Dear students

Below are some important regulations concerning the forthcoming exam. Please read it carefully as there are many details that you should follow.

1. The final examination on course **Introduction to programming I** will take place on the 7th of December at 11:00.

2. The exam is paper based and will be organized in the **offline mode**. However, **it would be required** for online students to attend to the Zoom meeting (link to be shared on Moodle and via Telegram *some minutes before the final*).

3. For the Zoom meeting, you should **turn on** your camera and microphone. You should be the only person in the room. Your background should be visible. Your desk should be clean, without any electronic nor lecture notes. Charge your computer in advance and install Zoom, if you haven’t done it before.

4. The exam will last **40 minutes**.

5. You will be given **two tasks**: one in C and one in Java. The tasks assume writing some small portion of C/Java code (that is not necessary to debug).

6. In Moodle you will see the assignment named "**Final Exam**" (it will become visible just on the start of the exam). It contains the PDF file with tasks.

7. You should read the task descriptions and write your answers on **PAPER** (and paper only, not in any electronic format). So, prepare a couple of sheets of paper and a pen. All the sheet should be signed.

8. You will not be allowed to go to the restroom during the exam. Go before you start. Professor and TAs will be watching you via Zoom or personally (in case of physical presence in 105).

9. Online students at the end of the exam, **take photos** of your **desk** and of your **sheets** with answers, and **upload** these file(s) (no compress is necessary) to **Moodle**, to the "**Final Exam**" assignment. This should be done by online students **as well as** by students taking exam in room 105. After uploading files to **Moodle**, offline students should submit signed hard copies before leaving the room.

If you have any questions, please ask in advance

Task 1.

Complete the code in C: program that swaps 2 integers

**Version 1:**

**void** swap( )

{

}

**int** main()

{

**int** a, b;

**printf**("Enter first integer ");

**scanf**("%d", &a);

**printf**("\nEnter second integer ");

**scanf**("%d", &b);

*// Write missing arguments*

    swap( );

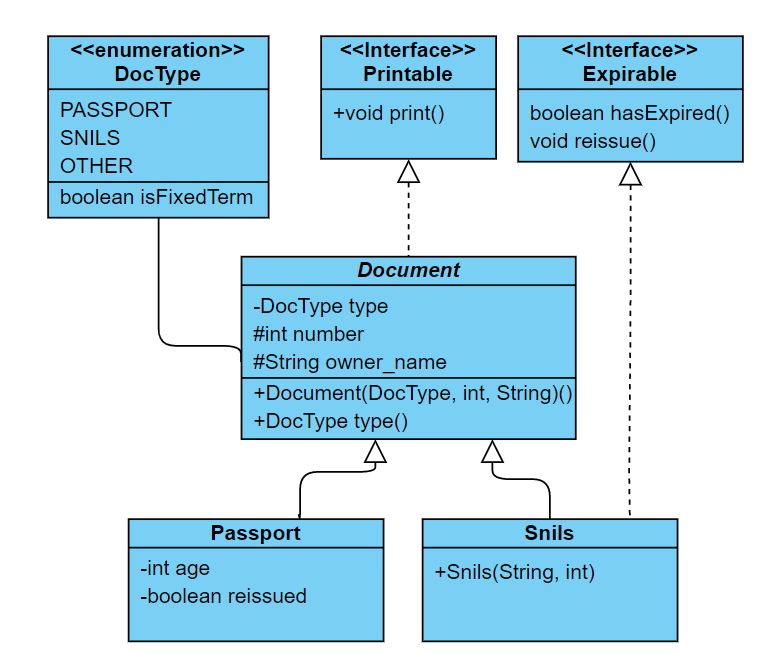
**printf**("\nAfter Swapping: a = %d, b = %d", a, b);

**return** 0;

}

Task 2

Consider the following UML diagram



- private

+ public

# protected

Methods are intentionally omitted

Part I. Interfaces

Write the code for interfaces: ***Printable*** and ***Expirable*** according to UML diagram above

Part II. Enums

Complete the implementation of the enum ***DocType*** according to UML diagram above

public enum DocType {  
 *PASSPORT* (true), *SNILS* (false), *OTHER*(false);

}

Part III: Classes

Provide an implementation for an abstract class ***Document*** according to UML diagram above

Complete ***Passport*** class declaration, constructor and the methods

public class Passport {  
 private int age;  
 private boolean reissued;  
  
 public Passport(String name, int age, int number) {

}  
  
 @Override  
 public void print() {  
 System.*out*.println(type() + " #" + number + " of "+owner\_name);  
 System.*out*.println(" Status: " + (hasExpired()?"expired":"valid"));  
 }  
  
 @Override  
 public boolean hasExpired() {  
 *// False if age is older than 44 and has not reissued*

}

}

Complete ***Snils*** class declaration, constructor

public class Snils {  
  
 public Snils(String name, int number) {

}

@Override  
 public void print() {  
 System.*out*.println(type() + " #" + number + " of "+owner\_name);  
 }  
}

Part IV: Runtime

Complete the implementation of ***Main***

public class Main {  
  
 public static void main(String[] args) {  
 Document document = new Passport("Alice",18, 11133);  
 Passport passport = new Passport("Bob", 50, 33333);  
 Snils snils = new Snils("Carl",7777);  
  
 */\* Initialize a list and add* ***document, passport*** *and* ***snils*** *to it.  
 \* Iterate through elements of the list:  
 \* output a message "Document #" + number + " has expired"  
 \* if document has expired, or call print() on the element otherwise  
 \* Make sure to handle exceptions (if any thrown) \*/*

}  
}