NAME -FIZA SIDDIQUI

Reg. no.-20BCE10077

SUB. CODE -CSE2001

SLOT -E11+E12+E13

SEMESTER -Winter Semester 2020-21

Faculty name- Dr. Venkatachalam

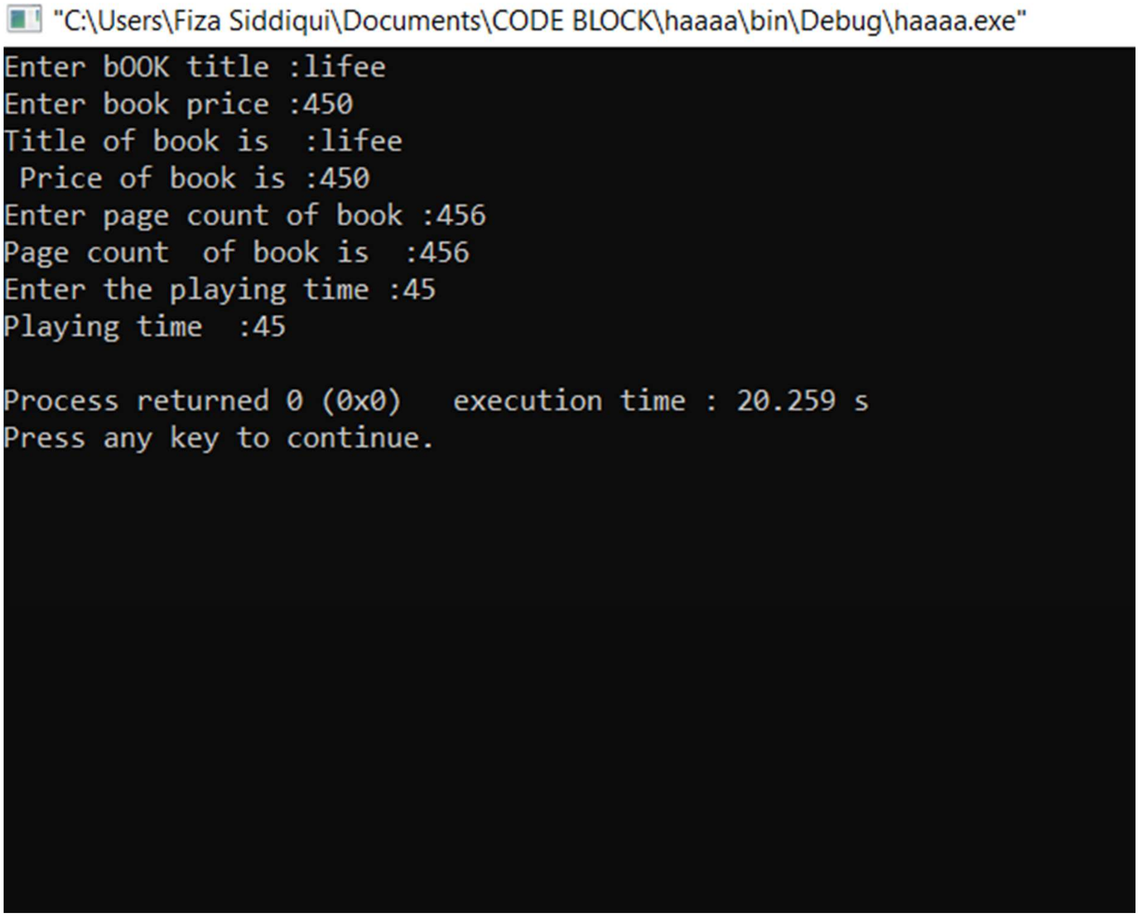
Assignment 2

1). Imagine a publishing Co. that markets both book and audio cassette version of its works. Create a class Publication that stores the title(string) and price(float) of a publication. From here, derive two classes : Book, which adds a page count(type int) and Tape which adds a playing time in minutes(float). Each of these three classes should have a getData() function to get its data from the user at the keyboard, and a putData() function to display its data.

SOURCE CODE:

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


class Publication
{


public:
    char title[20] ;
    float price ;


 public:
   void  getData()
  {
     cout<<"Enter bOOK title :";
     cin>>title;
     cout<<"Enter book price :";
     cin>>price;
}
   void putData()
   {
       cout<<"Title of book is  :"<<title<<endl;
     cout<<  " Price of book is :"<<price<<endl;
  }
};

class book: public Publication
{
public:
  int pagecount;

public:
   void getData()
{
cout<<"Enter page count of book :";
cin>>pagecount;
}
void putData()
  {
cout<<"Page count  of book is  :"<<pagecount<<endl;

  }
};
class Tape :public Publication
{
   float time ;

 public:
      void getData()
   {
     cout<<"Enter the playing time :";
       cin>>time;

  }
   void putData()
   {

       cout<<"Playing time  :"<<time<<endl;

   }
};

int main()
{   Publication oo;
    book ob ;
    Tape tt;
    oo.getData();
           oo.putData();
           ob.getData();
           ob.putData();
           tt.getData();
           tt.putData();
           return 0;
}
```

OUTPUT:



"C:\Users\Fiza Siddiqui\Documents\CODE BLOCK\haaaa\bin\Debug\haaaa.exe"

```
Enter bOOK title :lifee
Enter book price :450
Title of book is  :lifee
 Price of book is :450
Enter page count of book :456
Page count  of book is  :456
Enter the playing time :45
Playing time  :45

Process returned 0 (0x0)   execution time : 20.259 s
Press any key to continue.
```

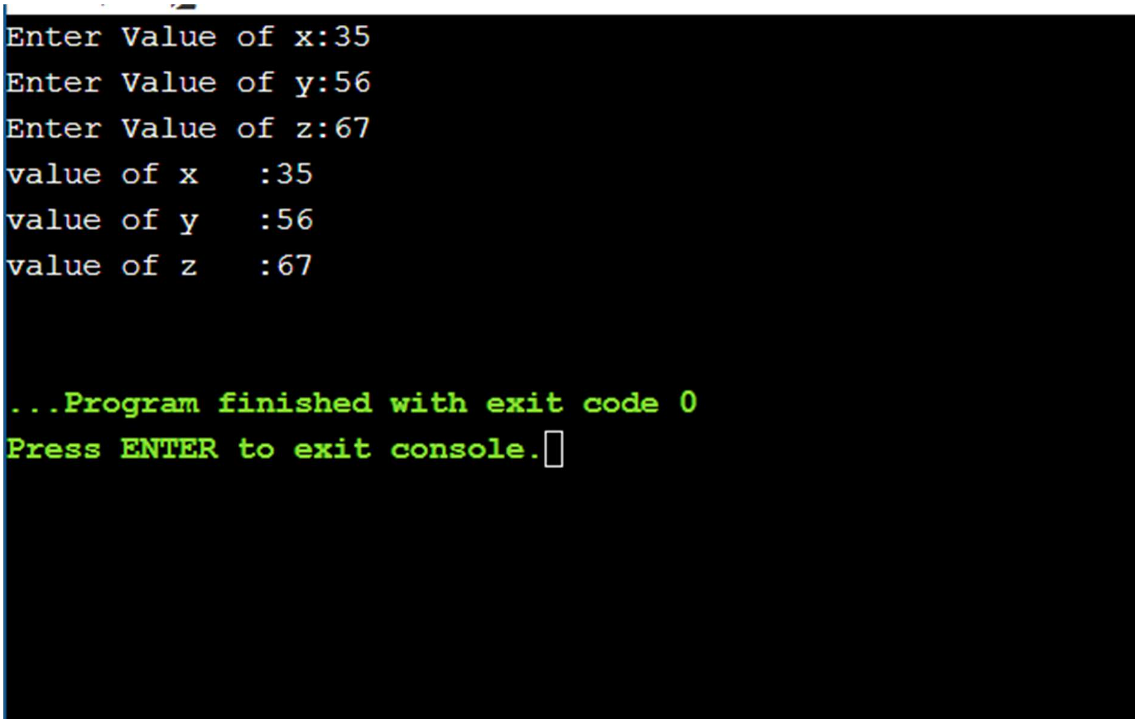
2). Create a class X with a protected variable x. A public method setX() is present to initialize the variable. Derive a class Y in public mode with a protected member variable y. A public method setY() is present to initialize the variable. Derive a class Z again in public mode from Y having a protected member z and a public method setZ(). Now write the main() to show the values of x, y and z from Z.

SOURCE CODE:

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

class X
{

protected:
    int x;

 public:
   void  setX()
  {
     cout<<"Enter Value of x:";
     cin>>x;
}
};

class Y: public X
{
protected:
   int y;
public:
    void setY()
{
cout<<"Enter Value of y:";
     cin>>y;;
}


};
class Z: public Y
{
protected:
   int z;


public:
    void setZ()
{
cout<<"Enter Value of z:";
     cin>>z;;
}
void show()
{
    cout<<"value of x   :"<<x<<endl;
    cout<<"value of y   :"<<y<<endl;
    cout<<"value of z   :"<<z<<endl;


}


};
```

```
int main()
{   Z oo;
    oo.setX();
    oo.setY();
    oo.setZ();
    oo.show();
           return 0;
}
```

OUTPUT:



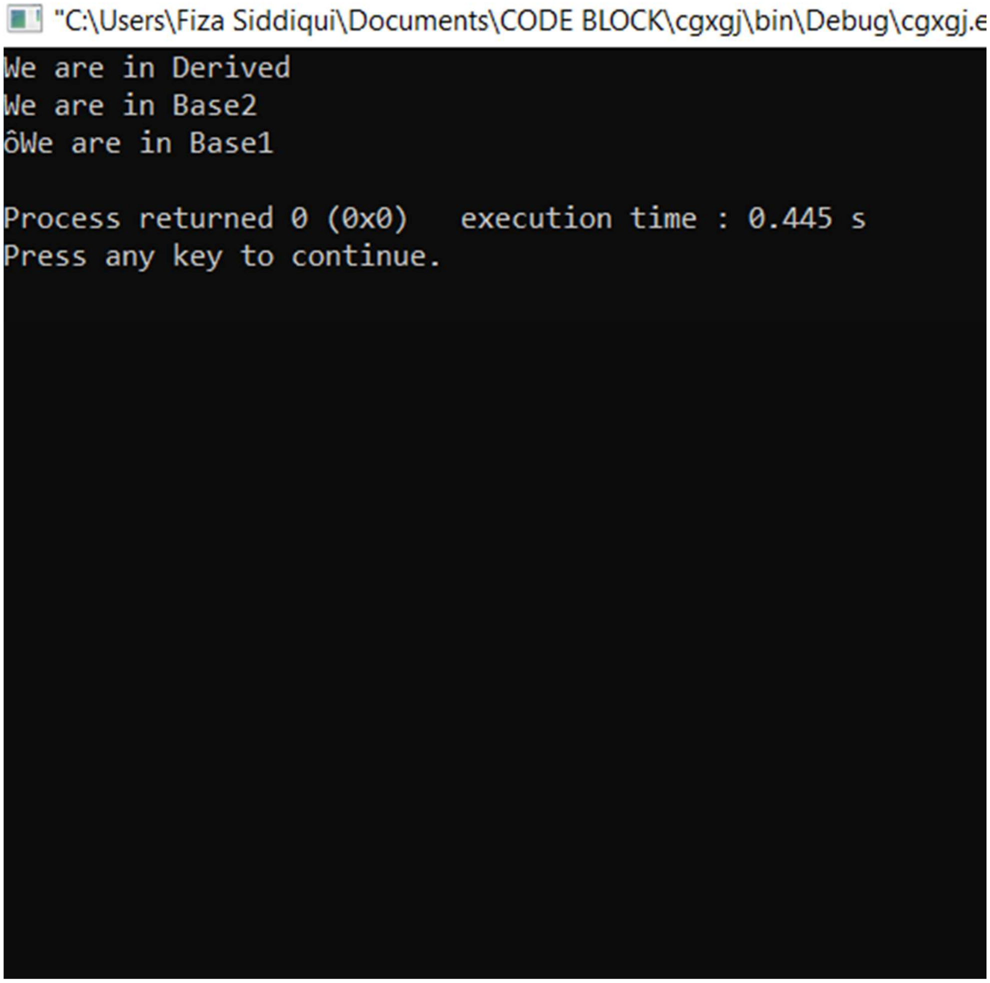
```
Enter Value of x:35
Enter Value of y:56
Enter Value of z:67
value of x   :35
value of y   :56
value of z   :67


...Program finished with exit code 0
Press ENTER to exit console.
```

3). Create a class Base1 with a public method display() having a output message such as “We are in Base1”. Derive a class Base2 with a public method display() with a similar message like “We are in Base2”. Now derive a class Derived in public way from both Base1 and Base2. Derived is also having a method display() having a message like “We are in Derived”. Now write a suitable main() so as to invoke the display() methods of Base1 and Base2 one after another with an instance of Derive.

SOURCE CODE:

```
#include <iostream>


using namespace std;


class Base1

{

 public:

   void  display()

  {

     cout<<"“We are in Base1"<<endl;

}


};


class Base2 : public Base1

{

public:

    void display()

{

cout<<"We are in Base2"<<endl;
}
};
class Derived :public Base2, public Base1
{
public:
   void display()
   {

    cout<<"We are in Derived"<<endl;
  }
};

int main()
{   Derived ob ;
        ob.display();
        ob.Base2::display();
        Base2 obb;
        obb.Base1::display();
        return 0;
}
```

OUTPUT:


"C:\Users\Fiza Siddiqui\Documents\CODE BLOCK\cgxgj\bin\Debug\cgxgj.e

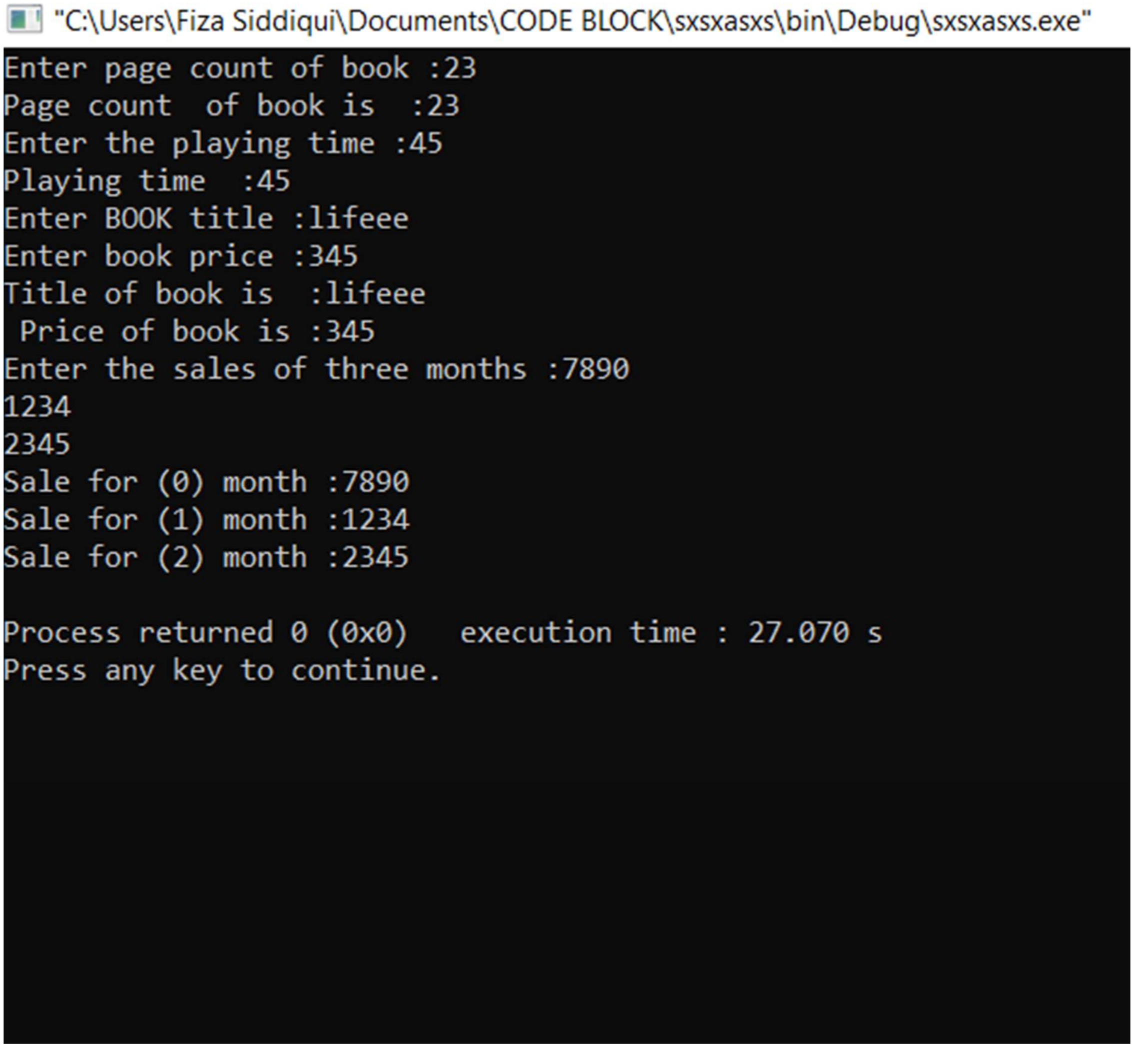
```
We are in Derived
We are in Base2
ôWe are in Base1

Process returned 0 (0x0)   execution time : 0.445 s
Press any key to continue.
```

4). Refer to the assignment 1. Add a base class Sales that holds an array of three floats so that it can record the dollar sales of a particular publication for the last three months. Include a getData() function to get three sales amounts from the user, and a putData() function to display the sales figures. Alter the Book and Tape classes so they are derived from both publications and sales. An object of class Book or Tape should input and output sales data along its other data. Write a main() function to create a book objectand a tape object and exercise input/output capabilities. 5)Assume that a bank maintains two kinds of accounts for custo

SOURCE CODE:

```
#include <iostream>


using namespace std;



class Publication
{


public:
    char title[20] ;
    float price ;


 public:


  void  getData()
  {
   cout<<"Enter BOOK title :";
     cin>>title;
   cout<<"Enter book price :";
     cin>>price;
 }
  void putData()
  {
     cout<<"Title of book is  :"<<title<<endl;
     cout<<  " Price of book is :"<<price<<endl;
  }
};

class Sales
{
   public :
   float sales[3];
   public:
     void   getData()
     {
        cout<<"Enter the sales of three months :";
        for(int i =0 ; i<3 ; i++)
        {
           cin>>sales[i];
        }
     }
     void putData()
     {
        for (int i =0 ;i<3 ; i++)
        {
              cout<<  "Sale for ("<<i<<") month :"<<sales[i]<<endl;
          }
      }
};
class book :public  Publication , public Sales
{
public:

 int pagecount ;

 public:
  void  getData()
  {
     cout<<"Enter page count of book :";
       cin>>pagecount;

  }
   void putData()
   {
       cout<<"Page count  of book is  :"<<pagecount<<endl;

   }
};
class Tape :public Publication, public Sales
{
    float time ;

 public:
  void  getData()
  {

    cout<<"Enter the playing time :";
      cin>>time;

  }
   void putData()
   {

      cout<<"Playing time  :"<<time<<endl;

   }
};

int  main()
{
   book ob ;
   Tape tt;
         ob.getData();
         ob.putData();
         tt.getData();
         tt.putData();
         ob.Publication::getData();
         ob.Publication::putData();
   ob.Sales::getData();
         ob.Sales::putData();
         return 0;

}
```

OUTPUT



"C:\Users\Fiza Siddiqui\Documents\CODE BLOCK\sxsxasxs\bin\Debug\sxsxasxs.exe"

```
Enter page count of book :23
Page count  of book is  :23
Enter the playing time :45
Playing time  :45
Enter BOOK title :lifeee
Enter book price :345
Title of book is  :lifeee
 Price of book is :345
Enter the sales of three months :7890
1234
2345
Sale for (0) month :7890
Sale for (1) month :1234
Sale for (2) month :2345

Process returned 0 (0x0)   execution time : 27.070 s
Press any key to continue.
```

5). 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 cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes

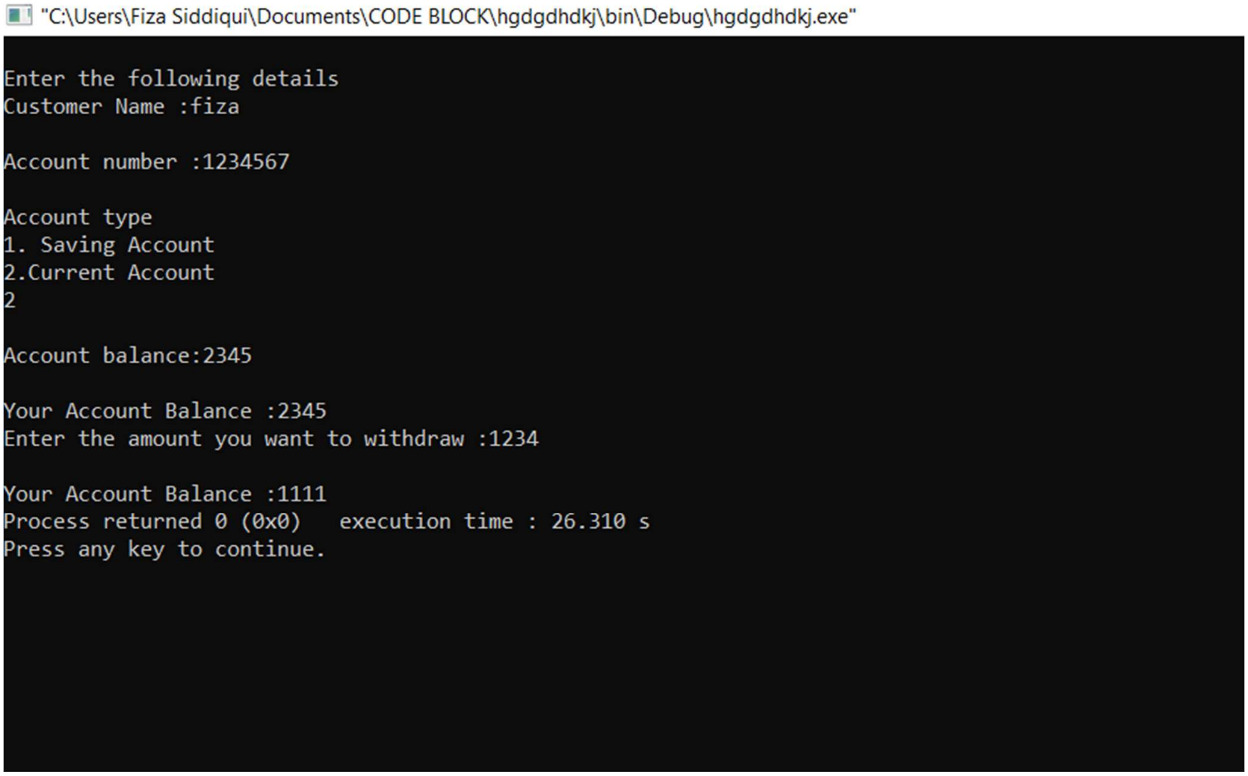
SOURCE CODE:

```
#include <iostream>

using namespace std;
//class account declaration and definition
class account
{
public:
char name[30];
int acc_num,acc_type;
int balance;
int amount;
void getData()
{
cout<<"\nEnter the following details\nCustomer Name :";
cin>>name;
cout<<"\nAccount number :";
cin>>acc_num;
cout<<"\nAccount type\n1. Saving Account\n2.Current Account\n";
```

```
cin>>acc_type;
cout<<"\nAccount balance:";
cin>>balance;
}
void display()
{
cout<<"\nYour Account Balance :"<<balance;
}
void withdraw()
{
cout<<"\nEnter the amount you want to withdraw :";
cin>>amount;
if(amount>balance)
cout<<"\nInsuficient balance";
else
balance=balance-amount;
display();
}
};
class cur_acct:public account
{
public:
void panelty()
{
if(balance<200 && acc_type==2)
{
balance=balance-20;
display();
}
```

```
}
};
class sav_acct:public account
{
public:
void interest()
{
int t;
cout<<"\nEnter time interval in year:";
cin>>t;
balance=balance*(1+2*t);
display();
}
};
int main()
{
account ac;
ac.getData();
ac.display();
ac.withdraw();
  return 0;
}
```

OUTPUT :


"C:\Users\Fiza Siddiqui\Documents\CODE BLOCK\hgdgdhdkj\bin\Debug\hgdgdhdkj.exe"

```
Enter the following details
Customer Name :fiza

Account number :1234567

Account type
1. Saving Account
2.Current Account
2

Account balance:2345

Your Account Balance :2345
Enter the amount you want to withdraw :1234

Your Account Balance :1111
Process returned 0 (0x0)   execution time : 26.310 s
Press any key to continue.
```

6). Consider the following class network. The class master derives information from both account and admin classes which in turn derive information from the class person. Define all the four classes and write a program to create, update and display the information contained in master objects

SOURCE CODE:

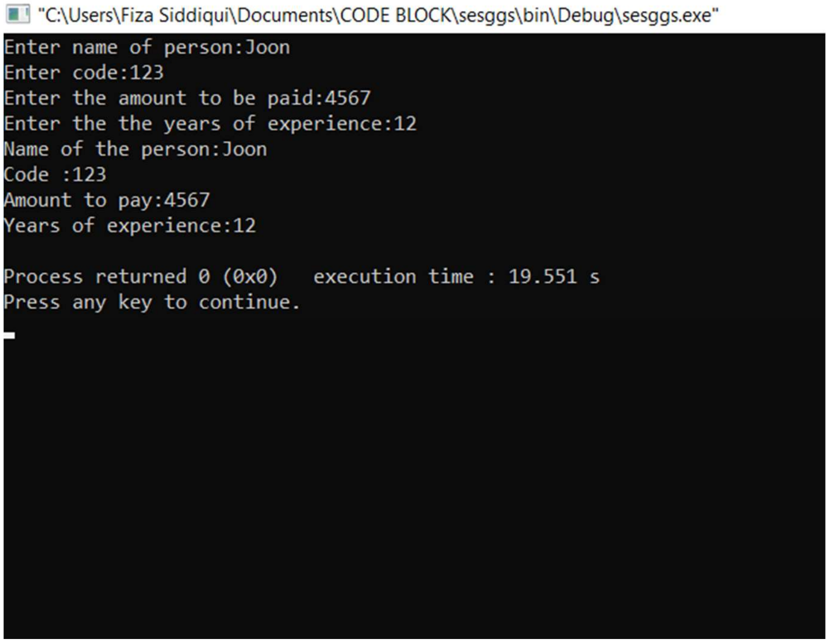
```
#include <iostream>


using namespace std;
//class account declaration and definition
class person
{
public :
     char name[20];
     int code ;
public :
    void update1()
    {
        cout<<"Enter name of person:";
       cin>>name;
     cout<<"Enter code:";
       cin>>code;
  }


};
class account :public  person
{
```

```
public:
    float pay ;
public:
void update2()
    {
        update1();
        cout<<"Enter the amount to be paid:";
       cin>>pay;
  }
   void display2()
   {
        cout<<"Name of the person:"<<name<<endl;
       cout<<  "Code :"<<code<<endl;

       cout<<"Amount to pay:"<<pay<<endl;
   }
};



class admin :public  person
{
public:
    int experience ;
public:
void update3()
    {
        cout<<"Enter the the years of experience:";
       cin>>experience;
  }
```

```
   void display3()
   {
       cout<<"Years of experience:"<<experience<<endl;
   }
};



class master  :public  account ,public admin
{
//public:
           //float pay;
           //int code,experience;
           //char name;

public:
     void get()
     {

         update2();
         update3();

         display2();
         display3();

     }



};
int main()
```

```
{
        master ob ;
        ob.get();


}
```

OUTPUT


"C:\Users\Fiza Siddiqui\Documents\CODE BLOCK\sesggs\bin\Debug\sesggs.exe"

```
Enter name of person:Joon
Enter code:123
Enter the amount to be paid:4567
Enter the the years of experience:12
Name of the person:Joon
Code :123
Amount to pay:4567
Years of experience:12

Process returned 0 (0x0)   execution time : 19.551 s
Press any key to continue.
```