Bachelor Degree Exam, June 2013
Computer Science - English

Subject 1

Write a program in C++, Java, or C# which:

  a. Define a class **Product** having a private field *name* of type string, a public accessor method for this field, a public constructor which initializes the name field, and a public method for printing the *name* field on the standard output.

  b. Define a classs **AlterableProduct** derived from **Product** having a private field *validity* of type integer, a public constructor for initializing the name and validity fields, and a public method for printing both fields by reusing the printing method defined by the base class.

  c. Build into the main program a list containing the following products *ordered by name*: a **Product** having the name ″Bulb″; an **AlterableProduct** having the name ″Cheese″ and the *validity* equals to 60; a **Product** having the name ″Mug″. Then, the program reads a name from the standard input and performs a sequential search in the above list for printing the product having that name (the search algorithm will take into account that the list is ordered).

  d. For the list type used in the program, write the specifications of the operations called by the program.

You can use existing libraries for data structures (C++, Java, C#). In case you do not use existing libraries you do not need to implement the list operations.

Subject 2

a. Identify the functional dependencies for the following information about an electronic library:

  • **documents**: document code, document type (journal article, book, research report, web document etc), list of authors (a document may have 0, 1, 2, ... authors), title, publishing year;
  • **readers**: reader code, name, list of examined documents;

Create a relational database, having all tables in 3NF, for the above information. **Justify** that the resulting tables are in 3NF.

b. Given the database created at point **a**, express the following queries using relational algebra **or** Select-SQL:

  b1. Code and name of readers that examined at least one "*research report* " and **did not** examined any "*book*".
  b2. Code and name of readers that examined documents with at least 5 authors.

Subject 3

a. The following code fragments are executed concurrently on the same computer. Consider that:
- the FIFO files `a2b` and `b2a` are already created and empty
- all instructions are executed without error
- the FIFO operations are not affected by `O_NDELAY`

Answer the following questions:
1. What will be displayed in the console of program P1? Enumerate all possibilities.
2. What happens if the order of the `open` instructions in program P2 is inverted?
3. How many processes are created by program P1 (excluding the parent process)?
4. How many processes are created by program P1 (excluding the parent process) if the instruction `exit(0)` is missing?

P1	P2
| ```c
int main() {
    int i, n=1;
    int a2b = open("a2b", O_WRONLY);
    int b2a = open("b2a", O_RDONLY);

    for(i=0; i<3; i++) {
        if(fork() == 0) {
            n += i;
            write(a2b, &n, sizeof(int));
            read(b2a, &n, sizeof(int));
            printf("%d: %d\n", getpid(),
n);
            exit(0);
        }
    }

    wait(0); wait(0); wait(0);
    close(a2b); close(b2a);
    return 0;
}
``` | ```c
int main() {
    int i, n;
    int a2b = open("a2b",
O_RDONLY);
    int b2a = open("b2a",
O_WRONLY);

    for(i=0; i<3; i++) {
        read(a2b, &n, sizeof(int));
        n++;
        write(b2a, &n,
sizeof(int));
    }

    close(a2b); close(b2a);
    return 0;
}
``` |

b. Consider the UNIX shell script fragment below.
1. Explain the functionality of line 2.
2. Explain the functionality of line 3.
3. What will display each of the executions below?

Line	Script a.sh	Executions
1	`SUM=0`	E1: `./a.sh 1 2 3 4`
2	`for A in $*; do`	
3	`    N=`echo $A |grep "^[0-9]$"``	E2: `./a.sh a b1 c d2 5`
4	`    if [ "$N" != "" ]; then`	E3: `./a.sh 1 20 300`
5	`        SUM=`expr $SUM + $N``	
6	`    fi`	E4: `./a.sh`
7	`done`	
8	`echo $SUM`	

Remarks: All subjects are compulsory. Each subject will be graded with an integer mark between 1 and 10 by both evaluators.
             Time limit: 2 hours