

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION
DURATION: 3 HOURS

WINTER SEMESTER, 2022-2023
FULL MARKS: 150

CSE 4301: Object Oriented Programming

Programmable calculators are not allowed. Do not write anything on the question paper.

Answer all **6 (six)** questions. Figures in the right margin indicate full marks of questions whereas corresponding CO and PO are written within parentheses.

1. Assume that you are asked to develop a portion of a program called Course Management System for the CSE Department of IUT that considers the following requirements.

30

(CO4)

(PO3)

Teachers are assigned to different courses in an academic semester. A course has attributes like course title, course code, effective contact hour, number of sections/groups, etc. Here, effective contact hours indicate the amount of contact hours counted for a teacher in his teaching load if the course is assigned to the teacher for one section. There are three categories of courses: Theory, Lab, and Thesis/Project. The number of teachers assigned to a course is based on this category. For a theory and thesis/project course, a section is assigned to a single teacher. For a lab course, a section is assigned to two teachers. For analyzing course related requirements in the future, unique functionalities may be added for each category of courses.

A teacher has attributes like Name, Designation, Minimum Contact Hours, etc. The sum of the effective contact hours of assigned courses of each teacher (also known as load) should be equal to or greater than the minimum contact hours.

You will need a separate class that deals with the assignment of courses to teachers. This class needs to provide the following functionalities by creating one or more public methods:

- Assign a course object to a teacher object. It should be noted that different teacher objects can be assigned to a single course. For example, assume that a theory course, CSE 4301 has 2 sections. Two teachers can be assigned to this course, each of whom is assigned to one section. You can assign one teacher to both sections as well. You cannot assign a course to a teacher if all sections of that course are already assigned.
- Detach a course object from a teacher object. It should remove the assignment of the teacher to that course. Partial detachment is possible, meaning not all sections of a course will be detached from that teacher.
- Display all the teachers whose load is lower than the minimum contact hour.
- Display all the courses that are not fully assigned, indicating the number of sections yet to be assigned.
- Display the current assignment of courses to teachers.

Identify different classes from the scenario and write their definition. You can add necessary member functions and variables that are not explicitly mentioned in the scenario. Consider adding the `const` modifier to any member function if necessary.

2. Banks have many different types of accounts, often with different rules for fees associated with minimum balance, withdrawals, etc. Customers are allowed to perform three categories of transactions on their accounts: deposit, withdraw funds, and transfer funds between accounts incurring the appropriate fees associated with the withdrawal of funds from one account. It should be noted that no transaction with a negative amount is allowed and the balance of any account cannot be negative at any moment. If the balance is negative due to application of fees by the Bank, then the account is frozen. No transaction is allowed on a frozen account. A transfer transaction from the first account to the second account will count as a withdrawal from the first account and deposit at the second account while calculating any fees for an account (although it will be a transfer transaction).

Banks have two categories/types of accounts. They are current account and savings account. In case of a current account, the account holder can deposit and withdraw money without any restriction. These accounts will have a yearly account maintenance fee of BDT 1000.

On the other hand, there is a daily limit on the number of free withdrawals and the number of free deposits for each savings account. It will be different for each savings account and initialized when the account is created (the default value for each is 5). There is also a daily limit on the amount of withdrawal (no limit on deposit). It will be different for each savings account and initialized when the account is created (the default value is BDT 10,000). If the daily limit is crossed, then a fee of BDT 100 or 2% of the transacted amount (whichever is higher) will be applied.

All categories of accounts have an account holder(s) name, an account number, and a list of transactions associated with that account. One can perform functions like deposit, withdrawal, and transfer of money on any account following the policy mentioned above. It should be noted that transaction history associated with the account will not be removed from the bank even if the account is removed.

A bank has a name, a list of bank accounts, and a list of transactions. The functionalities of a bank are clearing the daily limit on all accounts (it will reset the count of daily transactions to zero for all accounts of that bank), printing all transaction histories of that bank, and printing the current balance of all bank accounts. It should be noted that all transactions of a bank will be removed if that bank is removed. All the transactions should be executed using the bank object.

Design a program for the above-mentioned scenario using object-oriented features such as encapsulation, abstraction, and polymorphism. Use an STL container to store a list.

- 3. a) What is friend class and friend function? What are the pitfalls of using the friend function or friend class in your program?
- b) Should one declare the destructor function as virtual if s/he plans to use the class as an abstraction for other classes? Explain your opinion with an appropriate example.
- c) Can a single catch block handle multiple exception class objects? If so, give an example code to handle multiple exceptions under one try block.
- d) Give an example code demonstrating the single accessor function of a member variable, which can both read and write.
- e) What is the necessity of a pure virtual function compared to a virtual function? Explain it with an example code.
- f) What is the use of namespace? How can different identifiers (variables, functions, or classes) be present in different files be under the same namespace? For example: `cout` object, `min` function, `string` class, etc. are defined under different classes in different files but present in the `std` namespace.

4. Complete the definition of the classes according to the instructions given as comments in the Code Snippet 1. You cannot use any library other than `iostream`. The output of the given `main()` is (A: 2), (B: 1).

```
1 #include <iostream>
2 using namespace std;
3 class element {
4     char key; int count;
5 public: /*Implement appropriate constructors, getter and setters*/
6 };
7 class dictionary {
8     /*Declare an object pointer of 'element' class and *other properties
9     as necessary*/
10 public: /*Create necessary constructors*/
11     /*Check if element with the given key exists*/
12     bool hasElement(char key);
13     /* Add a new element to the array of 'element' objects. Return false
14     * if element already exists. Else return true.*/
15     bool addElement(element elm);
16     /* Add a new element to the array of 'element' objects initialized
17     key, count.
18     * Return false if element already exists. Else return true.*/
19     bool addElement(char key, int count);
20     /* Remove element from array of 'element' objects. Return false if no
21     * such element exists. Else return true.*/
22     bool removeElement(element elm);
23     /* Remove element from array of "element" objects by matching the
24     * key. Return false if no such element exists. Else return true.*/
25     bool removeElement(char key);
26     /* Increase count of the element that matches the key. Return false
27     * if no such element exists. Else return true.*/
28     bool increaseCount(char key);
29     /* Decrease count of the element that matches the key. Return false
30     * if no such element exists. Else return true. If element's count
31     * decreases to zero, remove the particular element*/
32     bool decreaseCount(char key);
33     /*Print all existing elements along with their count.*/
34     void printElements();
35 };
36
37 int main()
38 {
39     char addingList[] = "ABACB";
40     char deleteList[] = "BCCGG";
41     dictionary dict;
42
43     for (int i=0;i<5;i++)
44         if (!dict.addElement (addingList[i],1))
45             dict.increaseCount (addingList[i]);
46     for (int i=0;i<5;i++)
47         dict.decreaseCount (deleteList[i]);
48     dict.printElements();
49 }
```

Code Snippet 1: An incomplete C++ program for Question 4.