Military Institute of Science and Technology
B.Sc. in Computer Science and Engineering
Online-1, Spring 2022
Subject: CSE-206, Object Oriented Programming Language Sessional

Time: 50 mins **Full Marks: 20**

ID	Name	Question-1 (15)	Question-2 (5)

Question-1

Consider a small money-transferring system that is implemented using C++. The system consists of two classes, which are: ***User*** and ***Account***: All the attributes in both of the classes are **private**, on the other hand all the methods in both of the classes are **public**. Descriptions of the attributes are illustrated in *Table-1*, the constructors for both of the classes are described in *Table-2.* Description of the methods are illustrated in *Table-3.*

Class	Attribute	Description
User	char *id	Stores a string as the identification number of an user
	char *name	Stores a string as the name of an user
Account	User *user	Stores the information of an user as the account holder
	char *phone	Stores a string as the associated phone number of an account
	int *balance	Stores a single integer as the balance of an account

Table-1

Class	Constructor	Description
User	User(char*, char*);	Initializes ***user->id*** with the first parameter and ***user->name*** with the second parameter
Account	Account(char*, char*, char*);	Initializes ***user*** with the first two parameters and the ***phone*** with the third parameter. Sets the value of ***balance*** to 500.

Table-2

Class	Method	Description
User	~User( );	Deallocates the allocated memory
Account	void setId(char *x);	Sets the value of ***user->id*** to ***x***
	void setName(char *x);	Sets the value of ***user->name*** to ***x***
	void setPhone(char *x);	Sets the value of ***phone*** to ***x***
	void print( );	Prints the information of an account. For example: Let the ***user->id***, ***user->name***, ***phone***, ***balance*** of an **Account** are respectively (“S-11”, “Zarif”, “01211997711”, 200) then the method will print these information in the following format: ID: S-11, Name: Zarif, Phone: 01211997711, Balance: 200
	~Account( );	Deallocates the allocated memory

Table-3

In the global scope create an array of objects of **Account** class of size 4 named ***acc***. Initialize the objects of **Account** with the following values:
{("10", "Rafi", "010"), ("15", "Binita", "015"), ("29", "Nabil", "029"), (“36", "Maisha", "036")};

There is also a global function named ***transferMoney*** that takes three integers (***i***, ***j***, ***amount***) as parameters. It represents that ***acc***[***i***] is the sender and ***acc***[***j***] is the receiver. If the ***balance*** of sender is greater or equal to the ***amount*** then that ***amount*** is reduced from the ***balance*** of sender and added to the ***balance*** of receiver and prints a message “Transaction successful” otherwise prints “Insufficient balance”. Also if the ***phone*** of both **Account**s are the same then the function does nothing which means the transaction has failed and prints a message “Invalid transaction”.

Special Instruction

- Both of the classes do not have any other constructors except the constructors described in *Table-2*. But the copy constructors may be overwritten if required.
- It is allowed to define any function or class as friend of any class
- Implementing any other additional methods for any class is not allowed
- Using the header <bits/stdc++.h> is prohibited

Tasks

- Implement the two classes and the global function
- Create an **Account** in the *main* function named ***admin*** which will have the same set of values as ***acc***[0]. Note that ***acc***[0] and ***admin*** are two different entities though their attributes are being initialized with the same set of values. Later the ***admin*** may update its attributes but that will not update anything in ***acc***[0].
- Update the attributes of admin as the following
  user->id: "A-11"
  user->name: "Admin"
  phone: "01511"
- Do the following operations:
  transfer(2, 0, 100);
  transfer(1, 0, 150);
  transfer(1, 1, 100);
  transfer(2, 3, 800);
- Now print all the elements of the ***acc*** array and then print the ***admin*** also. Your output should look like as the following:
  Transaction successful
  Transaction successful
  Invalid transaction
  Insufficient balance
  ID: 10, Name: Rafi, Phone: 010, Balance: 750
  ID: 15, Name: Binita, Phone: 015, Balance: 350
  ID: 29, Name: Nabil, Phone: 029, Balance: 400
  ID: 36, Name: Maisha, Phone: 036, Balance: 500
  ID: A-11, Name: Admin, Phone: 01511, Balance: 500

Question - 2

"Even 5-month-old infants can calculate the results of simple arithmetical operations on small numbers of items. This indicates that infants possess true numerical concepts, and suggests that humans are innately endowed with arithmetical abilities."

[Karen Wynn, Addition and subtraction by human infants, Nature 1992]

In this problem, all you have to do is the job of an infant; ADDITION: a trivial arithmetic operation. Design a class named **Operation**, which can handle two data values, where the values can be of any data types. Next, write a Member function named Addition. The function shall be able to add the input values (Lets say C1 and C2) and output the resultant value, depending on the data types specified by the user.

The input of the program starts with an integer, indicating the number of test cases. Then, the program will take 2 inputs; first indicates the data type of the resultant value, while the second indicates the data type of the input values (C1 and C2).

Sample Input	**Sample Output**
3 int float 3 3.25 float int 3.65 3.78 string string NO FIRE	Test Case 1: 6, Test Case 2: 6, Test Case 3: NOFIRE