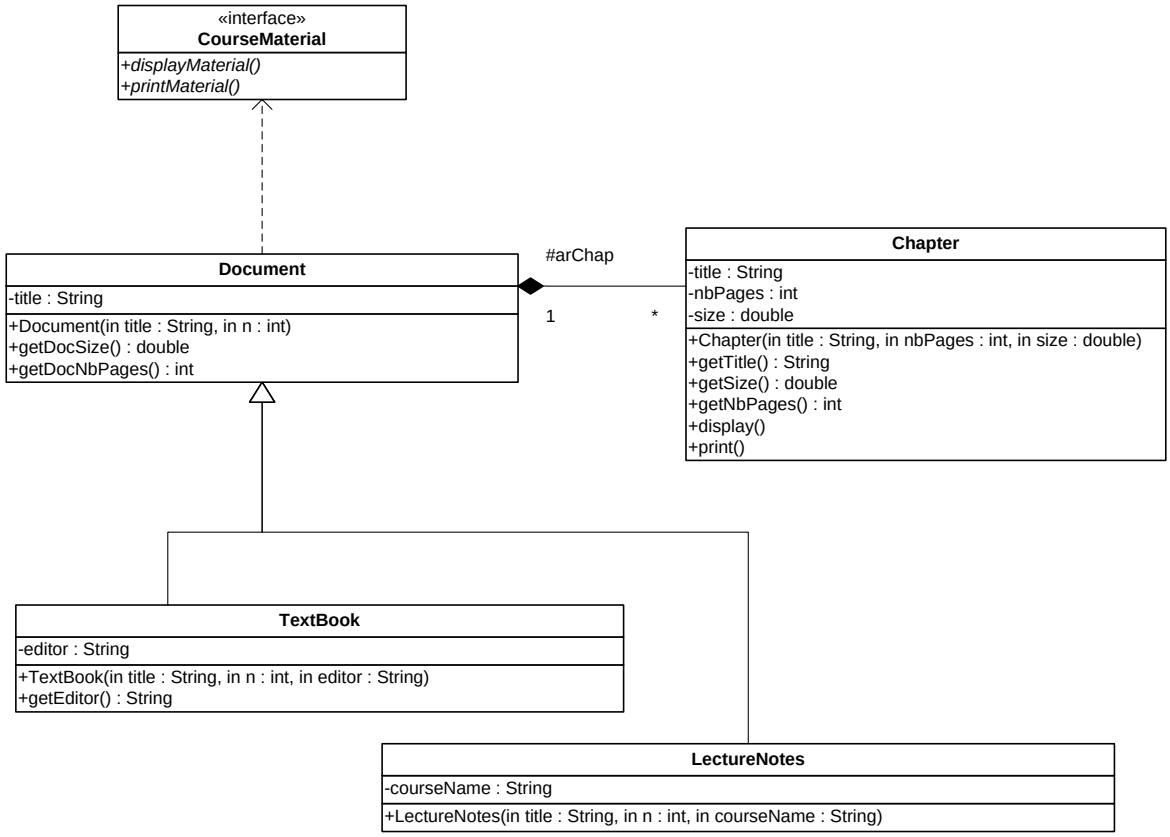
King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 – Computer Programming II – Midterm 2 Exam – Spring 2018

Exercise1:



Class ***Chapter :***

- Attributes:
  - ***title***: the title of the chapter.
  - ***nbPages:*** the number of pages of the chapter.
  - ***size:*** the size in kilo-bytes of the chapter.

- Methods:
  - ***Chapter(title:*** *String****, nbPages:*** *int****, size:*** *double****):*** constructor.
  - ***getTitle(), getSize()*** and ***getNbPages()*** : getters.
  - ***display():*** this method displays all attributes of the chapter.
  - ***print():*** this method prints the chapter.

Interface ***CourseMaterial :***

- Methods:
  - ***displayMaterial():*** this method displays all the attributes of the Document (TextBook or LectureNotes) as well as its chapters.
  - ***printMaterial():*** this method prints the chapters of the Document as follows:
    - For ***Text Books,*** it prints the chapter entitled "Introduction" **only**.
    - For ***Lecture Notes,*** it prints all chapters.

Class ***Document :***

- Attributes:
  - ***title***: the title of the document.
- Methods:
  - ***Document(title:*** *String****, n:*** *int****):*** constructor. The parameter ***n*** defines the maximum number of chapters of the Document.
  - ***getDocSize():*** this method returns the total size of all chapters of the Document.
  - ***getDocNbPages():*** this method returns the total number of pages of all chapters of the Document.

Class ***TextBook :***

- Attributes:
  - ***editor***: the editor name of the textbook.
- Methods:
  - ***TextBook(title:*** *String****, n:*** *int****, editor:*** *String****):*** constructor.
  - ***getEditor():*** getter.

QUESTION: Translate into Java code:

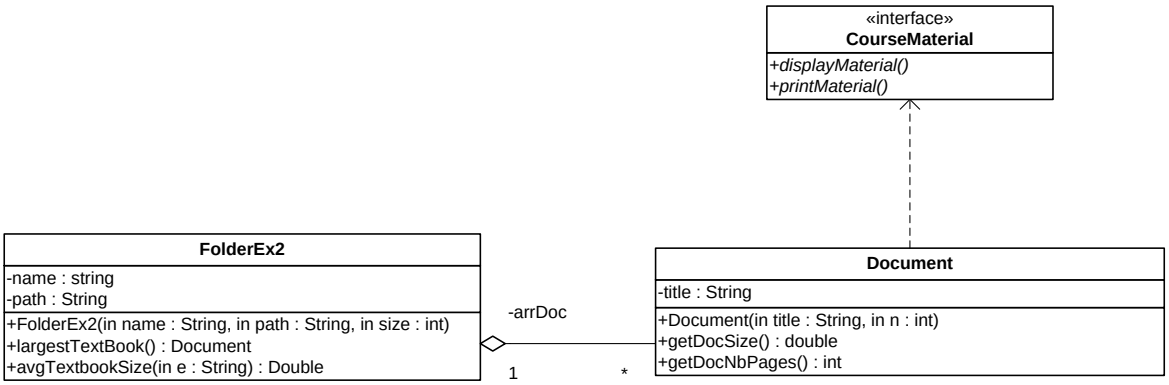
1. The interface ***CourseMaterial***,
2. the class ***Document*** and
3. the class ***TextBook***.

```
public interface CourseMaterial {   ……./1      ……./3
      public void displayMaterial(); ……./1
      public void printMaterial();   ……./1
}

public abstract class Document implements CourseMaterial { ……./1+1   ../17
      private String title;
      protected Chapter arChap[];    ……./1
      protected int nbChap;          ……./1

      public Document(String t,int n) {   ……./2
            title = t;  arChap = new Chapter[n]; nbChap = 0;
      }

      public void displayMaterial(){       ……./3
            System.out.println(title);           ……./1
            for (int i = 0; i < nbChap; i++)     ……./1
                  arChap[i].display();           ……./1
      }

      public double getDocSize() {   ……./4
            double totalSize = 0.0; ……./1
            for (int i = 0; i < nbChap; i++)     ……./1
                  totalSize += arChap[i].getSize();    ……./1

            return totalSize; ……./1
      }
      public int getDocNbPages() {   ……./4
            int totalNbP = 0; ……./1
            for (int i = 0; i < nbChap; i++) ……./1
                  totalNbP += arChap[i].getNbPages(); ……./1

            return totalNbP;  ……./1
      }
}

public class TextBook extends Document { ……./1      ……./10
      private String editor;
      public TextBook(String t,int n, String e) {      ……./2
            super(t, n); ……./1
            editor = e; ……./1
      }

      public void displayMaterial() {      ……./2
            super.displayMaterial();……./1
            System.out.println(editor);    ……./1
      }

      public void printMaterial() {  ……./4
            for (int i = 0; i < nbChap; i++) { ……./1
                  if (arChap[i].getTitle().equals("Introduction")) {     ……./1
                        arChap[i].print();       ……./1
                        return;                  ……./1
                  }
            }
      }
```

```
	public String getEditor() {	……./1
		return editor;
	}
}
```

Exercise 2:

Let's consider the same class ***Document*** described in exercise 1.



Class ***FolderEx2:***

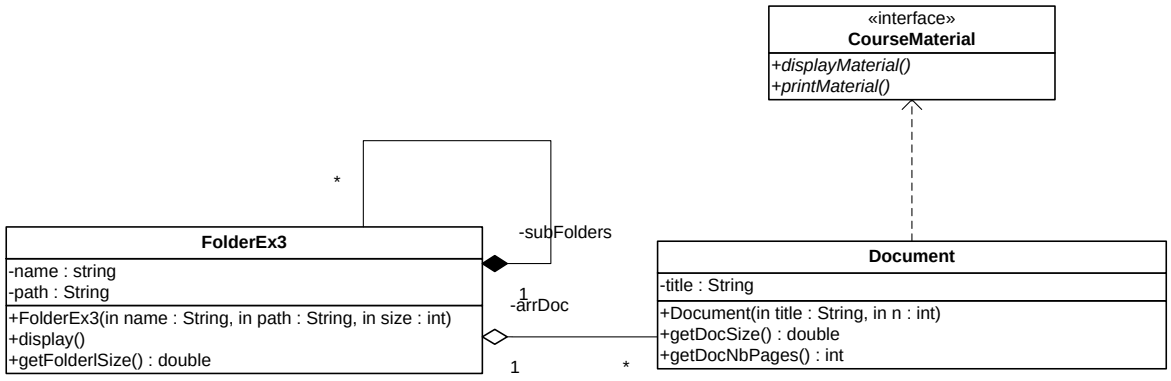
- Attributes:
  - ***name:*** the name of the folder.
  - ***path***: the path of the folder.
- Methods:
  - ***FolderEx2 (name:*** *String****, path:*** *String****, size:*** *int****)***: constructor. The parameter ***size*** defines the maximum number of documents in the folder.
  - ***largestTextBook():*** this method returns the Document of type TextBook having the largest total number of pages.
  - ***avgTextBookSize(e:*** String***):*** this method returns the average Document size of all TextBooks edited by the editor ***e***.

QUESTION: Translate into Java code the class ***FolderEx2.***

```
public class FolderEx2 {        ……./21

	private String name;
	private String path;

	private Document arDoc[];      ……./1
	private int nbDoc;             ……./1

	public FolderEx2(String s, String p, int size) {        ……./2
		name = s;
		path = p;

		arDoc = new Document[size];    ……./1
		nbDoc = 0;                     ……./1
	}

	public Document largestTextBook() {        ……./7
		Document res = null;     ……./1

		for (int i = 0; i < nbDoc; i++) {    ……./1
			if (arDoc[i] instanceof TextBook) { ……./1
				if (res == null  ||  ……./1
				    arDoc[i].getDocNbPages()>res.getDocNbPages())./1
					res = arDoc[i];    ……./1
			}
		}
		return res; ……./1
	}

	public double avgTextBookSize(String e) {        ……./11
		double total = 0.0;      ……./1
		int nb = 0;              ……./1

		for (int i = 0; i < nbDoc; i++) {    ……./1
			if (  arDoc[i] instanceof TextBook && ……./1
				((TextBook) arDoc[i]).getEditor().equals(e)) {……/1+1

				nb ++;       ……./1
				total += arDoc[i].getDocSize();      ……./1
			}
		}

		if (nb > 0) ……./1
			return total / nb;       ……./1
		else
			return 0.0; ……./1
	}
}
```

Exercise 3:

Let's consider the same class ***Document*** described in exercise 1.



Class ***FolderEx3:***

- Attributes:
  - ***name:*** the name of the folder.
  - ***path***: the path of the folder.
- Methods:
  - ***FolderEx3 (name:*** *String****, path:*** *String****, size:*** *int****)***: constructor. The parameter ***size*** defines the maximum number of documents in the folder.
  - ***display()*:** this method displays both; all documents and subfolders of the folder.
  - ***getFolderSize():*** this method returns the total size of the folder calculated as follows:

    Total size of the folder = $\sum size\_of\_Documents + \sum size\_of\_subfolders$

QUESTION: Translate into Java code the class ***FolderEx3.***

```
public class FolderEx3  {                    ……./19

      private String name;
      private String path;

      private Document arDoc[];              ……./1
      private int nbDoc;                     ……./1

      private FolderEx3 subfolders[];        ……./1
      private int nbSubFolders;              ……./1

      public FolderEx3(String s, String p, int size) {        ……./4
            name = s;
            path = p;

            arDoc = new Document[size];      ……./1
            nbDoc = 0;                       ……./1

            subfolders = new FolderEx3[10];  ……./1
            nbSubFolders = 0;                ……./1
      }

      public void display() {                ……./5
            Document res = null;             ……./1

            for (int i = 0; i < nbDoc; i++) {    ……./1
                  arDoc[i].displayMaterial();    ……./1
            }

            for (int i = 0; i < nbSubFolders; i++) {   ……./1
                  subfolders[i].display();             ……./1
            }
      }

      public double getFolderSize() {        ……./6
            double totalS = 0.0;             ……./1

            for (int i = 0; i < nbDoc; i++) {    ……./1
                  totalS += arDoc[i].getDocSize();    ……./1
            }

            for (int i = 0; i < nbSubFolders; i++) {   ……./1
                  totalS += subfolders[i].getFolderSize();   ……./1
            }

            return totalS;                   ……./1
      }
```