UNIVERSITATEA BABEŞ-BOLYAI

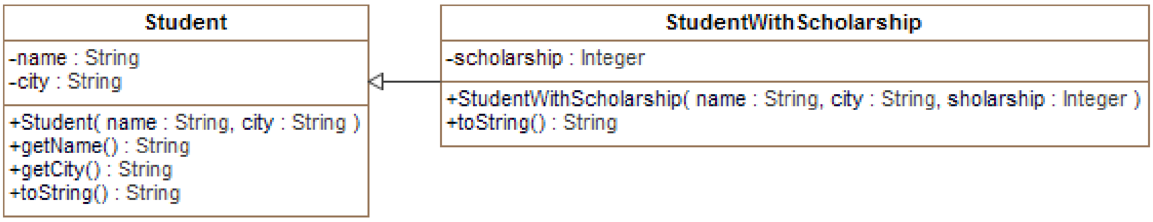
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

Bachelor Degree Exam, June 2016
Computer Science - English

Subject 1

Write a program in one of the programming languages Python, C++, Java, C# that:

(a) **Defines the classes** Student and StudentWithScholarship according to the following UML class diagram:



- ◦ The name and city must be not null and non empty, and the scholarship must be greater than zero. The constructors must enforce the constraints.
- ◦ The toString() method from the class Student returns the name concatenated with the city of the student, while the method toString() from class StudentWithScholarship appends the scholarship to the string containing the name and the city.

(b) **Defines a function** that inserts an object of type Student into a list of students ordered ascending by name, such that the list remains ordered.

(c) **Defines a function** that prints a list of students.

(d) **Defines a function** that has as parameter a list of students, and prints for each city the list of students with scholarship (ordered by name) from that city. Each city is printed only once, in any order.

(e) The **main function** of the program calls the function indicated at (b) to build a list of four students, from two cities “Arad” and“Oradea”, three of them having scholarships. Then it prints the list of students using the function defined at (c). Finally, it calls the function defined at (d).

(f) For the data types used in the program, write the specifications of the used operations.

Remarks

- **Do not use sorted containers and sorting operations.**
- **Do not define other methods than those required in the subject.**

You can use existing libraries for data structures (Python, C++, Java, C#).

Subject 2

a. Create a relational database with all the tables in 3NF that will store the following information for an online booking system:

tourists: name, email, city code, city name, country code, country name;

accommodation establishments: name, city code, city name, country code, country name, accommodation type code, accommodation type name (examples of values: pension, hostel, hotel, etc), number of stars, rating, price / night, and a list of **bookings**, for each such booking knowing the tourist, starting date and number of nights.

Prove that the created database is in 3NF, by identifying the functional dependencies.

b. For the database created for **a**, solve, using the relational algebra or SQL, the following queries:

i) The tourists (name and email) that booked at least a pension whose rating is greater than 9, but didn't book any 3 star hotel with a rating less than 9.

ii) The total number of bookings for all the tourists from Cluj in a 5 star hotel in Paris in 2015.

iii) The accommodation establishment (name, city name, accommodation type name, number of stars) that has the greatest number of bookings over the last 5 years.

Subject 3

3.1 The file **grep.c** contains the code fragment below and is compiled in the user's home directory under the name **grep**. Answer the following questions considering that all instructions are executed successfully.

```
int main(int c, char** v) {
  int p[2], n;
  char s[10] = "something";
  pipe(p);
  n = fork();
  if(n == 0) {
    close(p[0]);
    printf("before\n");
    if(c > 2)
      execlp("grep","grep",v[1],v[2],NULL);
    strcpy(s, "after");
    write(p[1], s, 6);
    close(p[1]);
    exit(0);
  }
  close(p[1]);
  read(p[0], s, 6);
  close(p[0]);
  printf("%s\n", s);
  return 0;
}
```

a) Enumerate and explain the possible values of variable **n**.

b) What will display the following executions on the screen, considering that the user's home directory does not appear in the PATH environment variable?

b.1) `grep grep grep.c`

b.2) `./grep grep grep.c`

b.3) `./grep grep`

3.2 File **abc.sh** contains the UNIX Shell script below. Answer the following questions.

```
1   n=0
2   for i in `cat $1`; do
3     c=`echo $i|cut -c1`
4     if echo $i|grep -q "^[0-9][0-9]*$"; then
5       echo $i >> $1.nr
6     elif echo $c|grep -q "[A-Za-z]"; then
7       echo $i >> $c
8     else
9       n=`expr $n + 1`
10    fi
11  done
12  echo $n
```

a) Explain the regular expression on line 4

b) What will happen if the script is executed without arguments?

c) What will print the execution `./abc.sh f3` on the screen and what files (name and content) will it create, if f3 contains "`abc 74 2-8 aa 3a =c b2`" and the execution is done in a directory containing only the files abc.sh and f3?

d) Give an example of file f3 so that the execution at the point above creates 4 new files, none of which has the prefix f3 in their name.

Remarks: All subjects are compulsory. Each subject will be graded with a mark between 1 and 10 by both evaluators.

Time limit: 3 hours

BAREM

INFORMATICĂ

Subiect 1 (Algoritmică şi Programare):

Oficiu – **1**p

Definirea clasei Student– **0.75**p din care

- atribute – 0.25
- constructor – 0.25
- metode - 0.25

Definirea clasei StudentBursier– **1.25**p din care

- relația de moștenire – 0.25
- constructor – 0.5
- metoda toString() – 0.5

Funcția de la punctul b) – **2**p din care

- signatura corectă - 0.1p
- algoritmul de inserare în interiorul listei - 1.8p
  - parcurgere listă și determinarea poziției de inserare – 1.6p
  - adăugare element pe poziția determinată anterior – 0.2
- returnare rezultat - 0.1p

Funcția de la punctul c) - **0.5**p din care

- signatura corectă - 0.1p
- parcurgere listă și afișare – 0.4p

Funcția de la punctul d) – **2.5**p din care

- determinare lista bursieri din orașe – 2p
- tipărire perechi – 0.5p

Funcția de la punctul e) – **0.5p**

Specificațiile operațiilor tipurilor de dată folosite– **1.5**p

Subiect 2 (Baze de date)

1 punct oficiu

Problema a:

- **1 punct** pentru dependențe funcționale
- **2 punct** pentru tabelele în 3NF;
- **1 punct** pentru justificare.

Problema b:

- **1 puncte** pentru b1
- **1.5 puncte** pentru b2
- **2.5 puncte** pentru b3

Subiect 3 (Sisteme de operare):

Oficiu: 1p

3.1

- a) 1p - Valorile posibile
  1p - Explicații
- b.1) 1p - Afișează linia 10
- b.2) 1p - Afișează "înainte", linia 10 și "ceva"
- b.3) 1p - Afișează "înainte" și "după"

3.2

- a) 1p - Secvență una sau mai multe cifre
- b) 0.5p - Comanda "cat" așteaptă input de la intrarea standard

c) 0.5p - Afișează 3 pe ecran
   0.5p - Fișierul f3.nr conținând 74
   0.5p - Fișierele dicționar a și b, conținând “abc aa” respectiv “b2”
d) 1p - Orice secvență de cuvinte dintre care 4 încep cu litere diferite