The Hong Kong Polytechnic University
Department of Electronic and Information Engineering

EIE3320 Tutorial 4: Object-Oriented Programming (Inheritance)

(Deadline for Submission: Check the course information)

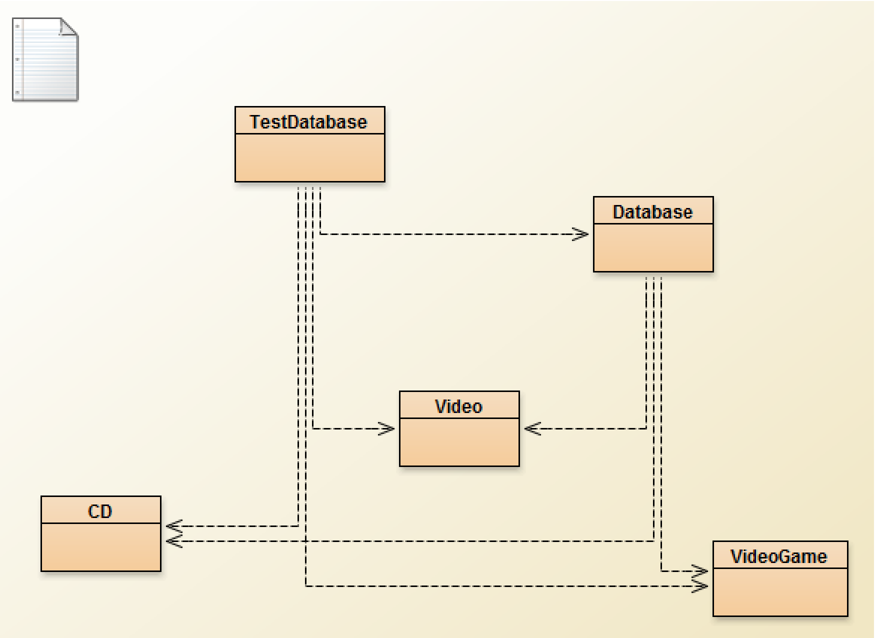
1. **(Assignment)** Create a new class `VideoGame`.

   a. Download the BlueJ project "demo-v1" from http://www.eie.polyu.edu.hk/~encccl.

   b. Add a new class called "`VideoGame`". Add instance members "`title`", "`platform`", "`numberOfPlayers`", "`playingTime`", "`gotIt`" and "`comment`" to the new class. Moreover, add methods `setComment()`, `getComment()`, `setOwn()`, `getOwn()`, and `print()` to the new class.

   c. Modify the class `Database`:
      i. Create an array list called `videoGames` to store some `VideoGames` objects.
      ii. Add a new method `addVideoGame(VideoGame theVideoGame)` which is used to add a `VideoGame` object to the array list `videoGames`.
      iii. Modify the method `list()` to print all `CD`, `Video` and `VideoGame` objects.

   d. Use the class `TestDatabase` (uncomment all statements related to the class `VideoGame` inside the class file) to check the correctness of your program codes. The class diagram and the output should be shown below:

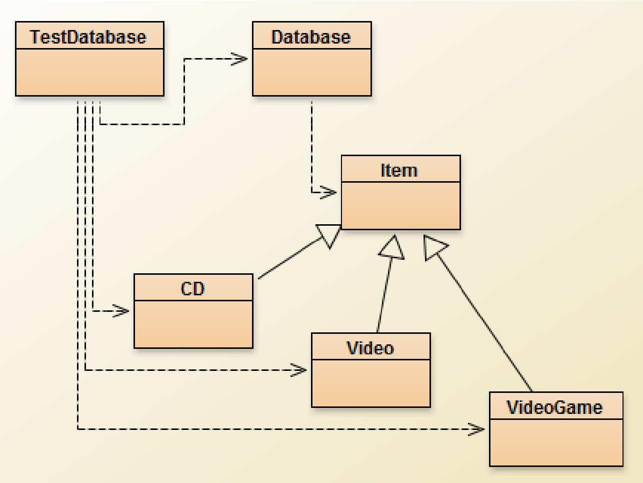


```
CD
Title: Joey & Joey
Got it: Yes
Playing time: 60
Comment: My favorite album
Artist: Joey Yung
No. of tracks: 11

Video
Title: Matrix
Got it: Yes
Playing time: 137
Comment: My favorite movie
Director: Wachowski Brothers

VideoGame
Title: Final Fantasy
Got it: No
Playing time: 30
Comment: My favorite VideoGame
Platform: Windows
No. of players: 2
```

2. **(Assignment)** Build inheritance for the class `CD`, `Video` and `VideoGame`.

   a. Download the BlueJ project “dome-v2-dorisl” from http://www.eie.polyu.edu.hk/~encccl.

   b. Create the class `CD` and `Video`, and modify the class `VideoGame`. Their instance members and methods should be the same as that in Question 1 but this time you need to make use of the superclass `Item`. Note that some instance members and methods built in the superclass are not required to be built again in the subclass. You should modify the codes in this project to get the following class hierarchy. The output of the class `TestDatabase` is shown on the next page (uncomment all statements related to the class `CD`, `Video` and `VideoGame` inside the class file).



```
CD
Title: Joey & Joey
Got it: Yes
Playing time: 60
Comment: My favorite album
Artist: Joey Yung
No. of tracks: 11

CD
Title: Miss You Mix
Got it: Yes
Playing time: 70
Comment: Best album
Artist: Leslie Cheung
No. of tracks: 12

Video
Title: Matrix
Got it: Yes
Playing time: 137
Comment: My favorite movie
Director: Wachowski Brothers

Video
Title: Star Trek
Got it: Yes
Playing time: 127
Comment:
Director: J. J. Abrams

VideoGame
Title: Final Fantasy
Got it: Yes
Playing time: 30
Comment: My favorite VideoGame
Platform: Windows
No. of players: 2
```

3. Download project files from http://www.eie.polyu.edu.hk/~enhylin/BlueJProjects.zip and decompress the zip file to your home directory.

   a. Invoking BlueJ and open the project "notebook2" in Chapter 4 by clicking "File" → "Open Project".

   b. Create a `Notebook` object. Store a few notes into it. Check that the number of notes returned by `numberOfNotes()` matches the number that you stored. When you use the `showNote()` method, you will need to use a parameter value of 0 to print the first note, 1 to print the second note, and so on. Try using the `removeNote()` method to remove a note. Then, use the `showNote()` method to print the notes in the list. *Note*: `showNote()` and `removeNote()` can be copied from the "notebook1" project.

   c. Modify `showNote()` and `removeNote()`to print out an error if the note number entered was invalid.

   d. Modify the `listNotes()` method so that it prints the value of the index local variable in front of each note. For instance:

      0: Buy some bread.
      1: Recharge phone.
      2: 11.30: Meeting with John.

   e. Within a single execution of the `listNotes()` method, the notes collection is asked repeatedly how many notes it is currently storing. This is done every time the loop condition is checked. Does the value returned by `size()` vary from one check to the next? If no, the execution speed can be increased by rewriting the `listNotes()` method so that the size of the notes collection is determined only once and stored in a local variable prior to the execution of the loop. Then use the local variable in the loop's condition rather than calling `size()`. Check that this version gives the same results.

   f. Change your notebook so that notes are numbered starting from 1 rather than 0. Remember that the `ArrayList` object will still be using indices starting from zero, but you can present the notes number from 1 in your listing. Make sure you modify `listNotes()` and `removeNote()` appropriately.

   g. Rewrite `listNotes()` so that its elements are accessed by using an `Iterator` object instead of using array indexes. State the advantage(s) of using `Iterator` instead of array indexing in accessing the array list.

4. The following program compares the run-time performance of `Iterator` and array indexing (using `ArrayList.get()`). Complete the program under the Eclipse IDE. If you encounter insufficient memory error, you can change the heap size of Eclipse. See http://wiki.eclipse.org/FAQ_How_do_I_increase_the_heap_size_available_to_Eclipse%3F for details. If you prefer to use command line, you may change the heap size by using the following options in a Command Window:

   ```
   cd <Folder storing RunTimeComparison.java>
   javac RunTimeComparison.java
   java  -Xms32m  -Xmx1024m RunTimeComparison
   ```

   Note that you cannot change the heap size in BlueJ.

   Hints: Use the `getTime()` method in the `Date` class to obtain the current time.

```
public class RunTimeComparison {
    public static void main(String args[]) {
        int N = 200000;
        // Reduce this value if you are impatient.

        long startTime;
        long stopTime;
        List<Integer> list;
        ArrayList<Integer> aList;
        LinkedList<Integer> lList;

        /* Create an array with N Integer objects */
        Integer array[] = new Integer[N];
        for (int i=0; i<N; i++) {
            array[i] = new Integer(i);
        }

        /* Convert the Integer array into a List object (list) using the
           asList() Method in the Arrays class */

        // Put your code here

        /* Convert the List object into an ArrayList object and an
           LinkedList object. Use the aList and lList references to refer
           to these objects */

        // Put your code here

        /* Estimate the run-time used by the Iterator to access all of the
           elements in ArrayList */

        // Put your code here
        System.out.println("Running Time using Iterator on ArrayList
                                   = "+(stopTime-startTime));

        /* Estimate the run-time used by array indexing (get()) to access
           all of the elements in ArrayList */

        // Put your code here
        System.out.println("Running Time using Indexing on ArrayList
                                  = "+(stopTime-startTime));
```

```
        /* Estimate the run-time used by the Iterator to access all of the
           elements in LinkedList */

        // Put your code here
        System.out.println("Running Time using Iterator on LinkedList
                                       = "+(stopTime-startTime));

        /* Estimate the run-time used by array indexing (get()) to access
           all of the elements in LinkedList */

        // Put your code here
        System.out.println("Running Time using Indexing on LinkedList
                                      = "+(stopTime-startTime));
    }
}
```

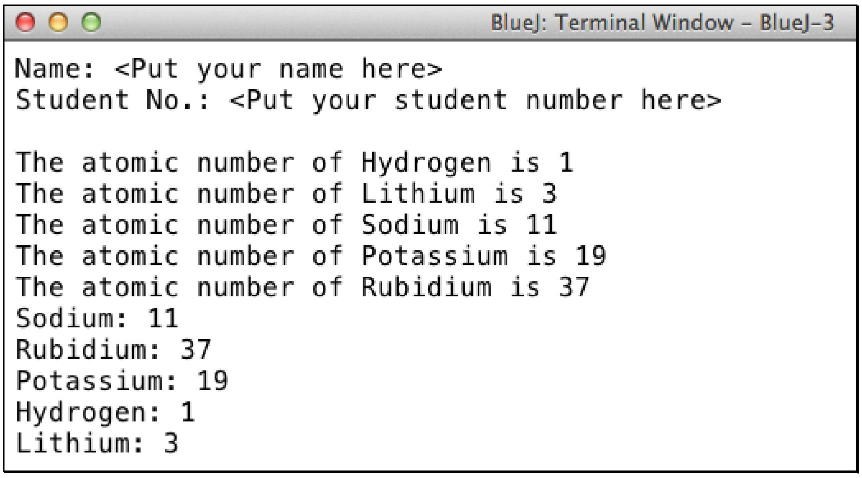
Capture the screen that contains the program output. Explain why it is advantageous to use `Iterator` instead of array indexing.

Note**:** You may need to find information of the `LinkedList` and `Arrays` classes from the Internet.

Hints: For beginners, the Java 1.4 API Documentation is easier to understand. But you can use Java 1.6 or later to compile the program.

5. The following program stores the atomic number of 5 elements in a `HashMap` object (Self-learning: search the Internet to learn this class). Complete the code in the constructor and the method `toString()` so that the program can produce the output shown in Fig. Q5 when it is executed. Capture the output of your program. Your screen capture should contain your name. Note that `HashMap` does not guarantee the order of elements in the data structure. Therefore, your program output may be different from Fig. Q5.

```
 // MyHashMap.java
 import java.util.*;
 public class MyHashMap<K,V> extends HashMap<K,V> {
   public MyHashMap(K[] keys, V[] values) {
         super();
              // Store the <key,value> pairs to the HashMap object
              // Put your code here
   }

   public String toString() {
          // Put your code here
   }

   public static void main(String[] args) {
         String[] elements =
                       {"Hydrogen","Lithium","Sodium","Potassium","Rubidium"};
         Integer[] atomicNumbers = {1,3,11,19,37};
         MyHashMap<String,Integer> map =
             new HyHashMap<String,Integer>(elements,atomicNumbers);
         for (int i=0; i<elements.length; i++) {
               System.out.println("The atomic number of " + elements[i] + "
          is "
                                        + map.get(elements[i]));
         }
         System.out.println(map.toString());
   }
}
```



```
Name: <Put your name here>
Student No.: <Put your student number here>

The atomic number of Hydrogen is 1
The atomic number of Lithium is 3
The atomic number of Sodium is 11
The atomic number of Potassium is 19
The atomic number of Rubidium is 37
Sodium: 11
Rubidium: 37
Potassium: 19
Hydrogen: 1
Lithium: 3
```

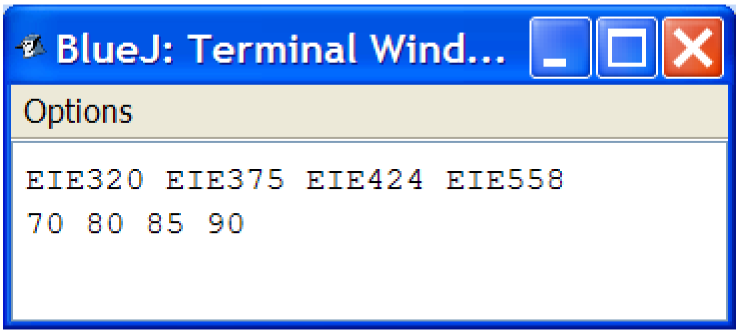
Fig. Q5

6. Create a class called `MyArrayList` by inheriting `java.util.ArrayList`. Add a constructor that accepts either a `String` array or an `Integer` array as input. The constructor should be able to add all of the elements in the input array to the `MyArrayList` object. Add a method called `print()` to display the contents of the `MyArrayList` object. Create a class called `TestMyArrayList` to test the constructor and the `print()` method in `MyArrayList`.

   // TestMyArrayList.java

```
import java.util.Collections;

public class TestMyArrayList
{
    public static void main(String args[])
    {
        String[] subjects = {"EIE320", "EIE558", "EIE375", "EIE424"};
        Integer[] marks = {90,85,70,80};
        MyArrayList<String> sList
            = new MyArrayList<String>(subjects);
        Collections.sort(sList);
        sList.print();
        MyArrayList<Integer> mList = new MyArrayList<Integer>(marks);
        Collections.sort(mList);
        mList.print();
    }
}
```

   The output of `TestMyArrayList` is shown below.



```
EIE320 EIE375 EIE424 EIE558
70 80 85 90
```

   Hints: You may need to use “Generic Data Type” in Java.

Lawrence Cheung
August 2017