

Bangladesh Army University of Engineering & Technology (BAUET)

Qadirabad Cantonment, Natore-6431



Department of COMPUTER SCIENCE AND ENGINEERING (CSE)

LAB REPORT: 05

Course Code : CSE-2104
Course Title : Object Oriented Programming Laboratory

Experiment No : 05
Experiment Name : Inheritance in Object Oriented Programming

Experiment Date : 23/02/2025
Submission Date : 02/03/2025

Submitted By

Name : Saiyedul Abrar
ID : 0812320205101022
Batch : 18th
Year : 2nd
Semester : 1st
Section : A
Session : 2023-2024

Submitted To

Aroni Saha Prapty
Lecturer
Dept. of CSE, BAUET

Muhtasim
Lecturer
Dept. of CSE, BAUET

Problem 1. Create two classes named Mammals and MarineAnimals. Create another class named BlueWhale which inherits both the above classes. Now, create a function in each of these classes which prints "I am mammal", "I am a marine animal" and "I belong to both the categories: Mammals as well as Marine Animals" respectively. Now, create an object for each of the above class and try calling

- a - function of Mammals by the object of Mammal
- b - function of MarineAnimal by the object of MarineAnimal
- c - function of BlueWhale by the object of BlueWhale
- d - function of each of its parent by the object of BlueWhale

```
Task1.cpp > MarineAnimals
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 class Mammals
5 {
6 public:
7     void mamaldisplay()
8     {
9         cout << "I am mammal" << endl;
10    }
11};
12
13 class MarineAnimals
14 {
15 public:
16     void marinedisplay()
17     {
18         cout << "I am a marine animal" << endl;
19     }
20};
21
22 class BlueWhale : public Mammals, public MarineAnimals
23 {
24 public:
25     void bluewhaledisplay()
26     {
27         cout << "I belong to both categories: Mammals as well as Marine Animals" << endl;
28     }
29};
30
31 int main()
32 {
33     Mammals mammalObj;
34     mammalObj.mamaldisplay();
35
36     MarineAnimals marineObj;
37     marineObj.marinedisplay();
38
39     BlueWhale blueWhaleObj;
40     blueWhaleObj.bluewhaledisplay();
41
42     blueWhaleObj.mamaldisplay();
43     blueWhaleObj.marinedisplay();
44
45     return 0;
46 }
47
48
```

```
C:\WINDOWS\system32\cmd. x + -
I am mammal
I am a marine animal
I belong to both categories: Mammals as well as Marine Animals
I am mammal
I am a marine animal
Press any key to continue . . . |
```

Problem 2. Assume that a bank maintains two kinds of accounts for customers, one called as savings account and the other as current account. The savings account provides compound interest and withdrawal facilities but no check book facility. The current account provides check book facility but no interest. Current account holders should also maintain a minimum balance if the balance falls below this level, a service charge is imposed. Create a class named as account that stores customers name, account number and type of account. From this derive the classes cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions to achieve the following tasks:

- I. Accept deposit from customer and update the balance.
- II. Display the balance.
- III. Compute and deposit interest.
- IV. Permit withdrawal and update the balance.

The screenshot shows a C++ IDE with two windows. The left window displays the source code for 'Task2.cpp', and the right window shows the output of the program in a command prompt.

Source Code (Task2.cpp):

```

4  class Account
5  {
6  public:
7      string cname;
8      int accnum;
9      double balance;
10
11     void deposit(double amount)
12     {
13         balance += amount;
14         cout << "Deposit of " << amount << " successful! New balance: " << balance << endl;
15     }
16
17     void disbalance()
18     {
19         cout << "Account Holder: " << cname << endl;
20         cout << "Account Number: " << accnum << endl;
21         cout << "Current Balance: " << balance << endl;
22     }
23 };
24
25 class SavingsAccount : public Account
26 {
27 public:
28     double intrate;
29
30     void savacc(string name, int acc_no, double bal, double rate)
31     {
32         cname = name;
33         accnum = acc_no;
34         balance = bal;
35         intrate = rate;
36     }
37
38     void compint()
39     {

```

Output (C:\WINDOWS\system32\cmd.exe):

```

Savings Account
Deposit of 1000 successful! New balance: 6000
Interest of 300 added. New balance: 6300
Account Holder: John Doe
Account Number: 101
Current Balance: 6300
Withdrawal of 2000 successful! Remaining balance: 4300
Account Holder: John Doe
Account Number: 101
Current Balance: 4300
Current Account
Deposit of 2000 successful! New balance: 5000
Account Holder: Jane Smith
Account Number: 102
Current Balance: 5000
Withdrawal of 4500 successful!
Balance below minimum! Service charge of 50 applied.
New balance: 450
Account Holder: Jane Smith
Account Number: 102
Current Balance: 450

```

The output demonstrates the functionality of the Account classes, including deposits, interest calculation for SavingsAccount, and withdrawals with service charges for CurrentAccount.