National University

O f C o m p u t e r & E m e r g i n g S c i e n c e s F a i s a l a b a d - C h i n i o t C a m p u s

CL-217
Object Oriented Programming

Lab No 8

Objectives:

- Classes and objects
- Data members
- Member function
- Data encapsulation
- Member access specifier (private , public , protected)
- Constructor, Destructor
- Copy Constructor
- This pointer
- Constant data members
- Inheritance

Note: Carefully read the following instructions (*Each instruction contains a weightage***)**

1. There must be a block of comments at start of every question's code by students; the block should contain brief description about functionality of code.
2. Comment on every function about its functionality.
3. Use understandable name of variables.
4. Proper indentation of code is essential
5. Write a C++ statement(s) for each of the following task one after the other, in the same order.
6. Make a Microsoft Word file and paste all of your C++ code with all possible screenshots of every **task outputs in MS word and do not submit .cpp file with word file**.
7. First think about statement problems and then write/draw your logic on copy.
8. After copy pencil work, code the problem statement on MS Studio C++ compiler.
9. At the end when you done your tasks, attached C++ created files in MS word file and make your submission on Microsoft teams. (Make sure your submission is completed).
10. Please submit your file in this format 19F1234_L7.
11. Do not submit your assignment after deadline.
12. Do not copy code from any source otherwise you will be penalized with negative marks.

Problem 1: | : (Classes, objects, Constructor, Destructor and Member functions, constant)

Write a class Employee with following data members and methods;

Private:

1. ID //A string to hold the ID of employee
2. Name //A string to hold the name of employee
3. Department //A string to hold the department name of employee
4. Bank Account number //A string to hold the organizational bank account number of employee (A read-only data member i.e. constant)
5. Grade //A string to hold the employee grade (1 to 8, 8 being the highest) of employee

Public:

1. **Default Constructor** should initialize all member variables with null value.
2. **Overloaded** Constructor should initialize all (initialize-able) data members of the employee class
3. **inputData** Input from user for all data members
4. **displayData** method that display data using (a const member function)
5. **Destructor**

Now dynamically create 3 other objects and initialize them with user input data. Show your results on console.

Explicitly call **destructors** for all **local** objects with proper message.

Problem 2: | (Classes, objects, Constructor, Destructor and Member functions, constant)

Write a class Person, having following private data members:

1. Name
2. Data of birth (a constant data member)
3. Count (a static data member)
4. CNIC (a constant data member)

Count should keep track of how many person objects are created. Set the value of count before any object is created. Display it at the end of the program.

Public member functions:

1. Constant member function to access the Date of Birth of each person (get DoB)
2. Constant member function to access the CNIC of each person (get CNIC)
3. Display function for Person record output (const)

Display a person record Mr. X with DoB 1st January 2001 having CNIC xxxxx-xxxxxxx-x

Problem 3: | (Constant keyword)

Write a menu driven program that will demonstrate the following concepts.

1. Const Variables
2. Constant Function Parameters
3. Constant Return type
4. Const Pointer
5. Pointer to Const Variable
6. Constant Data Members of Class
7. Constant Member Function of Class

Example Run:

Please enter the choice to see the example of above concepts or press -1 to exit:
4
Const Pointer:
If we make a pointer **const**, we cannot change the pointer. This means that the pointer will always point to the same address but we can change the value of that address.
For example
int a = 4;

```
int* const p = &a;   // const pointer p pointing to the variable a

Please enter the choice to see the example of above concepts or press -1 to exit:
-1
```

Problem 4: | (Classes, objects, Constructor, Destructor and Member functions, constant)

You are a programmer for the Home Software Company. You have been assigned to develop a class that models the basic workings of a bank account. The class should perform the following tasks:

- Save the account balance.
- Save the number of transactions performed on the account.
- Allow deposits to be made to the account.
- Allow withdrawals to be taken from the account.
- Calculate interest for the period.
- Report the current account balance at any time.
- Report the current number of transactions at any time.

Private Member Variables

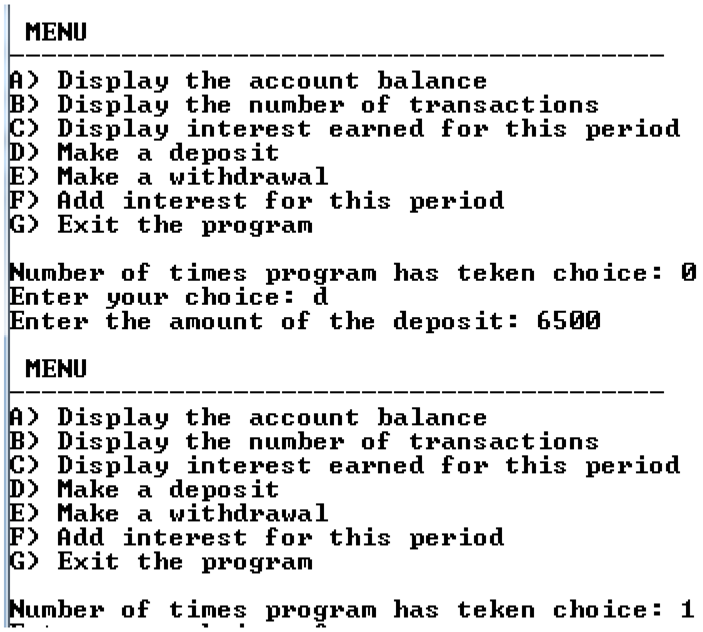
Variable	**Description**
balance	A double that holds the current account balance.
interestRate	A double that holds the interest rate for the period.
interest	A double that holds the interest earned for the current period.
transactions	An integer that holds the current number of transactions.
count	A static integer that hold the total number of time the program has taken choice

Public Member Functions

Function	**Description**
Constructor	Takes arguments to be initially stored in the balance and interestRate members. The default value for the balance is zero and the default value for the interest rate is 4.5%.
setInterestRate	Takes a double argument which is stored in the interestRate member.
makeDeposit	Takes a double argument, which is the amount of the deposit. This argument is added to balance.
withdraw	Takes a double argument which is the amount of the withdrawal. This value is subtracted from the balance, unless the withdrawal amount is greater than the balance. If this happens, the function reports an error.

calcInterest	Takes no arguments. This function calculates the amount of interest for the current period, stores this value in the interest member, and then adds it to the balance member.
incCount	increment the value of count
getCount	return the value of count
getInterestRate	Returns the current interest rate (stored in the interestRate member).
getBalance	Returns the current balance (stored in the balance member).
getInterest	Returns the interest earned for the current period (stored in the interest member).
getTransactions	Returns the number of transactions for the current period (stored in the transactions member).

Note: *All* ***get*** *methods must be constant. Output must in the same format as given on next page.*

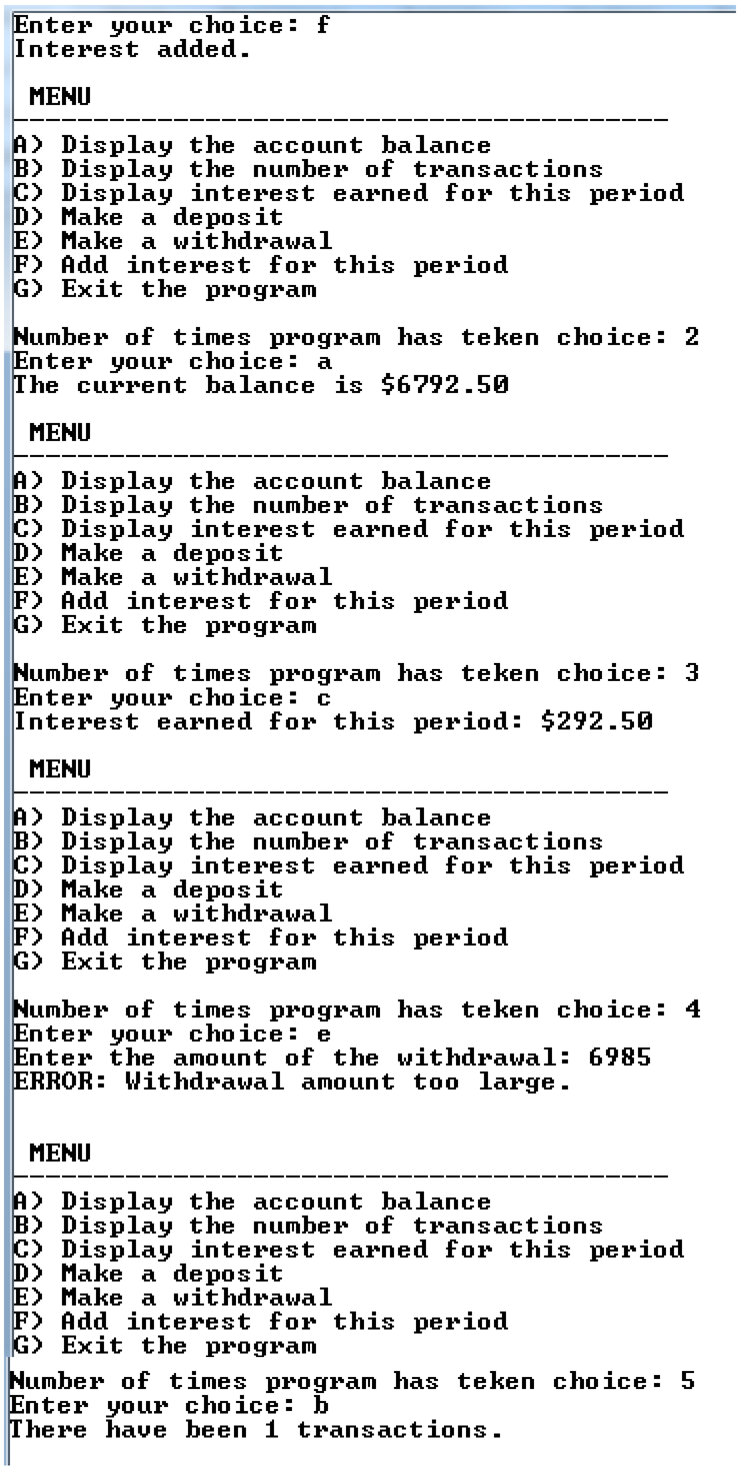


```
 MENU
-----------------------------------------
A> Display the account balance
B> Display the number of transactions
C> Display interest earned for this period
D> Make a deposit
E> Make a withdrawal
F> Add interest for this period
G> Exit the program

Number of times program has teken choice: 0
Enter your choice: d
Enter the amount of the deposit: 6500

 MENU
-----------------------------------------
A> Display the account balance
B> Display the number of transactions
C> Display interest earned for this period
D> Make a deposit
E> Make a withdrawal
F> Add interest for this period
G> Exit the program

Number of times program has teken choice: 1
```



```
Enter your choice: f
Interest added.

 MENU
-----------------------------------------
A) Display the account balance
B) Display the number of transactions
C) Display interest earned for this period
D) Make a deposit
E) Make a withdrawal
F) Add interest for this period
G) Exit the program

Number of times program has teken choice: 2
Enter your choice: a
The current balance is $6792.50

 MENU
-----------------------------------------
A) Display the account balance
B) Display the number of transactions
C) Display interest earned for this period
D) Make a deposit
E) Make a withdrawal
F) Add interest for this period
G) Exit the program

Number of times program has teken choice: 3
Enter your choice: c
Interest earned for this period: $292.50

 MENU
-----------------------------------------
A) Display the account balance
B) Display the number of transactions
C) Display interest earned for this period
D) Make a deposit
E) Make a withdrawal
F) Add interest for this period
G) Exit the program

Number of times program has teken choice: 4
Enter your choice: e
Enter the amount of the withdrawal: 6985
ERROR: Withdrawal amount too large.


 MENU
-----------------------------------------
A) Display the account balance
B) Display the number of transactions
C) Display interest earned for this period
D) Make a deposit
E) Make a withdrawal
F) Add interest for this period
G) Exit the program
Number of times program has teken choice: 5
Enter your choice: b
There have been 1 transactions.
```

Problem 5: | (Inheritance)

Design a class named **PersonData** with the following member variables:

- FirstName
- LastName
- Address

Write the appropriate **accessor** and **mutator** functions for these member variables.

Next, design a class named **CustomerData**, which is derived from the **PersonData** class. The CustomerData

Class should have the following member variables:

- CustomerNumber
- MailingList

The CustomerNumber variable will be used to hold a unique integer for each customer. The MailingList variable should be a bool. It will be set to true if the customer wishes to be on a mailing list, or false if the customer does not wish to be on a mailing list. Write appropriate Accessor and Mutator functions for these member variables. CustomerData class will have the

- InputCustomerData member function which will Input all the data for customer.
- DisplayCustomerData member function which will display all the data for customer.

Demonstrate an object of the CustomerData class in a simple program.

Proper code indentation will hold extra marks!

Best of luck ☺

You are done with your exercise, submit on Teams at given time.