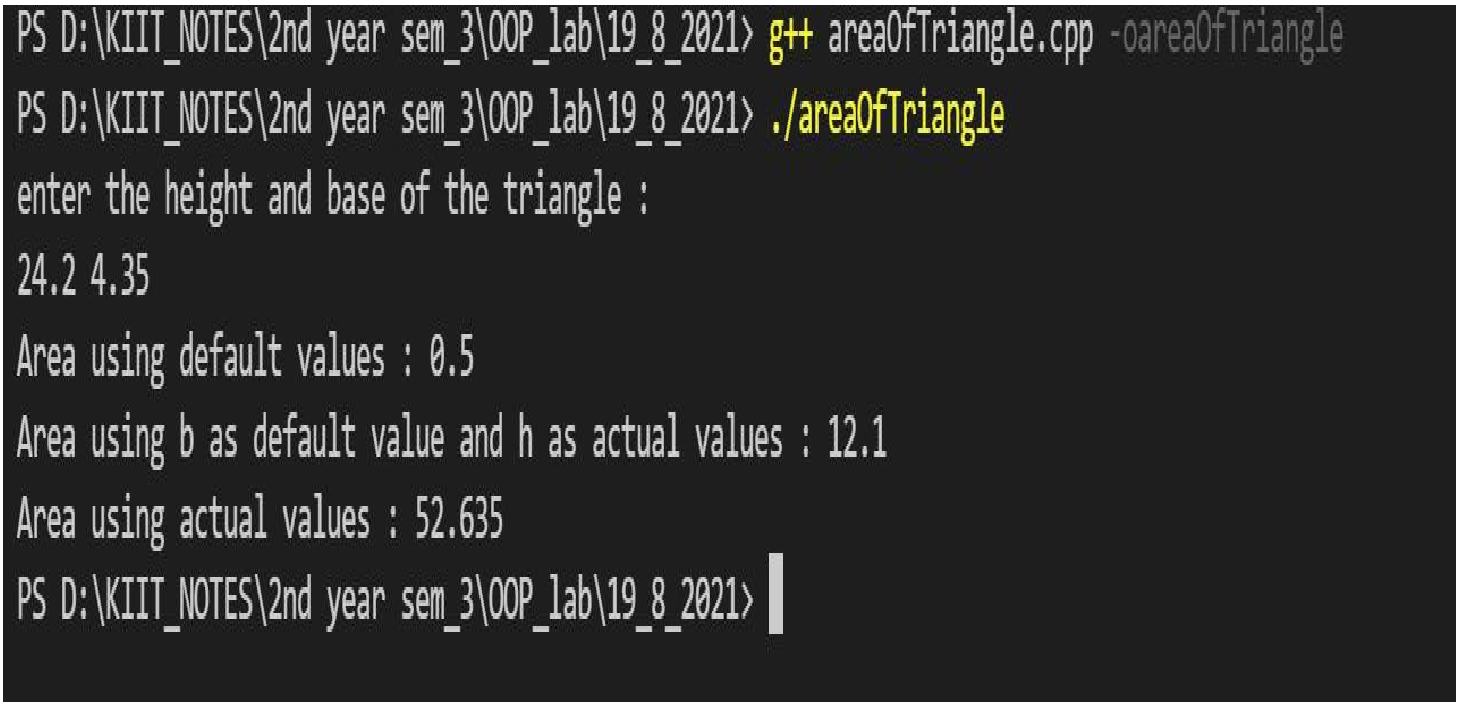
Name - Akriti Choudhary
Roll number - 2005776
Lab4
Subject - OOP lab
Class - B14
Branch - CSE
Date- 19/08/2021

*Question 1 :-WAP to calculate area of a triangle using default and actual values (Area of a triangle =0.5 * h *b)*

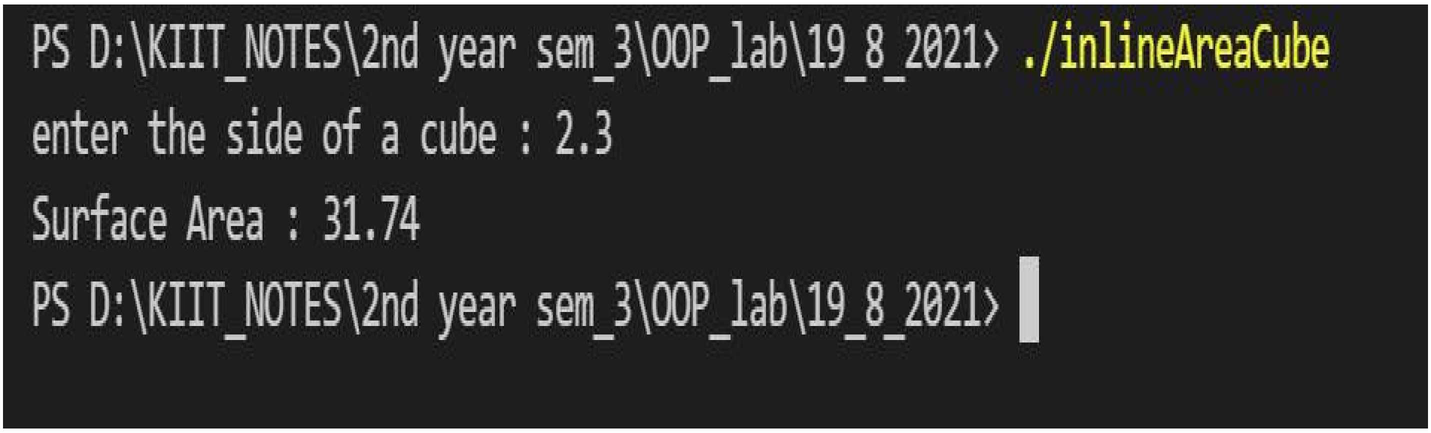
```
#include <iostream>
#include <iomanip>
using namespace std;
class area
{
public:
   float areaTri(float h = 1.0, float b = 1.0)
   {
      return (0.5 * h * b);
   }
};

int main()
{
   float h, b;
   cout << "enter the height and base of the triangle : " << endl;
   cin >> h;
   cin >> b;

   area obj;
   cout << "Area using default values : " << obj.areaTri() << endl;
   cout << "Area using b as default value and h as actual values : " << obj.areaTri(h) << endl;
   cout << "Area using actual values : " << obj.areaTri(h, b) << endl;
   return 0;
}
```



```
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\19_8_2021> g++ areaOfTriangle.cpp -oareaOfTriangle
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\19_8_2021> ./areaOfTriangle
enter the height and base of the triangle :
24.2 4.35
Area using default values : 0.5
Area using b as default value and h as actual values : 12.1
Area using actual values : 52.635
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\19_8_2021>
```

Question 2 :- WAP to define a function as inline to calculate area of a cube.

```
#include <iostream>
#include <iomanip>
using namespace std;
class area
{
public:
   inline float areaCube(float a)
   {
      return (6 * a * a);
   }
};

int main()
{
   float a;
   cout << "enter the side of a cube : ";
   cin >> a;

   area obj;
   cout << "Surface Area : " << obj.areaCube(a) << endl;
   return 0;
}
```

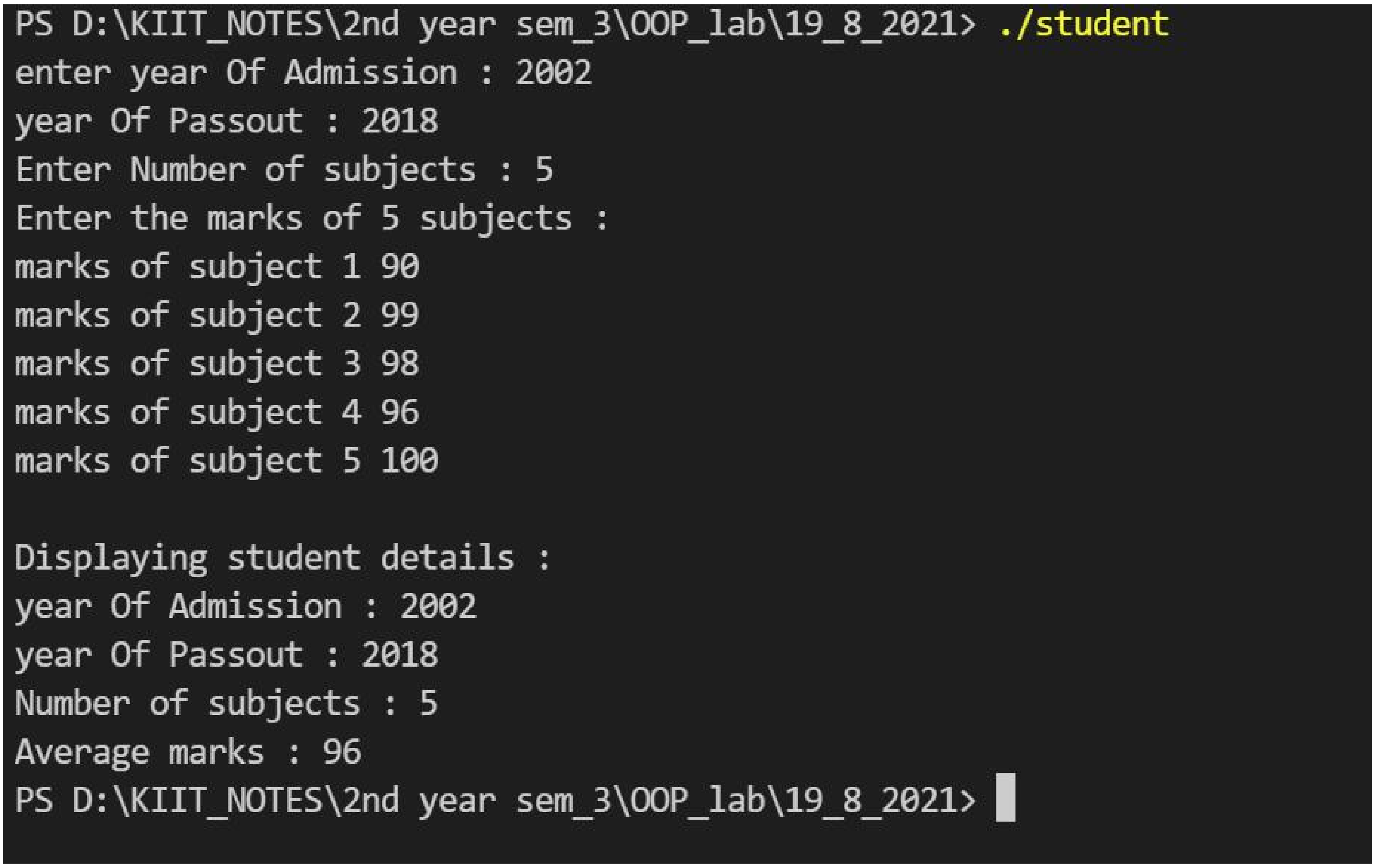


```
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\19_8_2021> ./inlineAreaCube
enter the side of a cube : 2.3
Surface Area : 31.74
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\19_8_2021>
```

Question 3 :-WAP to create a class student with data members yearOfAdmission ,yearOfPassout. Calculate average marks of student for n number of subjects through function with arguments initialized and display the corresponding value of data members.

```
#include <iostream>

using namespace std;
class student
{
public:
   student(int yA, int yP)
      : yearOfAdmission(yA), yearOfPassout(yP)
   {
   }
   float avgMarks(int marks[], int n);
   void display(int n);

private:
   int yearOfAdmission;
   int yearOfPassout;
   int avg;
};
float student::avgMarks(int marks[], int n)
{
   int sum = 0;
   cout << "Enter the marks of " << n << " subjects :" << endl;
   for (int i = 0; i < n; ++i)
   {
      cout << "marks of subject " << i + 1 << " ";
      cin >> marks[i];
      sum += marks[i];
   }
   avg = sum / n;
}
void student::display(int n)
{
   cout << endl;
   cout << "Displaying student details :" << endl;
   cout << "year Of Admission : " << yearOfAdmission << endl;
   cout << "year Of Passout : " << yearOfPassout << endl;
   cout << "Number of subjects : " << n << endl;
   cout << "Average marks : " << avg << endl;
}
int main()
{
   int yA;
   int yP;
   int n;
   cout << "enter year Of Admission : ";
   cin >> yA;

   cout << "year Of Passout : ";
   cin >> yP;
```

```
    cout << "Enter Number of subjects : ";
    cin >> n;
    int marks[n];
    student obj(yA, yP);
    obj.avgMarks(marks,n);
    obj.display(n);
    return 0;
}
```



```
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\19_8_2021> ./student
enter year Of Admission : 2002
year Of Passout : 2018
Enter Number of subjects : 5
Enter the marks of 5 subjects :
marks of subject 1 90
marks of subject 2 99
marks of subject 3 98
marks of subject 4 96
marks of subject 5 100

Displaying student details :
year Of Admission : 2002
year Of Passout : 2018
Number of subjects : 5
Average marks : 96
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\19_8_2021>
```

Question 4 :- WAP to define a class to represent a bank account. Include the following-

Data members :

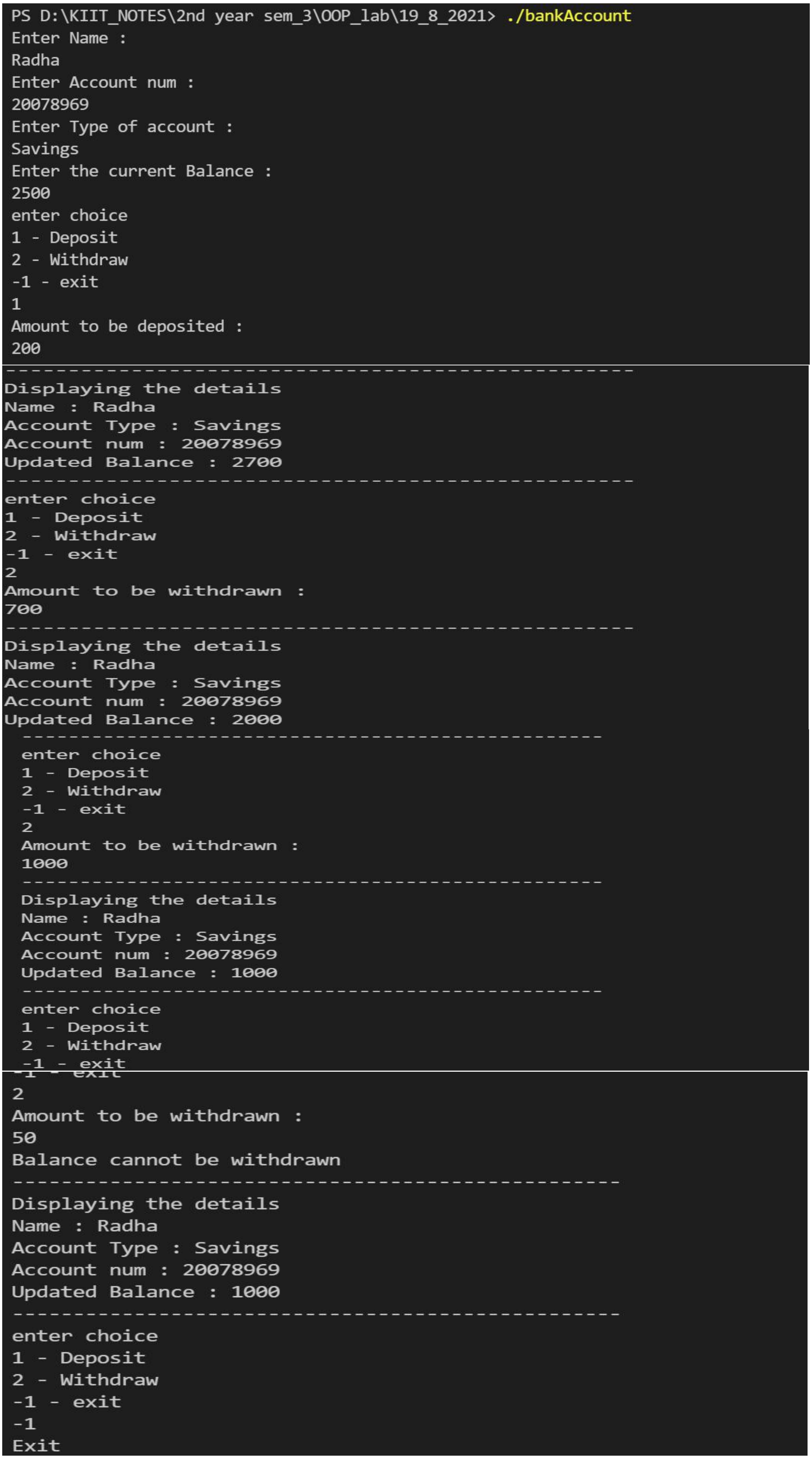
Name of the depositor , account number , type of account , Balance amount in account

Member functions:

To assign initial values ,to deposit an amount ,to withdraw an amount after checking balance ,to display name and balance.

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

class Account
{
public:
   Account()
      : accNum(0), balance(0.0)
   {
   }
   Account(string n, int accn, string t, double b)
      : name(n), accNum(accn), type(t), balance(b)
   {
   }
   void deposit(double amt)
   {
      balance += amt;
   }
   void withdraw(double amt)
   {
      if (balance <= 1000)
      {
         cout << "Balance cannot be withdrawn" << endl;
      }
      else if ((balance - amt) < 1000)
      {
         cout << "After performing the operation Balance will be below 1000" << endl;
         cout << "Balance cannot be withdrawn" << endl;
      }
      else
         balance -= amt;
   }
   void display()
   {
      cout << "--------------------------------------------------" << endl;
      cout << "Displaying the details" << endl;
      cout << "Name : " << name << endl;
      cout << "Account Type : " << type << endl;
      cout << "Account num : " << accNum << endl;
      cout << "Updated Balance : " << balance << endl;
      cout << "--------------------------------------------------" << endl;
   }

private:
   string name;
```

```
    int accNum;
    string type;
    double balance;
};

int main()
{
    double amt;
    double w_amt;

    string n;
    cout << "Enter Name : " << endl;
    cin >> n;
    cout << "Enter Account num : " << endl;
    int accn;
    cin >> accn;
    cout << "Enter Type of account : " << endl;
    string t;
    cin >> t;
    cout << "Enter the current Balance : " << endl;
    double b;
    cin >> b;
    Account obj(n, accn, t, b);
    int ch = 0;

    while (ch != -1)
    {
        cout << "enter choice \n1 - Deposit \n2 - Withdraw\n-1 - exit" << endl;
        cin >> ch;

        switch (ch)
        {
        case 1:
            cout << "Amount to be deposited : " << endl;
            cin >> amt;
            obj.deposit(amt);
            obj.display();
            break;

        case 2:
            cout << "Amount to be withdrawn : " << endl;
            cin >> w_amt;
            obj.withdraw(w_amt);
            obj.display();
            break;
        case -1:
            break;
        default:
            cout << "Invalid choice" << endl;
        }
    }

    cout << "Exit" << endl;

    return 0;
}
```



```
PS D:\KIIT_NOTES\2nd year sem_3\OOP_lab\19_8_2021> ./bankAccount
Enter Name :
Radha
Enter Account num :
20078969
Enter Type of account :
Savings
Enter the current Balance :
2500
enter choice
1 - Deposit
2 - Withdraw
-1 - exit
1
Amount to be deposited :
200
-----------------------------------------------------
Displaying the details
Name : Radha
Account Type : Savings
Account num : 20078969
Updated Balance : 2700
-----------------------------------------------------
enter choice
1 - Deposit
2 - Withdraw
-1 - exit
2
Amount to be withdrawn :
700
-----------------------------------------------------
Displaying the details
Name : Radha
Account Type : Savings
Account num : 20078969
Updated Balance : 2000
-----------------------------------------------------
enter choice
1 - Deposit
2 - Withdraw
-1 - exit
2
Amount to be withdrawn :
1000
-----------------------------------------------------
Displaying the details
Name : Radha
Account Type : Savings
Account num : 20078969
Updated Balance : 1000
-----------------------------------------------------
enter choice
1 - Deposit
2 - Withdraw
-1 - exit
2
Amount to be withdrawn :
50
Balance cannot be withdrawn
-----------------------------------------------------
Displaying the details
Name : Radha
Account Type : Savings
Account num : 20078969
Updated Balance : 1000
-----------------------------------------------------
enter choice
1 - Deposit
2 - Withdraw
-1 - exit
-1
Exit
```