Assignment 1(OOP)

MCA SEM-II, LAB-III(OOP)

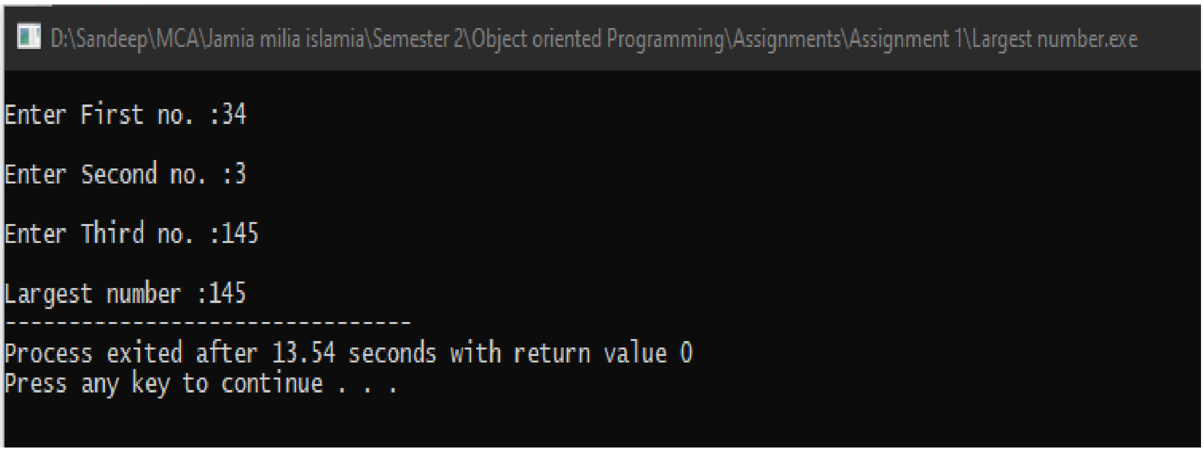
Sandeep Bhatt, MCA051

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

Sol:

```
#include<iostream>
using namespace std;
class Largest
{
        private:
                int First_no,Second_no,Third_no;
    public:
        void input()
        {
                cout<<"\nEnter First no. :";
                cin>>First_no;
                cout<<"\nEnter Second no. :";
                cin>>Second_no;
                cout<<"\nEnter Third no. :";
                cin>>Third_no;
        }
        void calc()
        {
                int largest;

                largest=((First_no>=Second_no)&&(First_no>=Third_no)?

            First_no:(Second_no>=First_no)&&(Second_no>=Third_no)?
                                        Second_no:Third_no);

                cout<<"\nLargest number :"<<largest;
        }
};
int main()
{
        Largest l;
        l.input();
        l.calc();       }
```

OUTPUT:



```
D:\Sandeep\MCA\Jamia milia islamia\Semester 2\Object oriented Programming\Assignments\Assignment 1\Largest number.exe

Enter First no. :34

Enter Second no. :3

Enter Third no. :145

Largest number :145
--------------------------------
Process exited after 13.54 seconds with return value 0
Press any key to continue . . .
```

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.

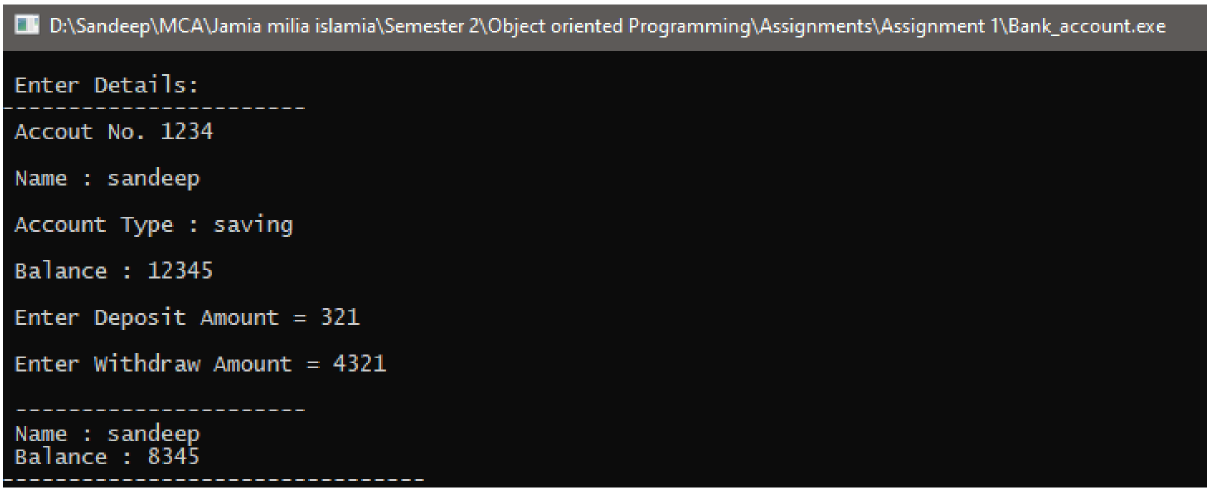
Sol:

```
#include<iostream>

#include<stdio.h>

#include<string.h>
```

```
using namespace std;

class bank_account
{
        int acno;
        char nm[100], acctype[100];
        float bal;
   public:
        bank_account(int acc_no, char *name, char *acc_type, float balance)
//Parameterized Constructor
        {
                acno=acc_no;
                strcpy(nm, name);
                strcpy(acctype, acc_type);
                bal=balance;
        }
        void deposit();
        void withdraw();
        void display();
};
void bank_account::deposit()   //depositing an amount
{
        int deposit_amount;
```

```
    cout<<"\n Enter Deposit Amount = ";
    cin>>deposit_amount;
    bal+=deposit_amount;
}
void bank_account::withdraw()  //withdrawing an amount
{
    int withdraw_amount;
    cout<<"\n Enter Withdraw Amount = ";
    cin>>withdraw_amount;
    if(withdraw_amount>bal)
        cout<<"\n Cannot Withdraw Amount";
    bal-=withdraw_amount;
}
void bank_account::display()  //displaying the details
{
    cout<<"\n ----------------------";
//    cout<<"\n Accout No. : "<<acno;
    cout<<"\n Name : "<<nm;
//    cout<<"\n Account Type : "<<acctype;
    cout<<"\n Balance : "<<bal;
}
int main()
{
    int acc_no;

    char name[100], acc_type[100];

    float balance;

    cout<<"\n Enter Details: \n";

    cout<<"-----------------------";

    cout<<"\n Accout No. ";

    cin>>acc_no;

    cout<<"\n Name : ";

    cin>>name;

    cout<<"\n Account Type : ";

    cin>>acc_type;

    cout<<"\n Balance : ";

    cin>>balance;


    bank_account b1(acc_no, name, acc_type, balance);  //object is created

    b1.deposit(); //calling member functions

    b1.withdraw(); // calling member functions

    b1.display(); //displaying the outcome

    return 0;

}
```

Output:



D:\Sandeep\MCA\Jamia milia islamia\Semester 2\Object oriented Programming\Assignments\Assignment 1\Bank_account.exe

```
Enter Details:
-----------------------
Accout No. 1234

Name : sandeep

Account Type : saving

Balance : 12345

Enter Deposit Amount = 321

Enter Withdraw Amount = 4321

----------------------
Name : sandeep
Balance : 8345
-------------------------------
```

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.

Quit the program.

If login was not successful (for example the id or password did not match) then the user will be

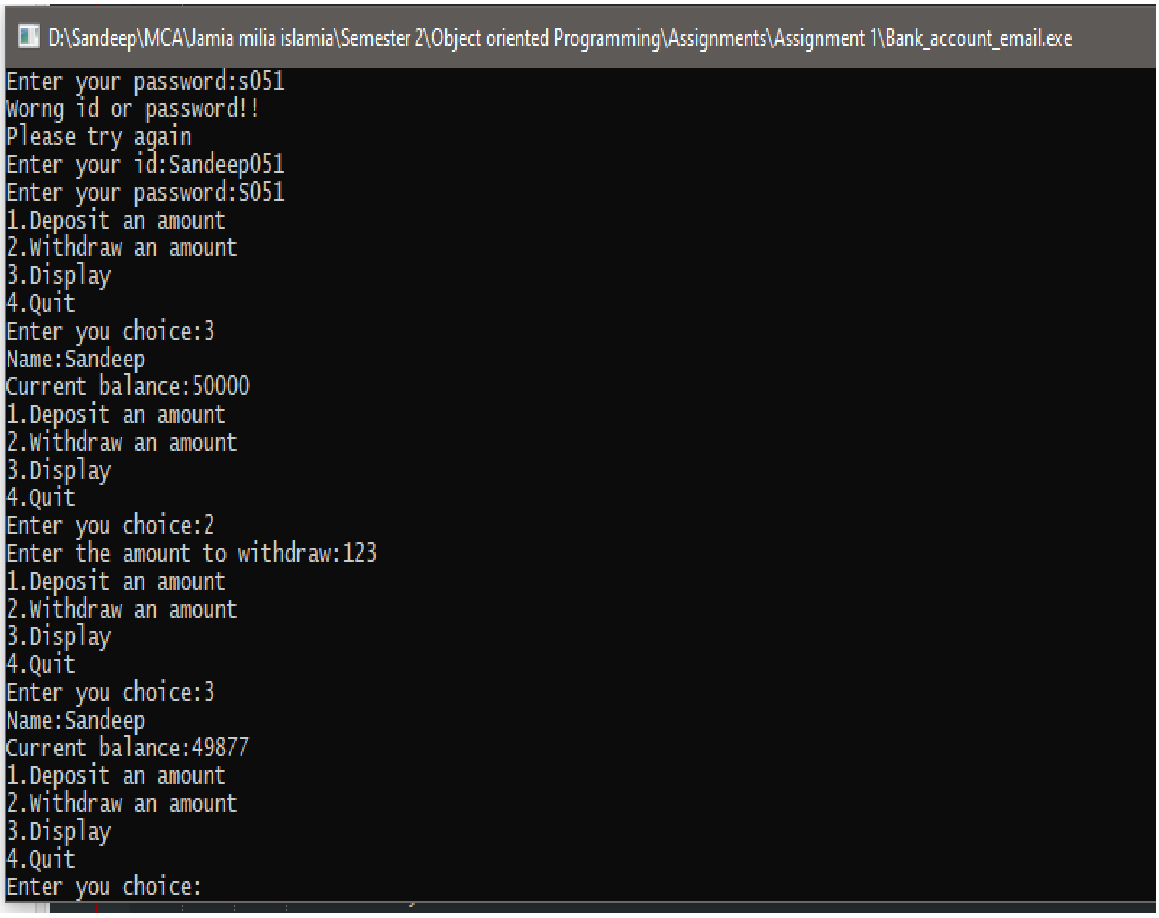
taken back to the introduction menu.

Sol:

```
#include<iostream>

using namespace std;

class Bank_Account

{

    private:

            string name;

            int a_number;

            string type;

            float balance;

    public:

            Bank_Account(string name,int a_number,string type,float
balance)

            {

                    this->name=name;

                    this->a_number=a_number;

                    this->type=type;

                    this->balance=balance;

            }

            void deposit(int d_amount)

            {

                    this->balance=this->balance+d_amount;

                    cout<<"Amount sucessfully deposited"<<endl;

            }
```

```
        void withdraw(int w_amount)

        {

            if(this->balance-w_amount>5000)

                this->balance=this->balance-w_amount;

            else

                cout<<"Current balance not sufficient"<<endl;

        }

        void display()

        {

            cout<<"Name:"<<this->name<<endl;

            cout<<"Current balance:"<<this->balance<<endl;

        }

};

int main()

{

    string id="",pass="";

    int chance=0;

    while(id!="Sandeep051"&&pass!="S051")

    {

        if(chance!=0)

            cout<<"Worng id or password!!\nPlease try again"<<endl;

        if(chance==3)

        {
                cout<<"Please try again later"<<endl;
                return 0;
        }
        cout<<"Enter your id:";
        cin>>id;
        cout<<"Enter your password:";
        cin>>pass;
        chance++;
}
Bank_Account obj("Sandeep",1234,"savings",50000);
char c='Y';
int choice,amount;
do
{
        cout<<"1.Deposit an amount"<<endl;
        cout<<"2.Withdraw an amount"<<endl;
        cout<<"3.Display"<<endl;
        cout<<"4.Quit"<<endl;
        cout<<"Enter you choice:";
        cin>>choice;
        switch(choice)
        {
                case 1:
                        cout<<"Enter the amount to deposit:";

                        cin>>amount;

                        obj.deposit(amount);

                        break;

                case 2:

                        cout<<"Enter the amount to withdraw:";

                        cin>>amount;

                        obj.withdraw(amount);

                        break;

                case 3:

                        obj.display();

                        break;

                case 4:

                        c='N';

                        break;

                default:

                        cout<<"Wrong choice"<<endl;

                        c='N';

                        break;

        }

}while(c=='Y'||c=='y');

cout<<"Thanks for visiting!"<<endl;

return 0;}
```

OUTPUT:



D:\Sandeep\MCA\Jamia milia islamia\Semester 2\Object oriented Programming\Assignments\Assignment 1\Bank_account_email.exe

```
Enter your password:s051
Worng id or password!!
Please try again
Enter your id:Sandeep051
Enter your password:S051
1.Deposit an amount
2.Withdraw an amount
3.Display
4.Quit
Enter you choice:3
Name:Sandeep
Current balance:50000
1.Deposit an amount
2.Withdraw an amount
3.Display
4.Quit
Enter you choice:2
Enter the amount to withdraw:123
1.Deposit an amount
2.Withdraw an amount
3.Display
4.Quit
Enter you choice:3
Name:Sandeep
Current balance:49877
1.Deposit an amount
2.Withdraw an amount
3.Display
4.Quit
Enter you choice:
```

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 <oj>) to add minutes and hours for two Time objects. If minutes is > 60 add 1 with

hour.

d) void display() to display the result.

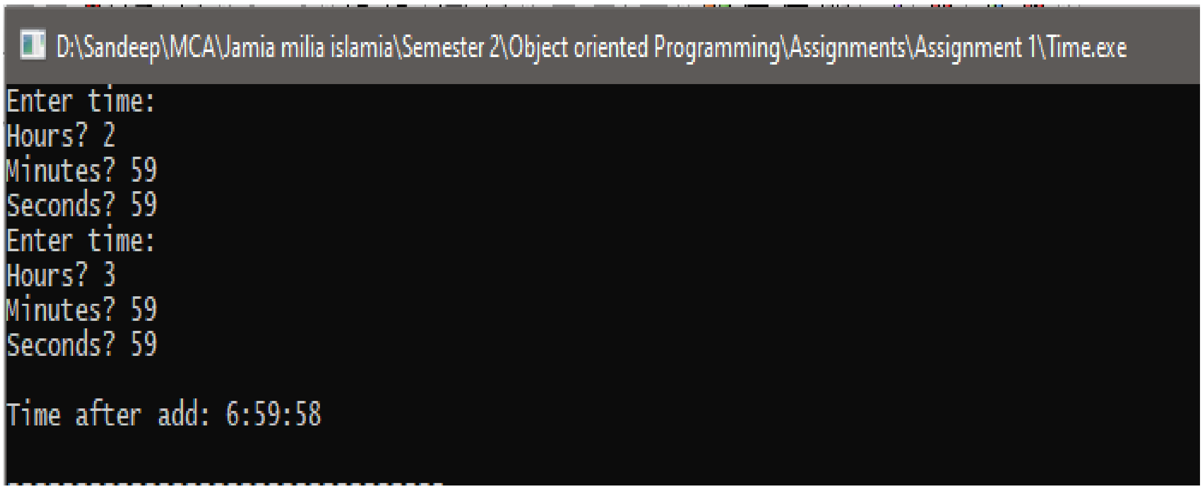
Sol:

```
#include <iostream>

using namespace std;


class Time

{

private:

   int hours;

   int minutes;

   int seconds;


public:

   void getTime(void);

   void putTime(void);

   void addTime(Time T1,Time T2);
```

```
};

void Time::getTime(void)
{
   cout << "Enter time:" << endl;
   cout << "Hours? ";    cin>>hours;
   cout << "Minutes? ";  cin>>minutes;
   cout << "Seconds? ";  cin>>seconds;
}

void Time::putTime(void)
{
   cout << endl;
   cout << "Time after add: ";
   cout << hours << ":" << minutes << ":" << seconds << endl;
}

void Time::addTime(Time T1,Time T2)
{

   this->seconds=T1.seconds+T2.seconds;
   this->minutes=T1.minutes+T2.minutes + this->seconds/60;;
   this->hours= T1.hours+T2.hours + (this->minutes/60);
```

```
    this->minutes %=60;

    this->seconds %=60;

}




int main()

{

    Time T1,T2,T3;

    T1.getTime();

    T2.getTime();

    //add two times

    T3.addTime(T1,T2);

    T3.putTime();


    return 0;

}
```

Output:



D:\Sandeep\MCA\Jamia milia islamia\Semester 2\Object oriented Programming\Assignments\Assignment 1\Time.exe

```
Enter time:
Hours? 2
Minutes? 59
Seconds? 59
Enter time:
Hours? 3
Minutes? 59
Seconds? 59

Time after add: 6:59:58
```

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

imag 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.

Sol:

```
//adding two complex numbers using class
#include<iostream>

using namespace std;

class complex

{
```

```
int real,imag;
public:
void set()
{
cout<<"\nEnter real and imag part  : ";
cin>>real>>imag;
}
friend complex sum(complex,complex);
void display();
};
void complex::display()
{
cout<<"\nThe sum of complex num is : "<<real<<" +i "<<imag;
```

```
}
complex sum(complex a,complex b)
{
complex t;
t.real=a.real+b.real;
t.imag=a.imag+b.imag;
return t;
}
int main()
{
complex a,b,c;
a.set();
b.set();
```

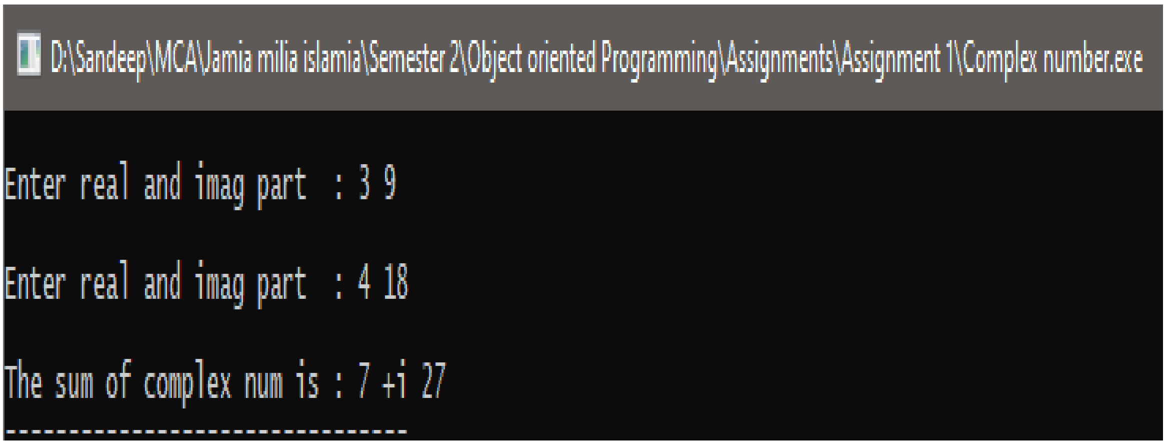
c=sum(a,b);

c.display();

return(0);

}

Output:



```
D:\Sandeep\MCA\Jamia milia islamia\Semester 2\Object oriented Programming\Assignments\Assignment 1\Complex number.exe

Enter real and imag part  : 3 9

Enter real and imag part  : 4 18

The sum of complex num is : 7 +i 27
--------------------------------
```