

LAB ASSIGNMENT - 1

ARZOO KHAN

(20MCA11)

MCA

(Semester - II)

April 26 , 2021

—

CSC26: Lab – III (OOP)

—

Prof. S. Zeeshan Hussain

1. Define a Class Largest to obtain the largest of three given numbers.

SOURCE CODE:

```
#include <iostream>
using namespace std;

class Largest
{
private:

    int a, b, c;

public:

    void input()
    {
        cout << "Enter three numbers: ";
        cin >> a >> b >> c;
    }

    void max()
    {
        cout << "Largest number is: ";

        if (a > b)
        {
            if (a > c)
                cout << a;

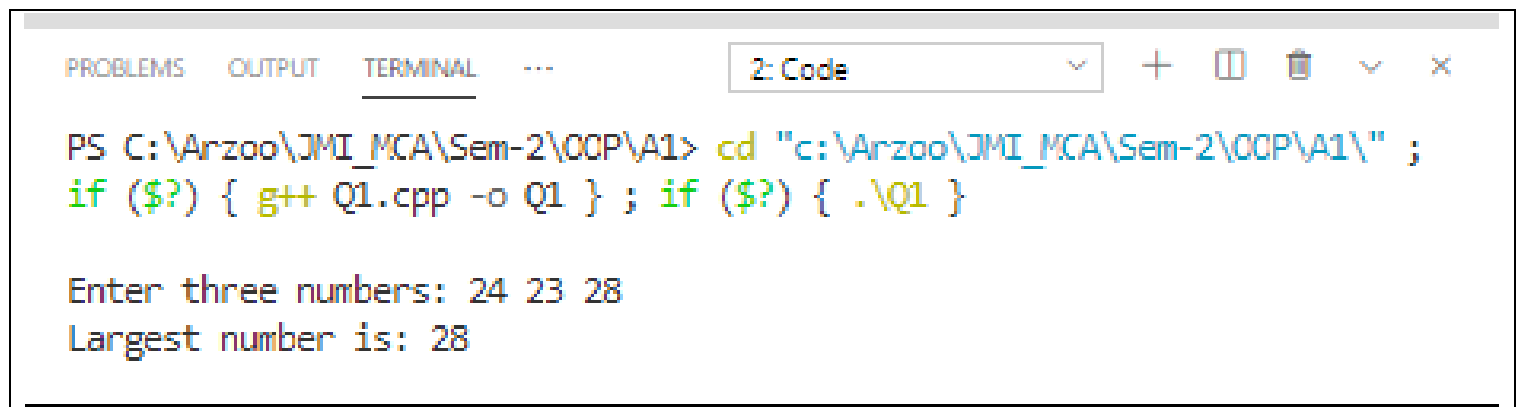
            else
                cout << c;
        }

        else if (b > c)
            cout << b;
```

```
        else  
            cout << c;  
    }  
};
```

```
int main(void)  
{  
    Largest x;  
    x.input();  
    x.max();  
    return 0;  
}
```

OUTPUT:



```
PROBLEMS OUTPUT TERMINAL ... 2: Code  
PS C:\Arzoo\JMI_MCA\Sem-2\OOP\A1> cd "c:\Arzoo\JMI_MCA\Sem-2\OOP\A1\" ;  
if ($?) { g++ Q1.cpp -o Q1 } ; if ($?) { .\Q1 }  
  
Enter three numbers: 24 23 28  
Largest number is: 28
```

2. *Define a class Bank Account to represent a bank account. It contains*

Data Members:

Name of the depositor

Account Number

Type of account

Balance

Member Functions:

To assign initial values

To deposit an amount

To withdraw an amount < amount available

Display the name and balance.

SOURCE CODE:

```
class Bank_Account
{
private:
    char name[20];
    long account_no;
    char account_type;
    float balance = 0;

public:
    void open_account();
    void deposit_money();
    void withdraw_money();
    void display_balance();
};

// Function to open the account
void Bank_Account::open_account()
{
    cout << "\nEnter details: "
         << "\n-----\n";

    cout << "Name: ";
    cin >> name;

    cout << "Account Number: ";
    cin >> account_no;

    cout << "Account Type [Saving (S) or Current (C)]: ";
    cin >> account_type;
```

```

cout << "Enter How much money you want to deposit: Rs ";
cin >> balance;

cout << "\nAccount Created Successfully!\n\n";
}

// Function to deposite money
void Bank_Account ::deposit_money()
{
    float deposite;
    cout << "Enter how much money you want to deposit: Rs ";
    cin >> deposite;
    balance += deposite;
    cout << "Balance Available: Rs " << balance << endl;
}

// Function to withdraw money
void Bank_Account ::withdraw_money()
{
    float withdraw;
    cout << "Enter how much money you want to withdraw: Rs ";
    cin >> withdraw;

    if (withdraw > balance)
        cout << "Insufficient Balance!" << endl;

    else
        balance -= withdraw;

    cout << "Balance Available: Rs " << balance << endl;
}

// Finction to display balance
void Bank_Account ::display_balance()
{
    cout << "Account Number: " << account_no << endl
        << "Name: " << name << endl
        << "Balance: Rs " << balance << endl;
}

```

3. Using the above class Bank Account and by supplying a user id and password allow users to Login using their id and password.

Now if login was successful the user will be able to do the following:

Withdraw money.

Deposit money.

Display Balance.

Logout

Quit the program.

If login was not successful (for example the id or password did not match) then the user will be taken back to the introduction menu.

SOURCE CODE:

```
#include <iostream>
#include <string.h>
using namespace std;
#include "Q2.cpp" // Including .cpp file of Q2 where class Bank_Account is defined

int main(void)
{
    Bank_Account customer;
    customer.open_account();

    char id[16], password[16];
    int choice;

logout:
    while (1)
    {
        cout << "\nId: ";
        cin >> id;
```

```

cout << "Password: ";
cin >> password;

if (!strcmp(id, "abcd") && !strcmp(password, "abcd1234"))
{
    while (1)
    {
        cout << "\n1) Withdraw Money." << endl
            << "2) Deposit Money." << endl
            << "3) Display Balance." << endl
            << "4) Logout" << endl
            << "5) Quit the program." << endl
            << "\nEnter your choice: ";

        cin >> choice;

        switch (choice)
        {
            case 1:
                cout << "\n-----\n";
                customer.withdraw_money();
                cout << "-----\n";
                break;

            case 2:
                cout << "\n-----\n";
                customer.deposit_money();
                cout << "-----\n";
                break;

            case 3:
                cout << "\n-----\n";

```

```

        customer.display_balance();
        cout << "-----\n";
        break;

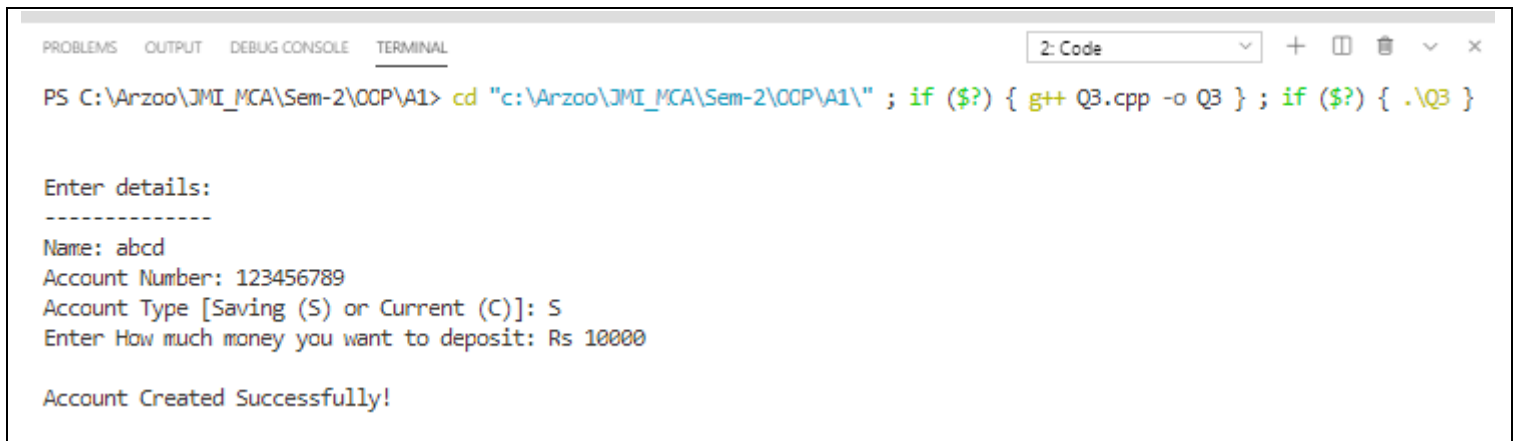
    case 4:
        goto logout;

    case 5:
        cout << "\n\n";
        return 0;

    default:
        cout << "Invalid Choice!\n\n";
    }
}
}
else
{
    cout << "\nIncorrect Id or Password! Try again.\n\n";
}
}
}
}

```

OUTPUT:



The screenshot shows a terminal window with the following content:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: Code
PS C:\Arzoo\JMI_MCA\Sem-2\OCP\A1> cd "c:\Arzoo\JMI_MCA\Sem-2\OCP\A1\" ; if ($?) { g++ Q3.cpp -o Q3 } ; if ($?) { .\Q3 }

Enter details:
-----
Name: abcd
Account Number: 123456789
Account Type [Saving (S) or Current (C)]: S
Enter How much money you want to deposit: Rs 10000

Account Created Successfully!

```


id: abcd
Password: abcd1234

- 1) Withdraw Money.
- 2) Deposit Money.
- 3) Display Balance.
- 4) Logout
- 5) Quit the program.

Enter your choice: 1

Enter how much money you want to withdraw: Rs 100
Balance Available: Rs 9900

- 1) Withdraw Money.
- 2) Deposit Money.
- 3) Display Balance.
- 4) Logout
- 5) Quit the program.

Enter your choice: 2

Enter how much money you want to deposit: Rs 500
Balance Available: Rs 10400

- 1) Withdraw Money.
- 2) Deposit Money.
- 3) Display Balance.
- 4) Logout
- 5) Quit the program.

Enter your choice: 4

id: abcd
Password: thrt448

Incorrect Id or Password! Try again.

id: abcd
Password: abcd1234

- 1) Withdraw Money.
- 2) Deposit Money.
- 3) Display Balance.
- 4) Logout
- 5) Quit the program.

Enter your choice: 3

Account Number: 123456789
Name: abcd
Balance: Rs 10400

- 1) Withdraw Money.
- 2) Deposit Money.
- 3) Display Balance.
- 4) Logout
- 5) Quit the program.

Enter your choice: 5

4. *Create a class Time to add Two times provided in hour-minute format. Use member Functions*

a) *void input() to provide hour and minute.*

b) *void gettime(int, int) to take hour and minute entered by user.*

c) *sum(time<obj>) to add minutes and hours for two Time objects. If minutes is > 60 add 1 with hour.*

d) *void display() to display the result.*

SOURCE CODE:

```
#include <iostream>
using namespace std;

class Time
{
private:
    int hour, minute;

public:
    void input()
    {
        cout << "Enter time in hour-minute format: ";
        cin >> hour >> minute;
    }

    void gettime(int h, int m)
    {
        hour = h;
        minute = m;
    }

    Time sum(Time t)
    {
        Time temp;

        temp.hour = t.hour + hour;
        temp.minute = t.minute + minute;

        if (temp.minute > 60)
        {
            temp.hour++;
            temp.minute -= 60;
        }
    }
}
```

```

    }

    return (temp);
}

void display()
{
    cout << hour << "h " << minute << "min" << endl;
}

};

int main(void)
{
    Time t1, t2, t3;

    t1.gettime(4, 57);
    t2.input();
    t3 = t1.sum(t2);

    cout << "\nt1 = ";
    t1.display();

    cout << "t2 = ";
    t2.display();

    cout << "t3 = t1 + t2 = ";
    t3.display();

    cout << "\n\n";
    return 0;
}

```

OUTPUT:

```

PROBLEMS OUTPUT TERMINAL ... 2: Code
PS C:\Arzoo\JMI_MCA\Sem-2\OCP\A1> cd "c:\Arzoo\JMI_MCA\Sem-2\OCP\A1\" ;
if ($?) { g++ Q4.cpp -o Q4 } ; if ($?) { .\Q4 }
Enter time in hour-minute fromat: 10 36

t1 = 4h 57min
t2 = 10h 36min
t3 = t1 + t2 = 15h 33min

```

5. Write a C++ program to add two complex numbers using class Complex having real and image as data members.

(i) The class Complex contains three constructors.

One with no parameter. (Used for the object for storing result.)

With one parameter (Same value for real and imaginary part)

With two parameters (Different Values for real and imaginary part)

(ii) Two friend functions

a) One to add two complex numbers by taking two reference variables of class complex and returning another reference.

b) To display the result.

SOURCE CODE:

```
#include <iostream>
using namespace std;

class Complex
{
private:
    int real, imaginary;

public:
    Complex()
    {
    }

    Complex(int a)
    {
        real = imaginary = a;
    }

    Complex(int a, int b)
    {
        real = a;
        imaginary = b;
    }

    friend Complex operator+(Complex, Complex);
```

```

    friend void display(Complex);
};

Complex operator+(Complex z1, Complex z2)
{
    Complex temp;

    temp.real = z1.real + z2.real;
    temp.imaginary = z1.imaginary + z2.imaginary;

    return (temp);
}

void display(Complex z)
{
    if (z.real != 0)
    {
        cout << z.real;
    }

    if (z.imaginary != 0)
    {
        cout << (z.imaginary > 0 && z.real != 0 ? "+" : "") << z.imaginary << "i";
    }
}

int main(void)
{
    Complex z1(2);
    Complex z2(-4, -2);
    Complex z3;

    z3 = z1 + z2;

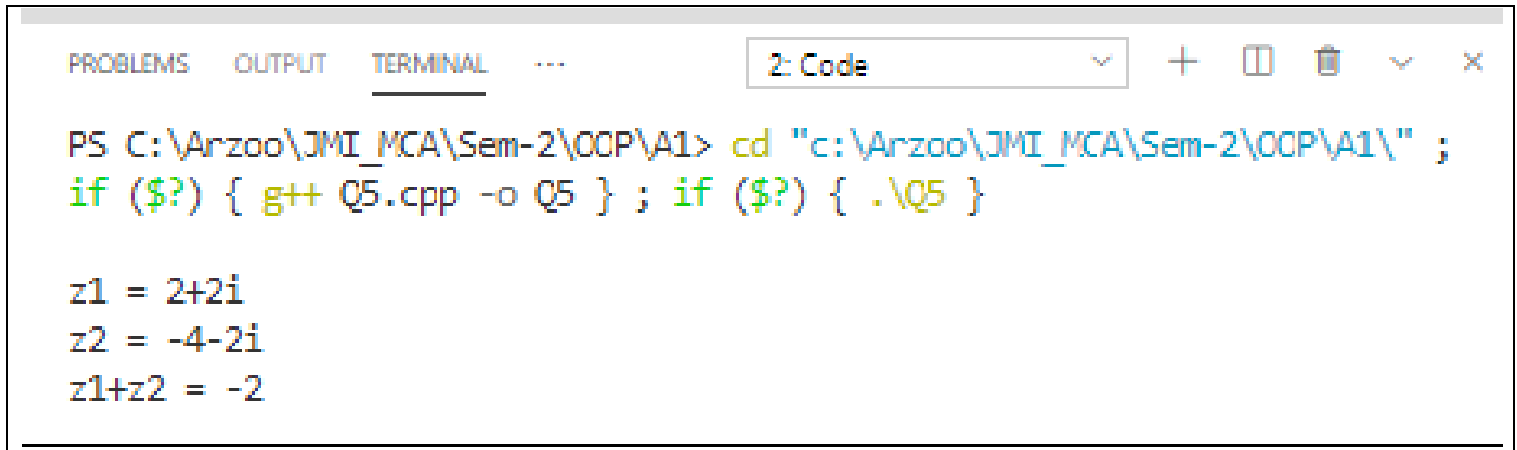
    cout << "\nz1 = ";
    display(z1);

    cout << "\nz2 = ";
    display(z2);
}

```

```
cout << "\nz1+z2 = ";  
display(z1 + z2);  
  
return 0;  
}
```

OUTPUT:



The screenshot shows a C++ IDE with a terminal window. The terminal displays the command prompt, the directory change command, the compilation command, and the program's output.

```
PROBLEMS OUTPUT TERMINAL ... 2: Code + [ ] [ ] [ ] [ ] [ ] [ ]  
PS C:\Arzoo\JMI_MCA\Sem-2\OCP\A1> cd "c:\Arzoo\JMI_MCA\Sem-2\OCP\A1\" ;  
if ($?) { g++ Q5.cpp -o Q5 } ; if ($?) { .\Q5 }  
  
z1 = 2+2i  
z2 = -4-2i  
z1+z2 = -2
```
