course` array, which is easily determined by the value of `numCourses`. Use an expanding, partially-full array approach – if too many `Course`s are added, then double the size of the array and copy the elements over into the new array before adding the new `Course`.

Define an integer method called `indexOf(String)`, which takes a single String parameter representing a CRN. It should return the index of the `Course` in this list with this CRN, or -1 if it is not found. Remember you must use the `equals` method to compare Strings.

Define a method called `removeCourse(String)`, which takes a single String parameter representing the CRN of the `Course` to delete from the array. It returns a reference to the `Course` that was deleted, if it was found, otherwise it returns `null`. Perform a search to find the appropriate `Course`. [HINT: you have already written a method that performs this search; USE IT!] If it is found, the method should "delete" the `Course` by **replacing the deleted item with the last filled item in the array**, thereby overwriting the `Course` to remove. This effectively deletes it from the list. The method should return a reference to the deleted course if the item was found and deleted, or `null` if it could not be found. You do not need to "shrink" your array when elements are removed.

Define a `toString()` method that returns the complete list of `Course`s in the list. It should use the `toString()` method for `Course` objects. It will return one long `String` containing newline characters and tab characters, as appropriate.

4.3 SAMPLE OUTPUT

```
Adding courses to list...
There are 13 courses available:
        CRN: 20161001: Term: Winter, Structure and Modelling, Dept: Chemistry (year 1)
        CRN: 20162002: Term: Spring, Design in Engineering, Dept: Engineering (year 1)
        CRN: 20163003: Term: Fall, Literary Topics, Dept: English (year 1)
        CRN: 20162004: Term: Spring, Biomachinery, Dept: Biosystems (year 4)
        CRN: 20163005: Term: Fall, Aristotle, Dept: Philosophy (year 2)
        CRN: 20161006: Term: Winter, Finite Element Analysis, Dept: Engineering (year 3)
        CRN: 20163007: Term: Fall, Plant and Animal Physiology, Dept: Agriculture (year 2)
        CRN: 20162008: Term: Spring, Natural Resources, Dept: Geography (year 2)
        CRN: 20163009: Term: Fall, Health and Disease, Dept: History (year 4)
        CRN: 20162010: Term: Spring, Ethics and Biomedicine, Dept: Philosophy (year 2)
        CRN: 20161011: Term: Winter, Programming Language Concepts, Dept: Comp Sci (year 3)
        CRN: 20163012: Term: Fall, Real-Time Systems, Dept: Comp Sci (year 4)
        CRN: 20161013: Term: Winter, Culture and Environment, Dept: Geography (year 2)

Removing courses from list...
Deleted CRN: 20163007: Term: Fall, Plant and Animal Physiology, Dept: Agriculture (year 2)
CRN: 21602007 not found in course list
Deleted CRN: 20163005: Term: Fall, Aristotle, Dept: Philosophy (year 2)
Deleted CRN: 20161001: Term: Winter, Structure and Modelling, Dept: Chemistry (year 1)

 Current course list...
There are 10 courses available:
        CRN: 20162002: Term: Spring, Design in Engineering, Dept: Engineering (year 1)
        CRN: 20163003: Term: Fall, Literary Topics, Dept: English (year 1)
        CRN: 20162004: Term: Spring, Biomachinery, Dept: Biosystems (year 4)
        CRN: 20163012: Term: Fall, Real-Time Systems, Dept: Comp Sci (year 4)
        CRN: 20161006: Term: Winter, Finite Element Analysis, Dept: Engineering (year 3)
        CRN: 20161013: Term: Winter, Culture and Environment, Dept: Geography (year 2)
        CRN: 20162008: Term: Spring, Natural Resources, Dept: Geography (year 2)
        CRN: 20163009: Term: Fall, Health and Disease, Dept: History (year 4)
        CRN: 20162010: Term: Spring, Ethics and Biomedicine, Dept: Philosophy (year 2)
        CRN: 20161011: Term: Winter, Programming Language Concepts, Dept: Comp Sci (year 3)

End of processing.
Programmed by Stew Dent.
```

4.4 HAND-IN

Submit your `CourseListQ2.java` file (but **not** the main program you were given) and also a text file containing the output of your program. Name your output file as specified in Section 2. Make your output match the appearance of the sample output as closely as possible.

5 QUESTION 3: YOUR ADVANCED COURSELIST

5.1 GENERAL DESCRIPTION

This is part 3 of a 3-part assignment. For this question, you will expand on your `CourseListQ2` class to incorporate more advanced features.

5.2 MORE HELPFUL COURSELIST METHODS

Create a duplicate of your `CourseListQ2.java` file, and name it `CourseListQ3.java`. (You do not need to prefix it with your name this time.) Name the class appropriately to

match. You have been provided with a main program file, called `TestA1Q3.java`. You **do not** need to create a copy of your original Question 1 `Course` Java file.

Define a `getCoursesByDept(String)` method that accepts a `String` parameter representing a department. This method should return a `CourseListQ3` object containing any `Courses` in your list that are offered by the specified department.

Define a method called `getCoursesByTerm(int)`. This method should return a `CourseListQ3` object containing any `Courses` in your list whose term matches the given integer.

Define a void method called `removeListOfCourses(CourseListQ3)`. This method should delete all courses in the argument list. Use the `removeCourse` method to perform the actual deletions.

5.3 SAMPLE OUTPUT

```
There are 13 courses available:
        CRN: 20161001: Term: Winter, Natural Resources, Dept: Geography (year 2)
        CRN: 20162002: Term: Spring, Real-Time Systems, Dept: Comp Sci (year 4)
        CRN: 20163003: Term: Fall, Culture and Environment, Dept: Geography (year 2)
        CRN: 20162004: Term: Spring, Aristotle, Dept: Philosophy (year 2)
        CRN: 20163005: Term: Fall, Design in Engineering, Dept: Engineering (year 1)
        CRN: 20163006: Term: Fall, Literary Topics, Dept: English (year 1)
        CRN: 20161007: Term: Winter, Structure and Modelling, Dept: Chemistry (year 1)
        CRN: 20162008: Term: Spring, Finite Element Analysis, Dept: Engineering (year 3)
        CRN: 20161009: Term: Winter, Plant and Animal Physiology, Dept: Agriculture (year 2)
        CRN: 20163010: Term: Fall, Biomachinery, Dept: Biosystems (year 4)
        CRN: 20161011: Term: Winter, Health and Disease, Dept: History (year 4)
        CRN: 20163012: Term: Fall, Ethics and Biomedicine, Dept: Philosophy (year 2)
        CRN: 20163013: Term: Fall, Programming Language Concepts, Dept: Comp Sci (year 3)

Cancelling courses offered by Engineering and Geography...
There are 9 courses available:
        CRN: 20161011: Term: Winter, Health and Disease, Dept: History (year 4)
        CRN: 20162002: Term: Spring, Real-Time Systems, Dept: Comp Sci (year 4)
        CRN: 20163010: Term: Fall, Biomachinery, Dept: Biosystems (year 4)
        CRN: 20162004: Term: Spring, Aristotle, Dept: Philosophy (year 2)
        CRN: 20163013: Term: Fall, Programming Language Concepts, Dept: Comp Sci (year 3)
        CRN: 20163006: Term: Fall, Literary Topics, Dept: English (year 1)
        CRN: 20161007: Term: Winter, Structure and Modelling, Dept: Chemistry (year 1)
        CRN: 20163012: Term: Fall, Ethics and Biomedicine, Dept: Philosophy (year 2)
        CRN: 20161009: Term: Winter, Plant and Animal Physiology, Dept: Agriculture (year 2)

List of remaining Term 3 courses:
There are 4 courses available:
        CRN: 20163010: Term: Fall, Biomachinery, Dept: Biosystems (year 4)
        CRN: 20163013: Term: Fall, Programming Language Concepts, Dept: Comp Sci (year 3)
        CRN: 20163006: Term: Fall, Literary Topics, Dept: English (year 1)
        CRN: 20163012: Term: Fall, Ethics and Biomedicine, Dept: Philosophy (year 2)

End of processing.
Programmed by Stew Dent.
```

5.4 HAND-IN

Submit your `CourseListQ3.java` file (but **not** the main program you were given) and also a text file containing the output of your program. Name your output file as specified in Section 2. Make your output match the appearance of the sample output as closely as possible.