Full name :_____

Group :_____

Exam regulations:

- The exam is a closed book: you cannot use any electronic devices or written materials.
- After completing all the tasks or when 40 minutes are over, you must leave the room, keeping the exam sheets on your desk face down.
- Follow the task description and comments to complete the program. Write your code in the blank A4 sheets you have prepared. Provide only missing parts of the code, mark your code snippets: Task1 a) … b)… etc.
- Code you are writing should be correct, readable, and according to the language syntax (C11 and Java8), however you might omit nuances like *import*s and *annotations*.
- In case of any concerns rise your hand in zoom, and proctor will refer to you.
- After completing the tasks or when the time is up, **take photos** of your **desk** and your **sheets** with answers, and **upload** these file(s) (no compress is necessary) to **Moodle** to the "**Final Exam**" assignment. Submission in Moodle to be made within 5 minutes; you might get a penalty for a late submission otherwise.
- After uploading files to **Moodle**, you will be allowed to leave the Zoom meeting.

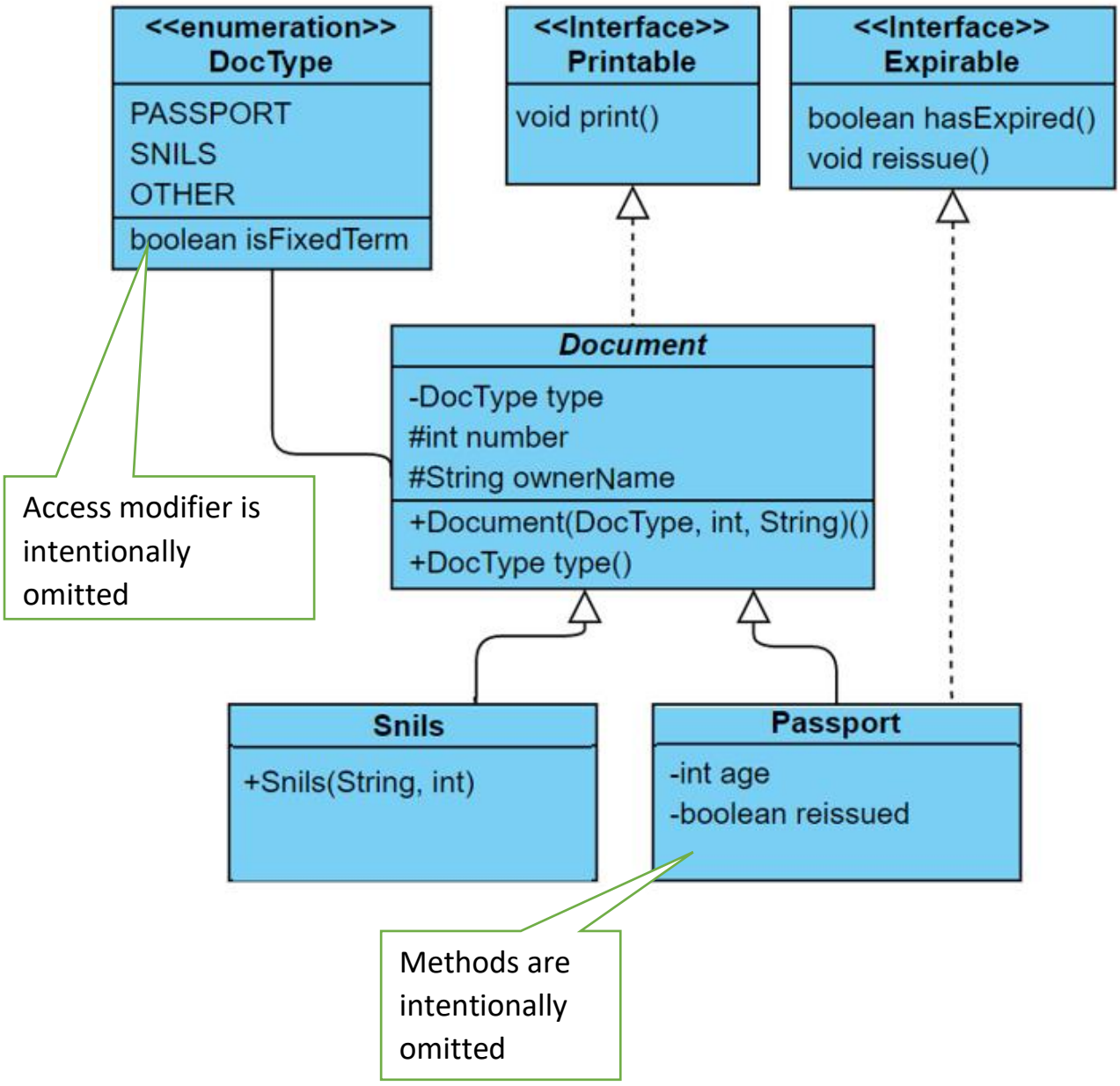
Task 1. (7 points)

Complete the code in C: program that swaps 2 integers

Main.c

```
#include <stdio.h>

a)

int main()
{
    int a, b;
    printf("Enter first integer ");
    scanf("%d", &a);
    printf("Enter second integer ");
    scanf("%d", &b);
    // Write missing arguments
b)  swap(       );
    printf("After Swapping: a = %d, b = %d\n", a, b);
    return 0;
}

c)  void swap(                  )
{




}
```

Task 2 (23 points)
Consider the following UML diagram



Full name :_____

Group :_____

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

a)

Full name :_____

Group :_____

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

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

c)

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

d)

```
}
```

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

e)
```
public class Snils_____{
    public Snils(String name, int number) {
```

f)

```
    }
    @Override
    public void print() {
        System.out.println(type() + " #" + number + " of " + ownerName);
    }
}
```

Part IV: Runtime

Complete the implementation of ***Main*** and provide the expected output of the program

```
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) */
```

a)

```
    }
}
```

Output of the program:

b)