**Muhammad Ali Jinnah University**

**Karachi, Pakistan**



**OBJECT ORIENTED PROGRAMMING**

**PROJECT NAME: BANK MANAGEMENT SYSTEM**

**Name: Syed Mustafa Hassan**

**Student ID: FA21-BSCS-0030**

**Section: AM**

**Batch: FA21**

***SUBMITTED TO: SIR AWAIS AHMED && SIR AJIAZ BUJATI***

**WHAT IS A BANK ACCOUNT?**

A bank account is a financial account maintained by a bank or other financial institution in which the financial transactions between the bank and a customer are recorded. Each financial institution sets the terms and conditions for each type of account it offers, which are classified in commonly understood types, such as deposit accounts, credit card accounts, current accounts, loan accounts or many other types of account. A customer may have more than one account. Once an account is opened, funds entrusted by the customer to the financial institution on deposit are recorded in the account designated by the customer. Funds can be withdrawn from loan loaders.

**WHAT IS A BANKING RECORD?**

The bank account record stores all bank account information you need to track and manage, such as account and routing numbers, current and minimum balances, bank details, adjustment categories, as well as any notes you want to associate with the bank account. From the bank account record, you can add alerts and adjustments, open the register, reconcile, and close accounts. We can use the bank record to keep track of our bank activity, reconciliations and how the bank account is performing. To view the bank record, go to Banking then click the required bank account.

**Aim:**

To develop a C++ program for Bank Management System using Object Oriented Programming concepts such as Class and Object, Switch case, File Handling.

**Objective:**

* To explore principles of object oriented programming.
* To use the object oriented paradigm in program design.
* To lay foundation to advanced programming.

**Assumption**:

We are finding interest on basis of per day concept to show the change in interest effectively. We reduce the scale of finding interest to days (instead of months or years plans). We work on only two type of account – Saving and Current.

**Approach:**

We Involved File handling to make a record of the customer. We work on csv formatted file to store data. We involved string and manipulator function to get the desired output. We used concept of interest calculation in Bank. We involve functions to get the current time from the system.

**Header Files Involved Those Which Provided By Compiler:**

**#include <iostream>**  -> It is the predefined library function used for input and output also called as header files. iostream is the header file which contains all the functions of program like cout, cin etc.

**#include <fstream>**  -> File streams include two member functions specifically designed to read and write binary data sequentially: write and read. The first one ( write ) is a member function of ostream (inherited by ofstream ). And read is a member function of istream (inherited by ifstream ). Objects of class fstream have both.

**#include <string>**  -> This header file defines several functions to manipulate strings **and** arrays.

**#include <iomanip>** -> The header <iomanip> is part of the Input/output library of the C++ Standard Library. It defines the manipulator functions resetiosflags() , setiosflags() , setbase() , setfill() , setprecision() , and setw() .

**#include <cstdio>** -> Input and Output operations can also be performed in C++ using the C Standard Input and Output Library. This library uses what are called streams to operate with physical devices such as keyboards, printers, terminals or with any other type of files supported by the system

**#include <conio.h>** -> conio.h is a C header file used mostly by MS-DOS compilers to provide console input/output.

**#include <time>** -> The time() function in C++ converts the given time since epoch to a calendar local time and then to a character representation. A call to time(time) is a combination of asctime() and localtime() functions, as asctime(localtime(time)) .

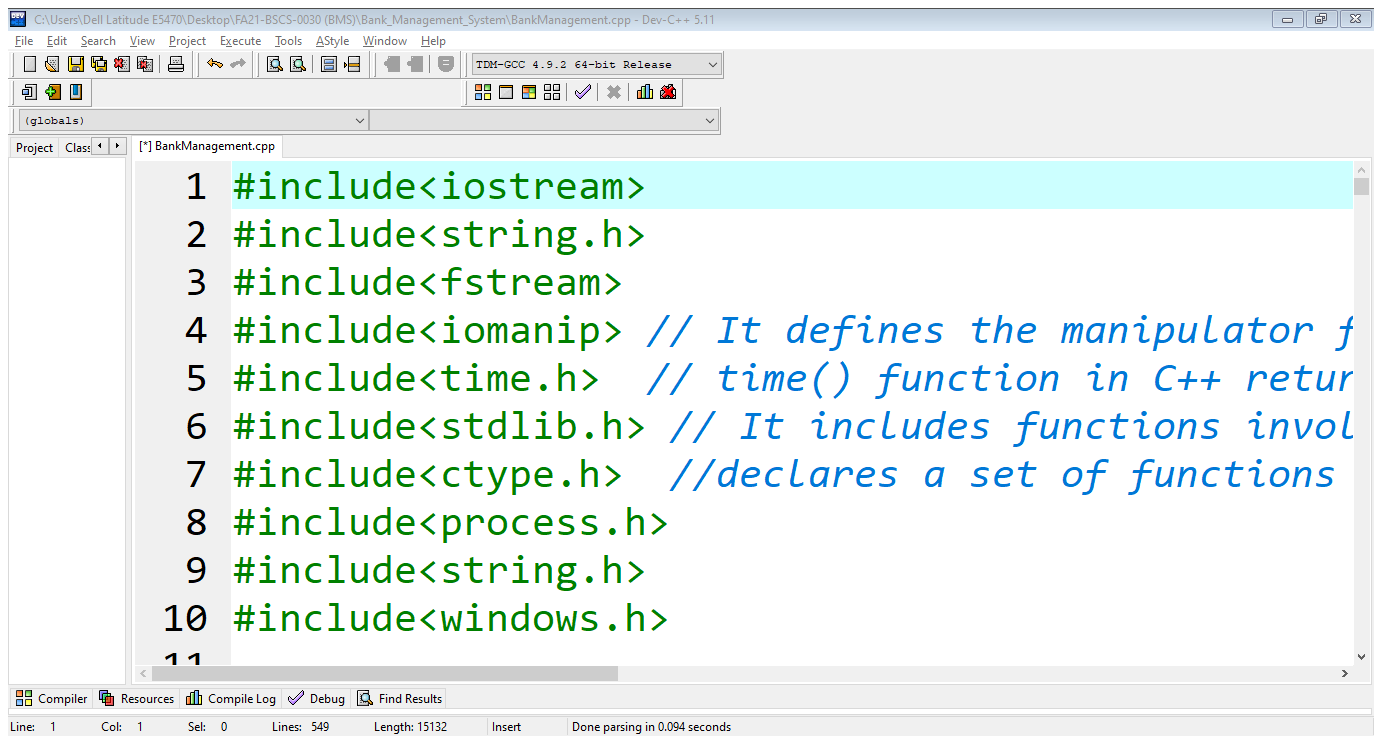
**#include <stdlib.h>** -> stdlib.h is the header of the general purpose standard library of C programming language which includes functions involving memory allocation, process control, conversions and others.

**Using namespace std:**

A namespace is a form of scope in C++ that holds its own definitions for variables, functions, etc. For example, both cout and cin, along with some useful tokens like endl , are defined inside of std for use.

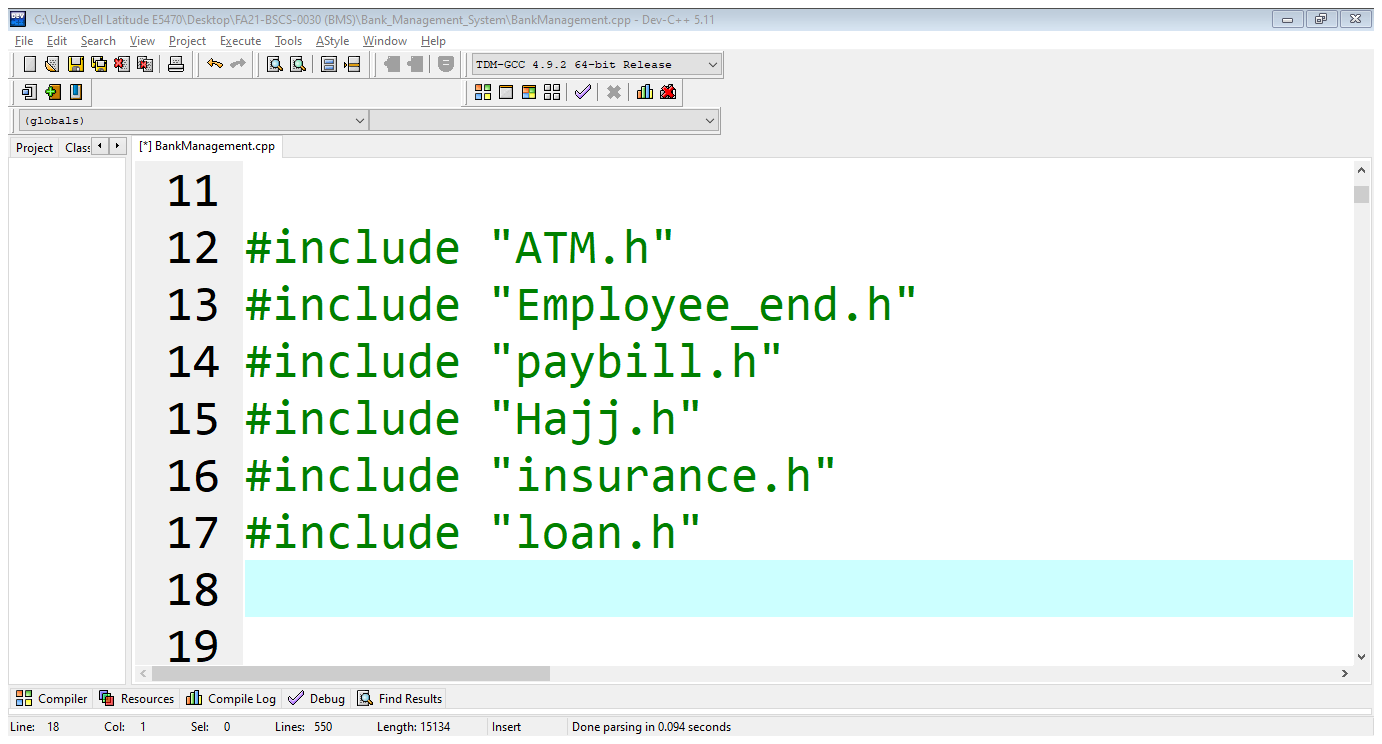
**Global Variable:**

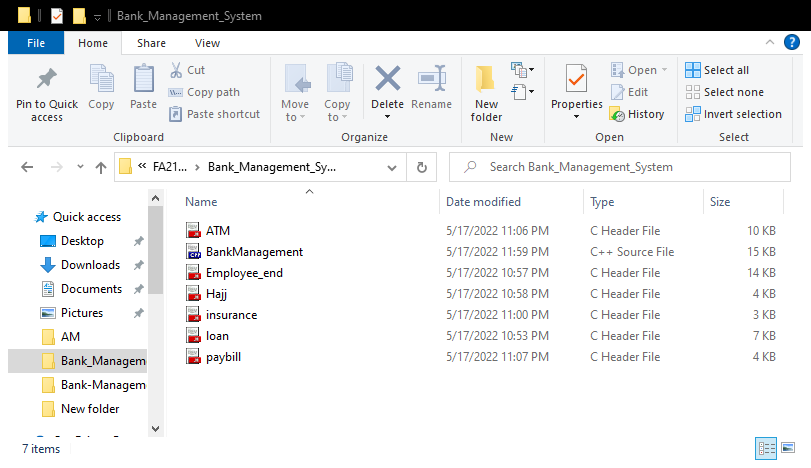
int main exit; -> variable used globally so that can be used for accepting a value for exit or main menu or for previous menu;

**Global Function:** Functions created globally so that can be accessed on any function.

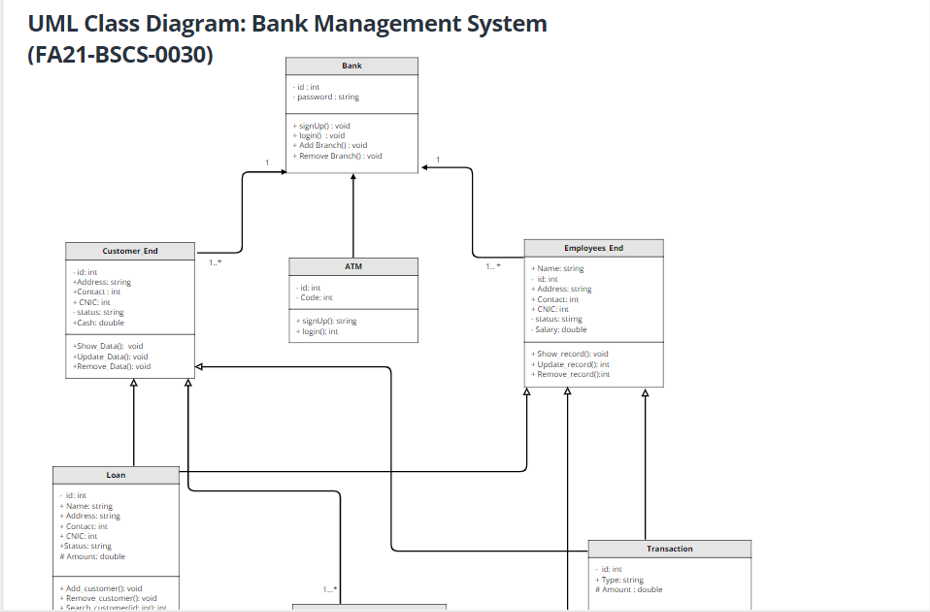
**Header Files Involved Those Which I Made:**

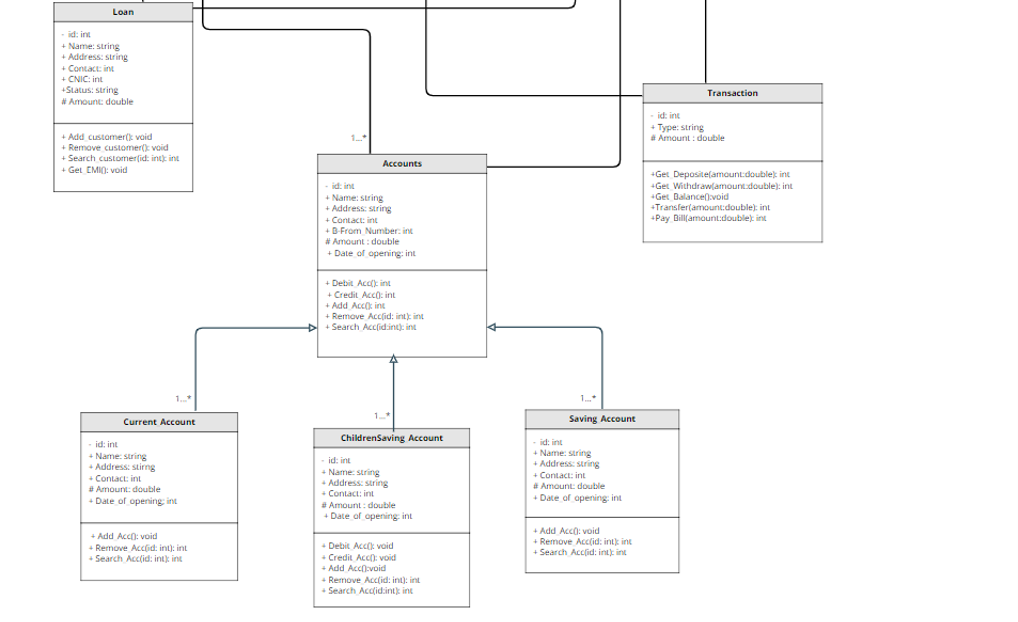
1. ATM
2. Employees\_end
3. Hajj
4. Insurance
5. Loan
6. Paybill





**UML Class Diagram:**

****

****

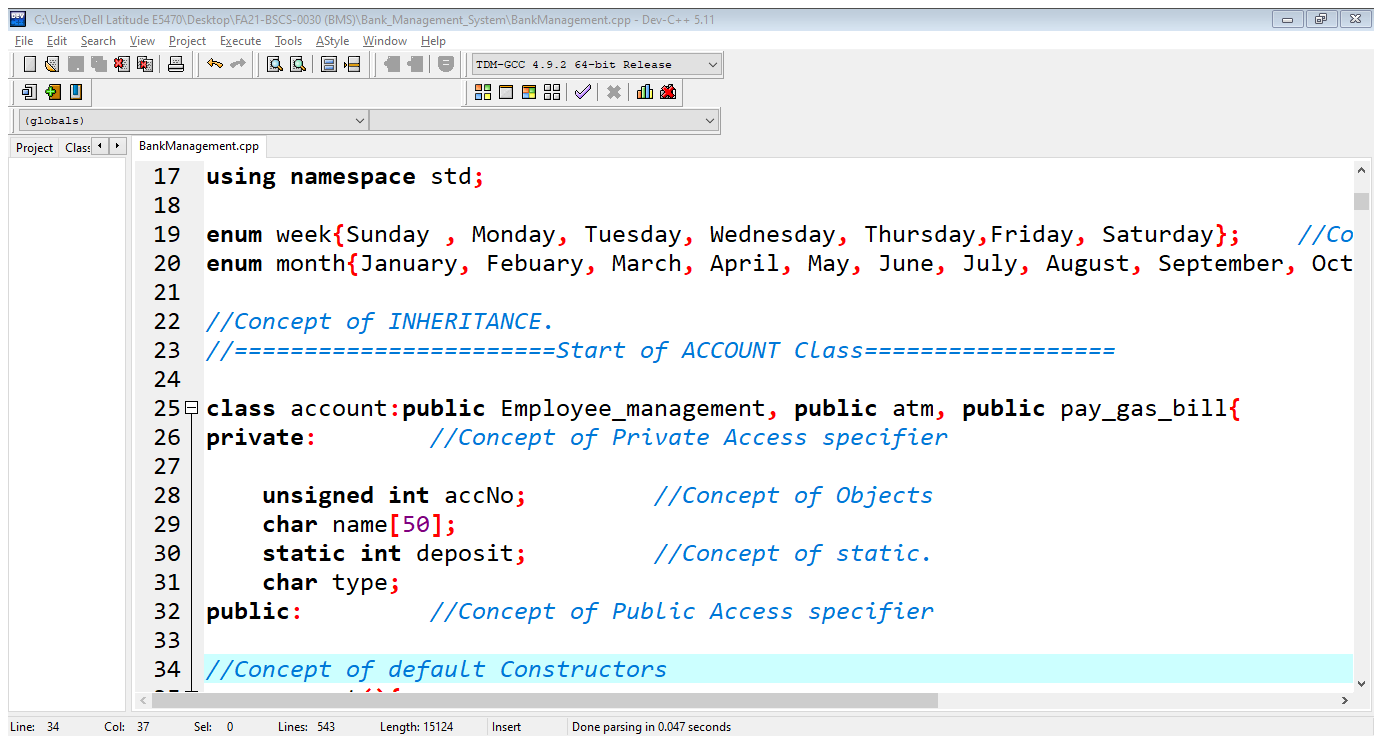
**OOP PLAIRS:**

1. **INHERETANCE:**

Inheritance is one of the core concepts of object-oriented programming (OOP) languages. It is **a mechanism where you can to derive a class from another class for a hierarchy of classes that share a set of attributes and methods**.

In this project I prefer Multiple Inheritance in which I make different classes to perform different task in different header files that inherited with “BankManagement.cpp” File which is the main file (Source File).

BankManagement.cpp

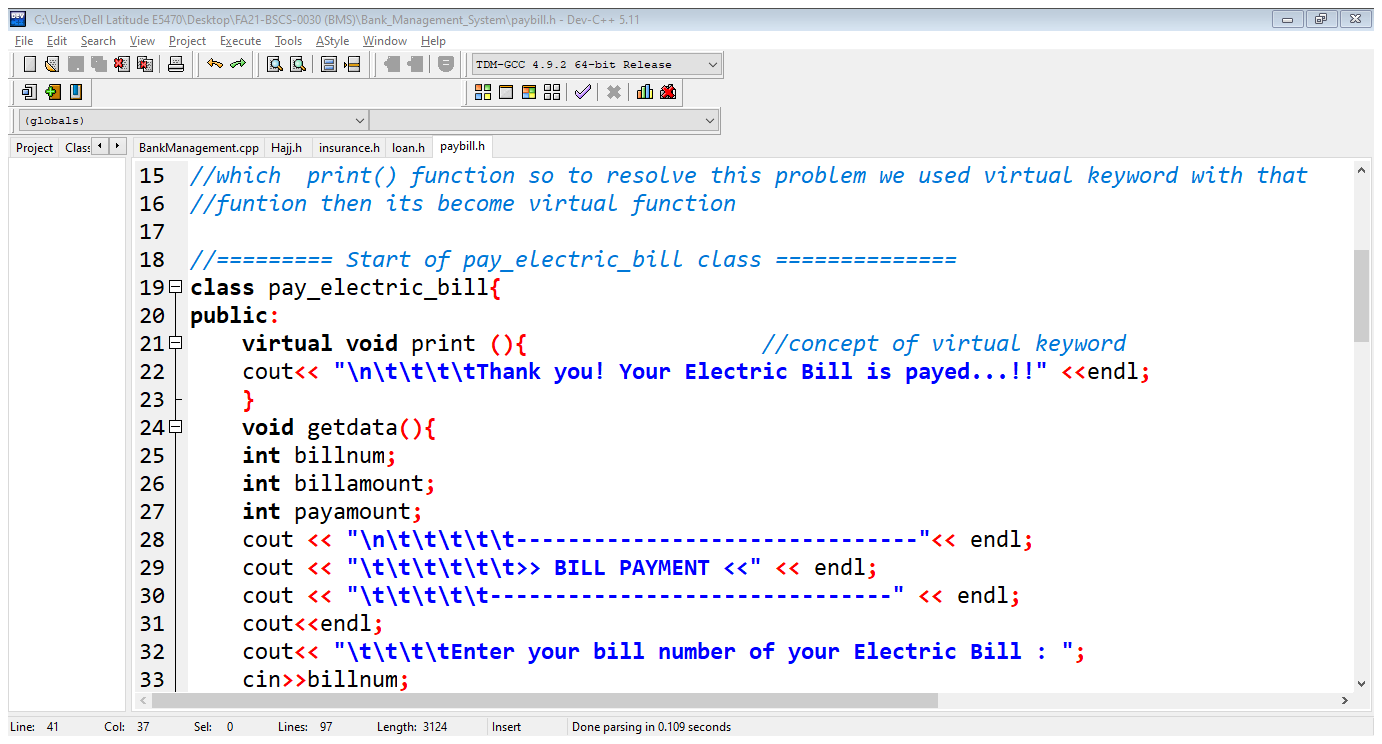


1. **POLYMORPHISM**

Polymorphism is one of the core concepts of object-oriented programming (OOP) and **describes situations in which something occurs in several different forms**. In computer science, it describes the concept that you can access objects of different types through the same interface.

In this project there are two class pay\_electric\_bill (base class) & pay\_gas\_bill (Derived class), in base class there is a function void print () this function is also in derived class with different work in calling process compiler confused to display which void print () function so to resolve this problem we used virtual keyword with that function then it’s become virtual function & here I perform POLYMORPHISM.

Paybill.h

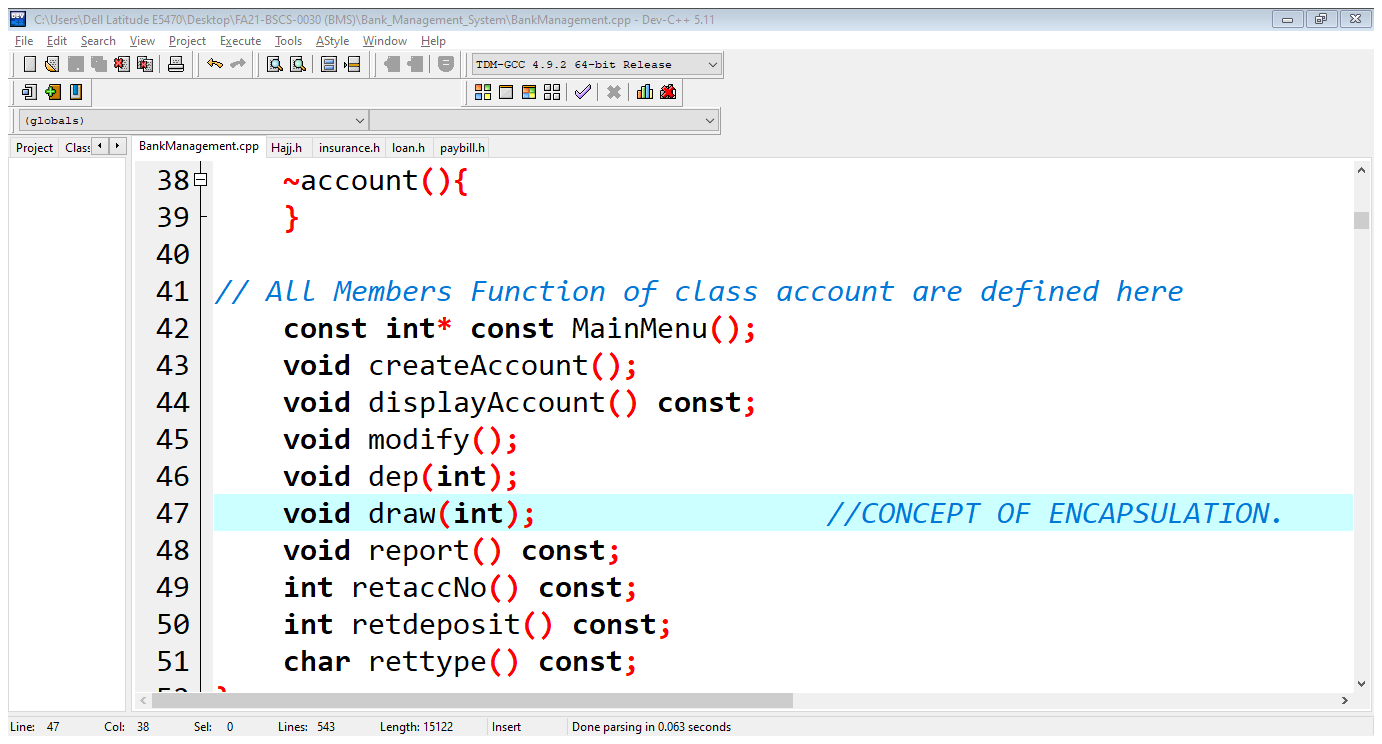


1. **ENCAPSULATION:**

The notion of encapsulation (or OOP Encapsulation) refers to **the bundling of data, along with the methods that operate on that data, into a single unit**. Many programming languages use encapsulation frequently in the form of classes.

In this project encapsulation is perform in each files.

BankManagement.cpp

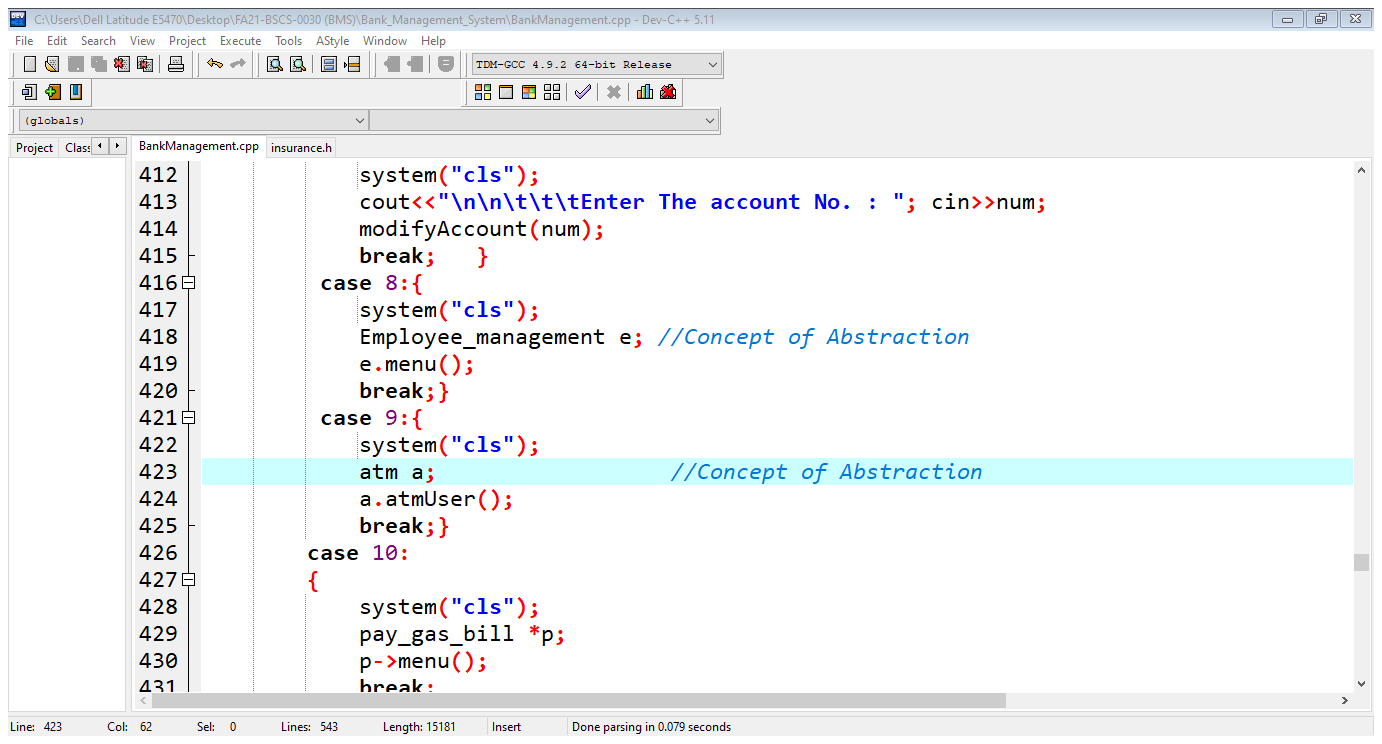


1. **ABSTRACTION:**

Abstraction is the concept of object-oriented programming that "shows" only essential attributes and "hides" unnecessary information. The main purpose of abstraction is hiding the unnecessary details from the users.

In this project abstraction is perform in each files.

BankManagement.cpp



**Essential Information required for the implementation of the program:**

**Bank:**

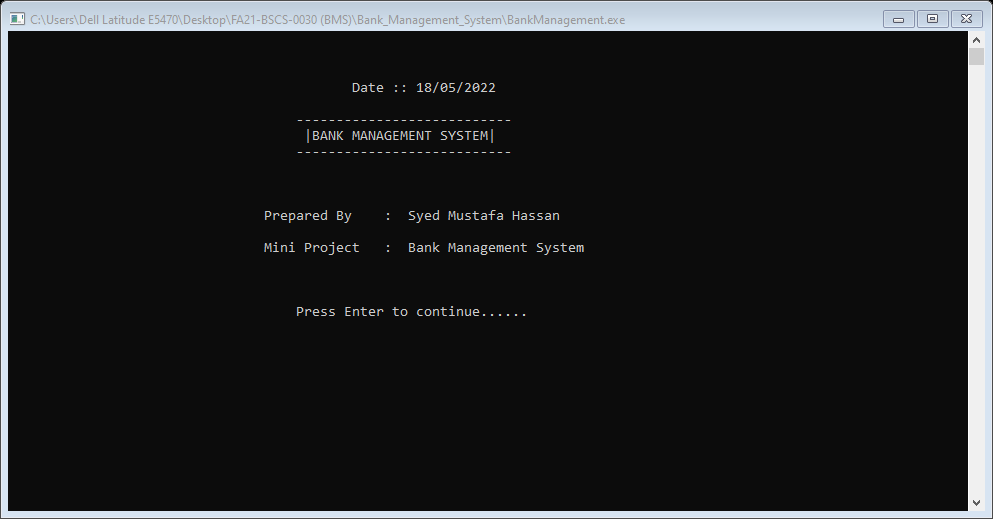
A bank is a financial institution licensed to receive deposits and make loans. Banks may also provide financial services such as wealth management, currency exchange, and safe deposit boxes.

**Project:**

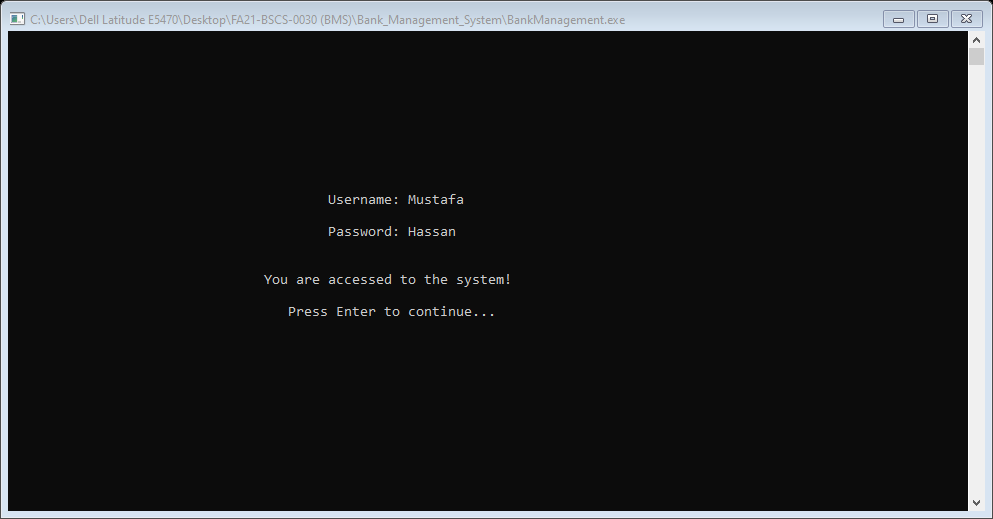
This project provides the following functionalities:

**1.** **Main Page:**

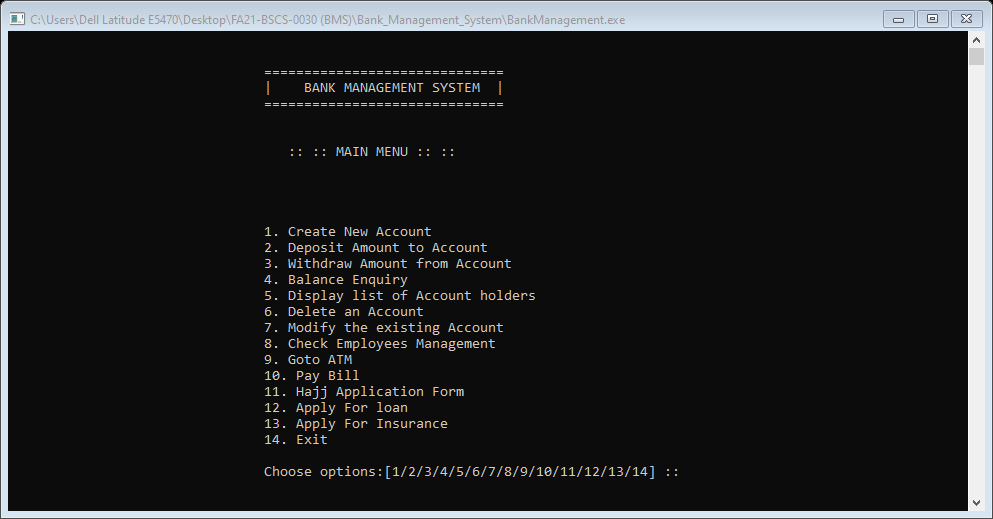
This project having the one of the most important and attractive feature that is first page which called main page which having this following look.



Then Admin supposed to enter username and password to access the other core feature of this project.

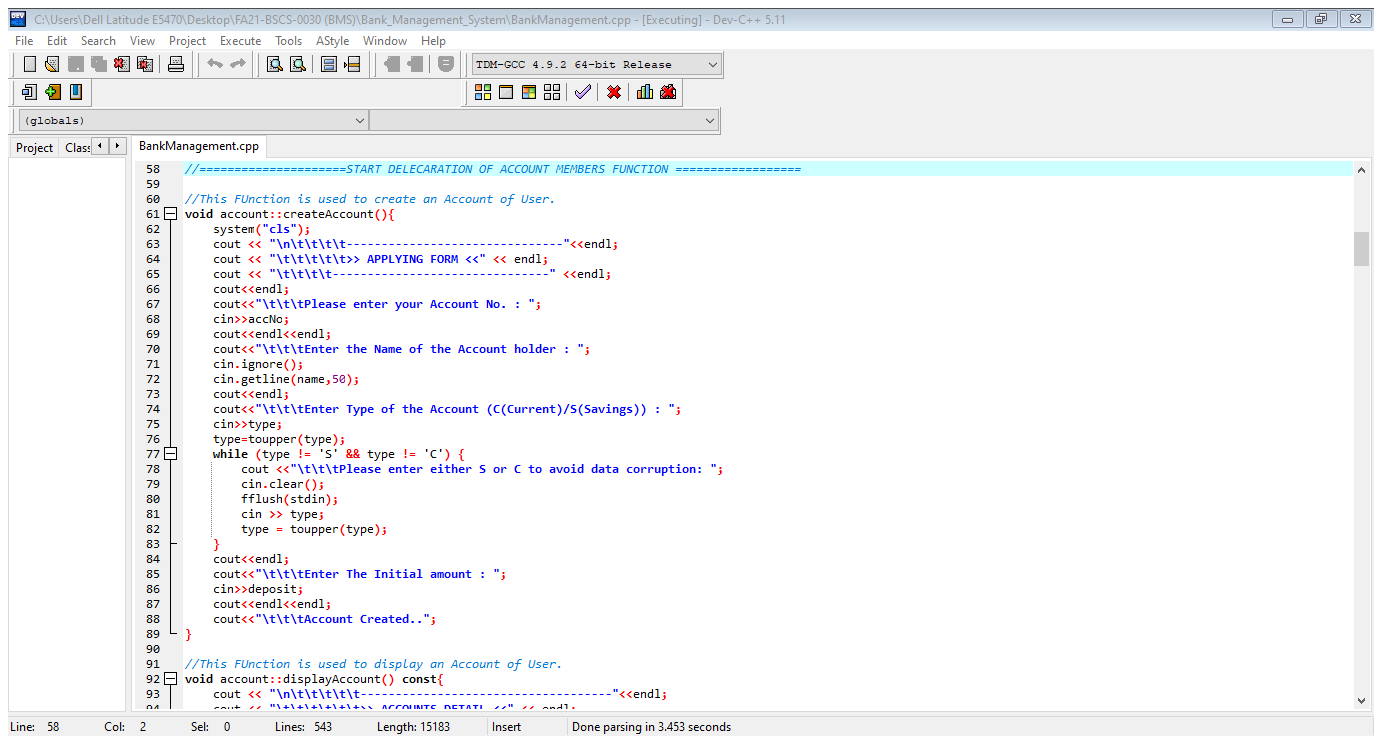


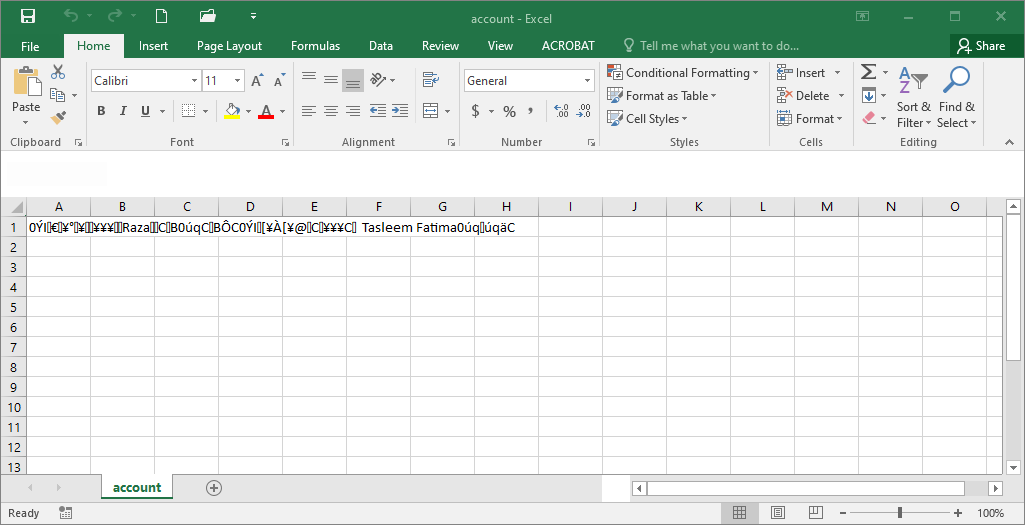
Then the core feature which the main menu of this program is display to the admin

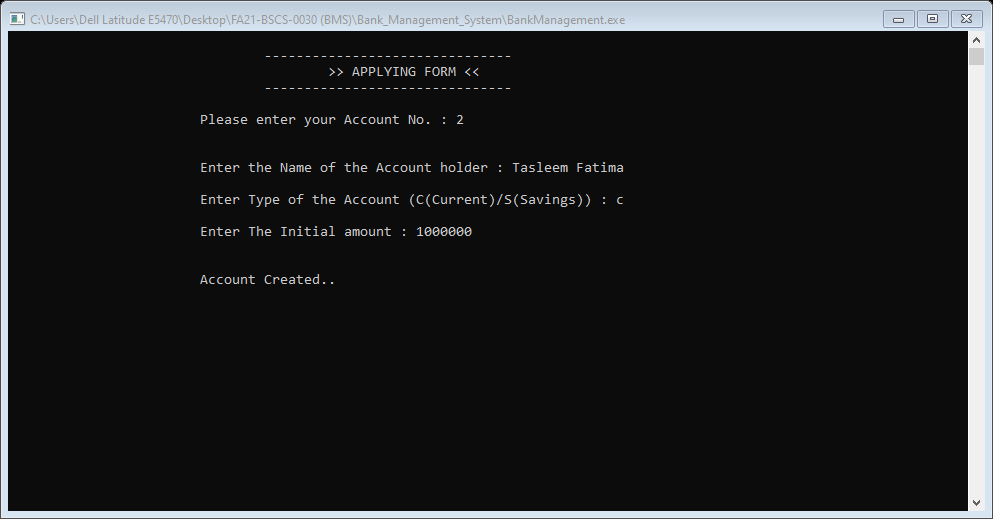


**2. Create new Account:**

This feature is one of the most important in my project in which the user can create his/her account by filling the simple form and the account is created.

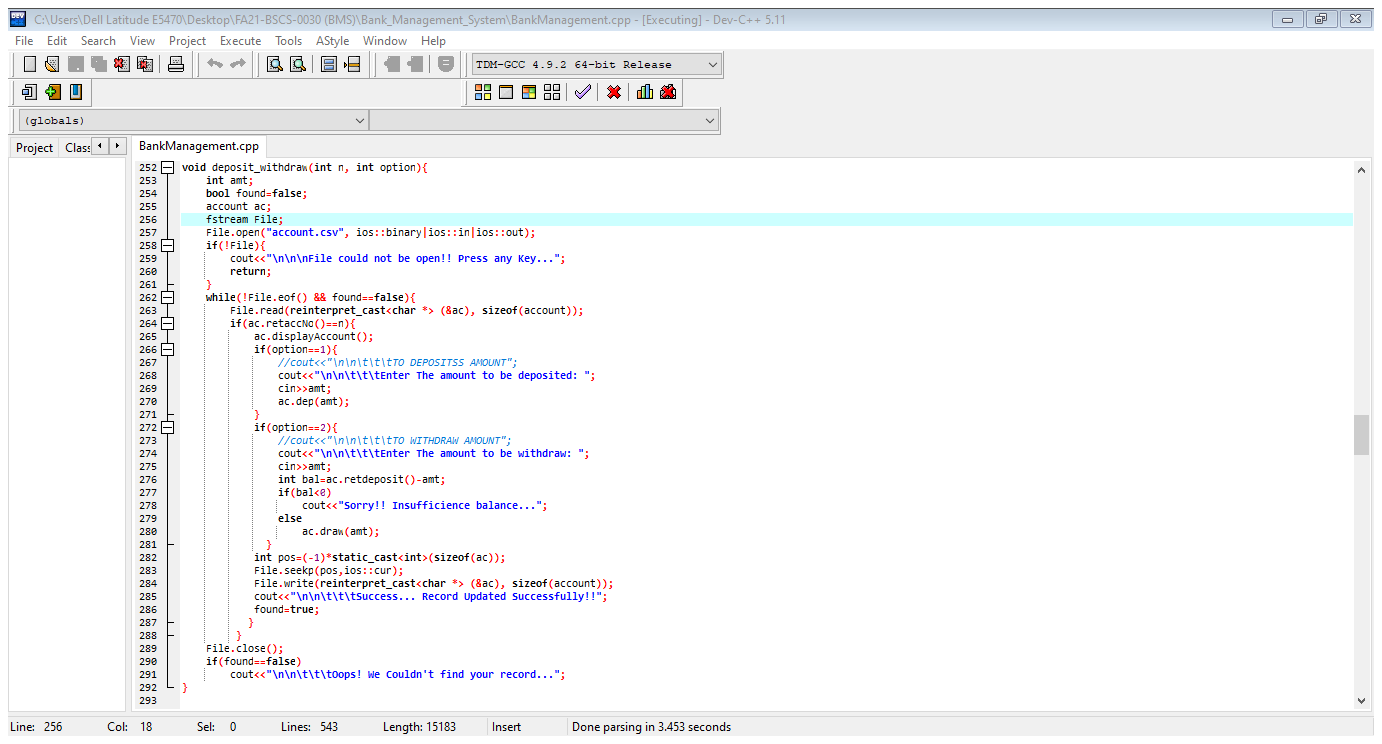


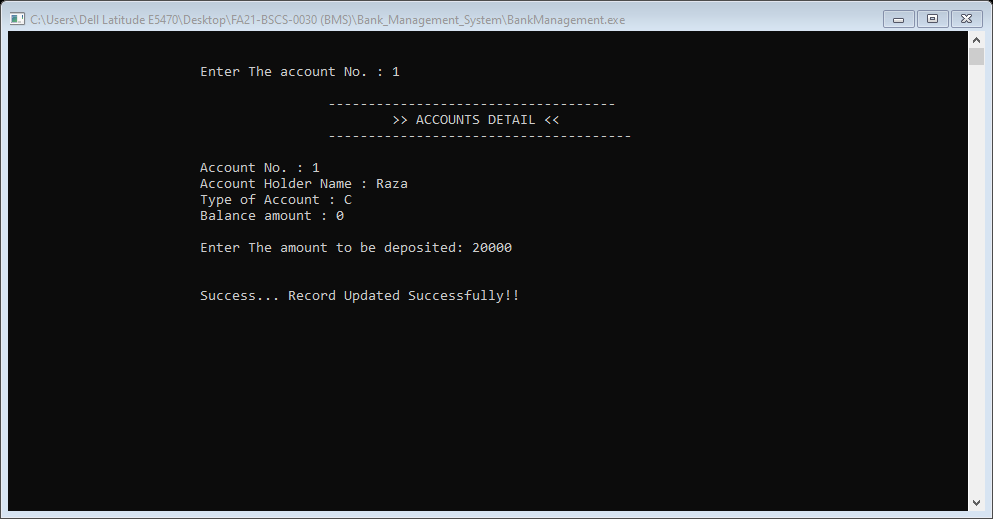




**3. Deposit amount to Account:**

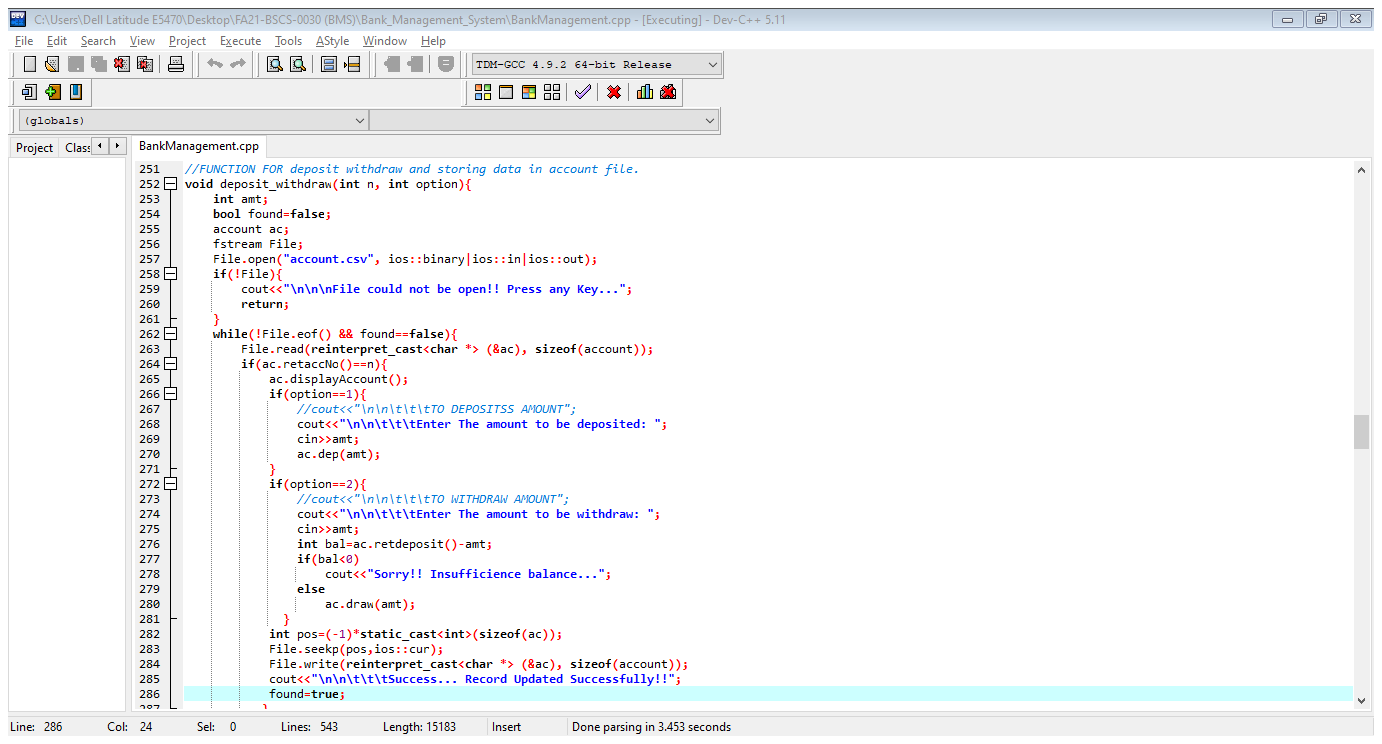
This feature is one of the most important in my project in which those users whose account is already created they can deposit amount.

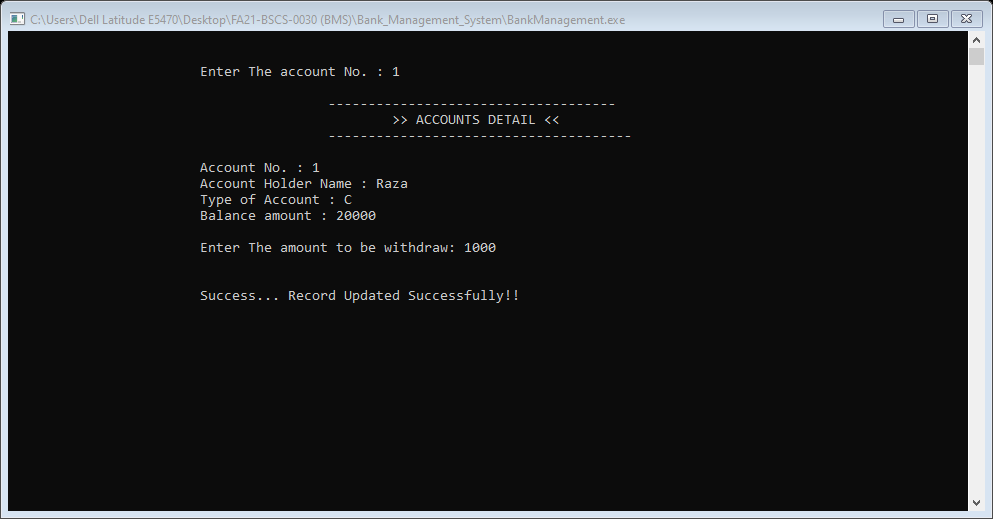




**4. Withdraw amount from Account:**

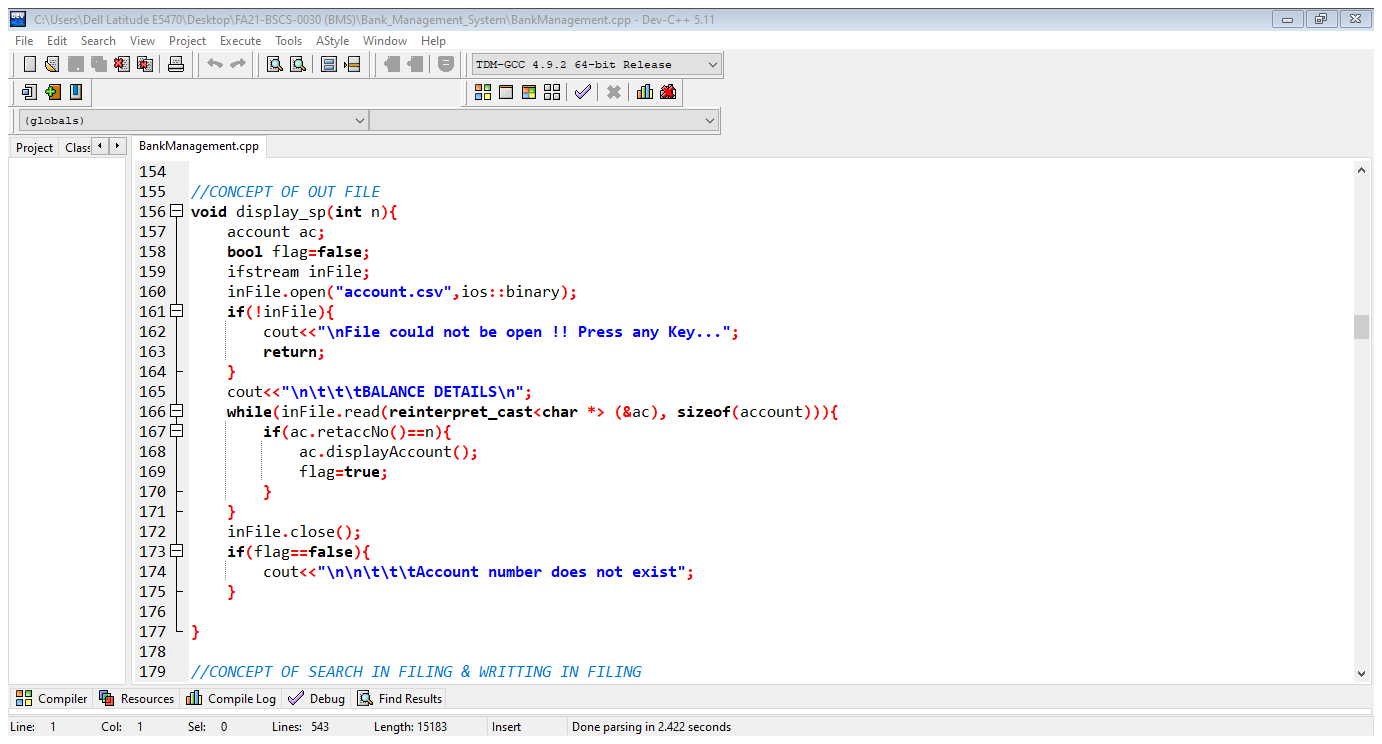
This feature is one of the most important in my project in which those users whose account is already created they can Withdraw amount in his/her account.

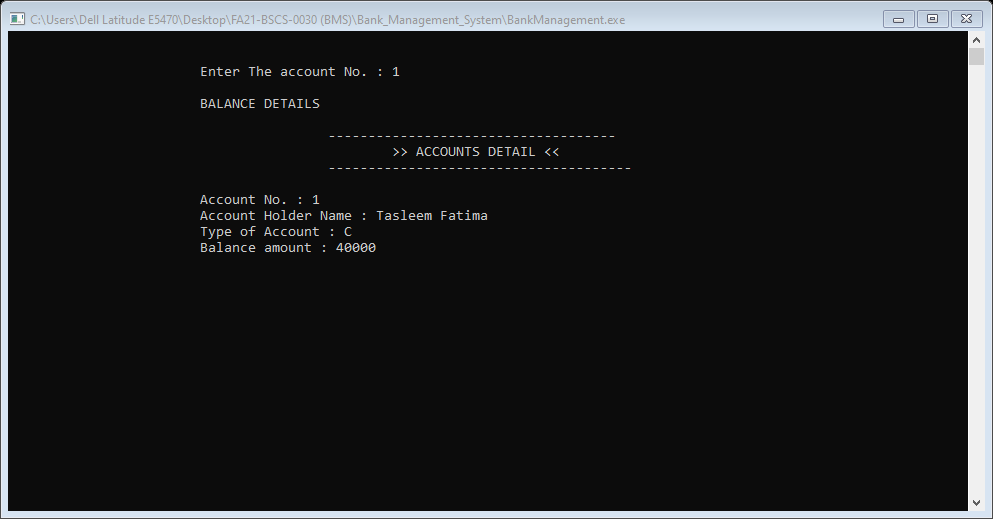




**5. Balance Enquiry:**

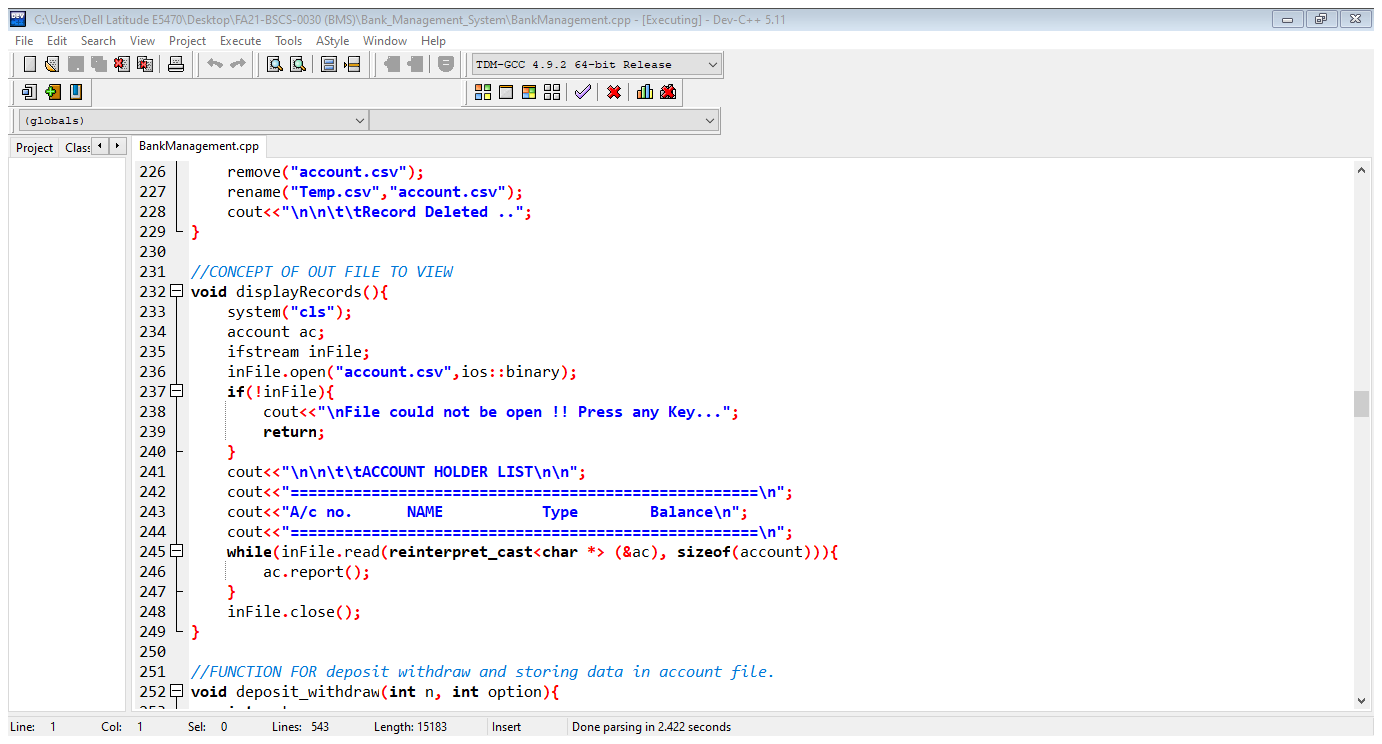
The other one of the most important feature in my project in which those users whose account is already created they can do the balance enquiry and can see the reaming amount of his/her account.

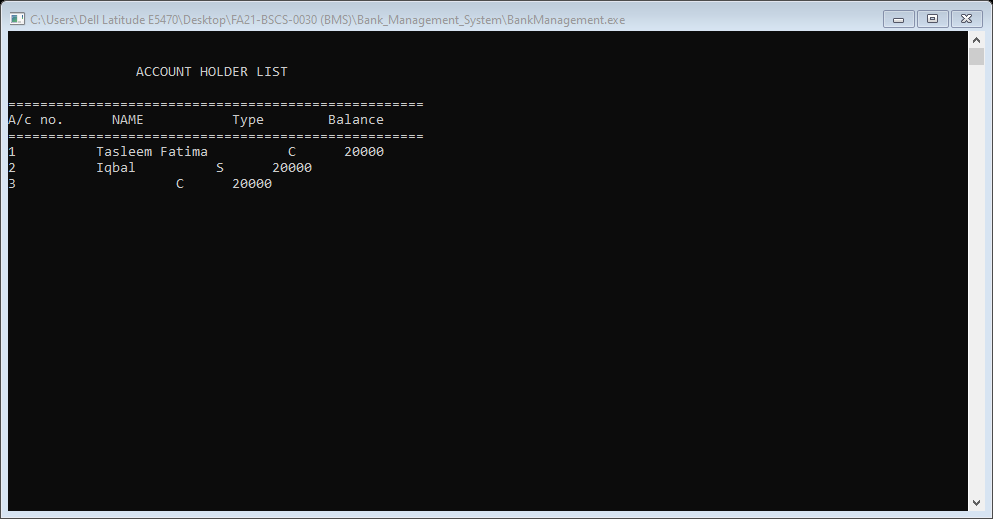




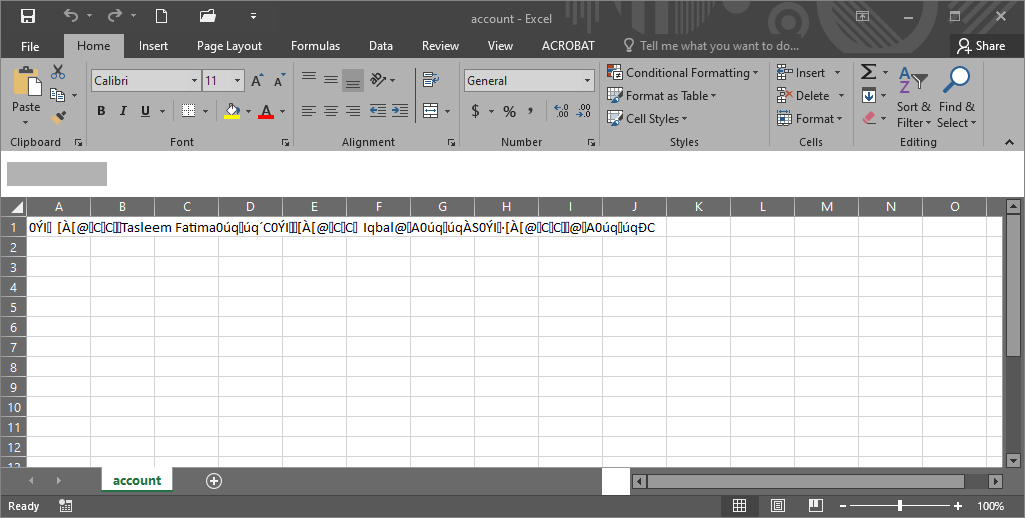
**6. Display list of Account Holders:**

This feature is one of the most important in my project in which those users whose account is already created the admin can able to see all list of the account holders.



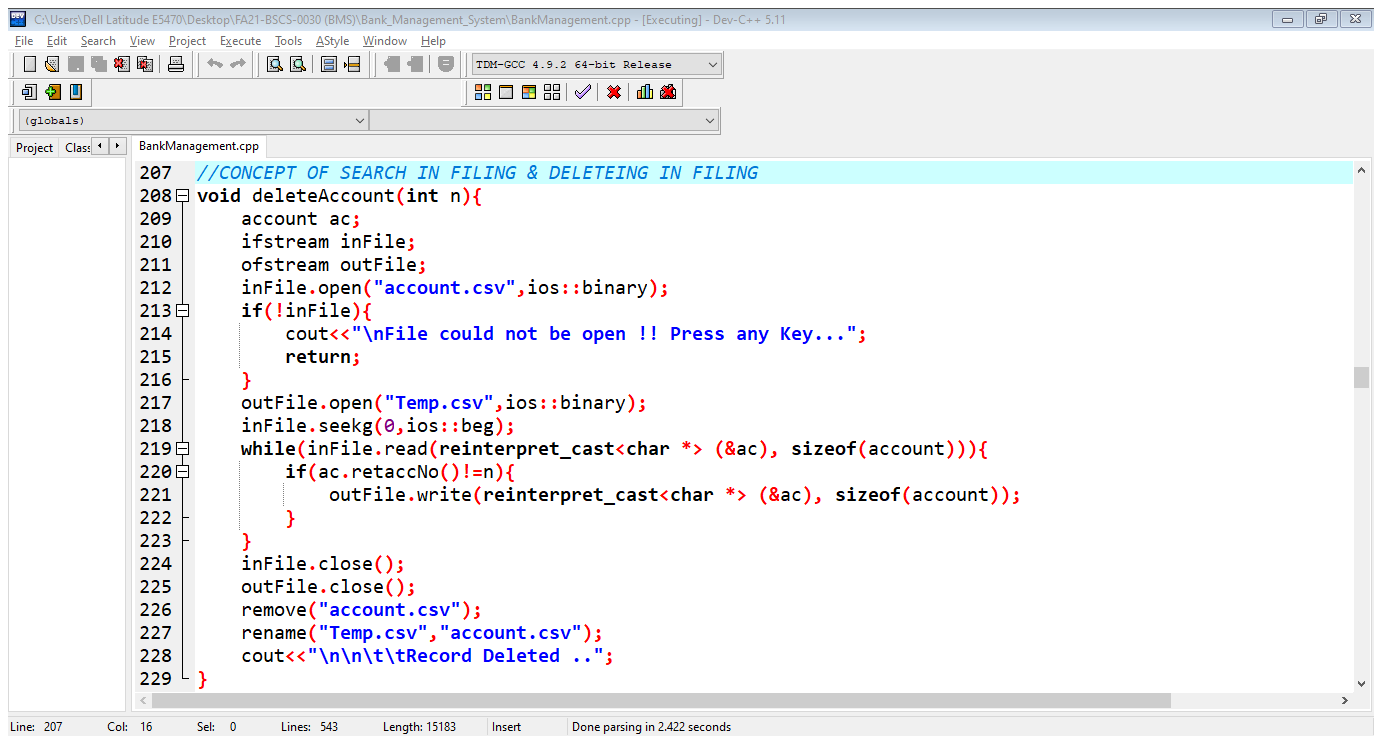


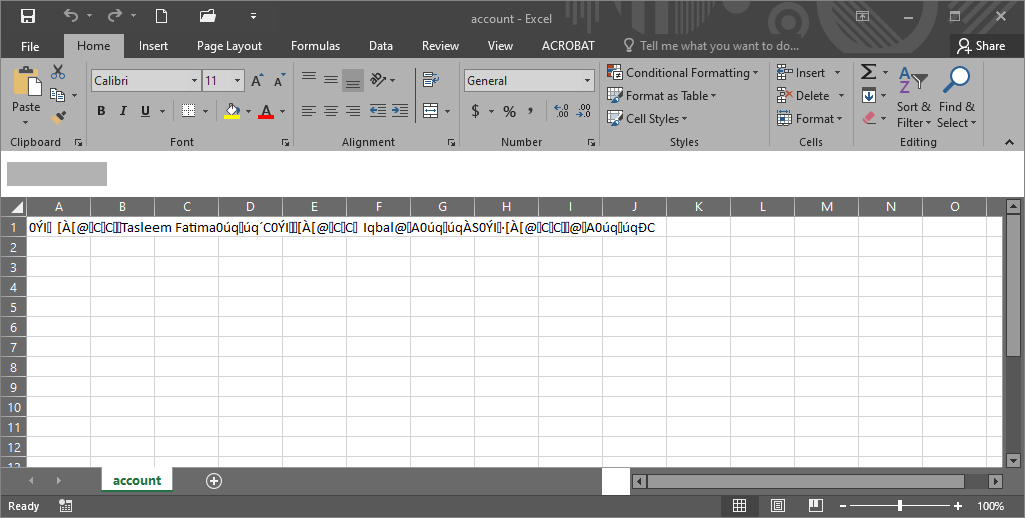
This all data is store in account.cvc file and form their it can show to admin.



**7. Delete an Account:**

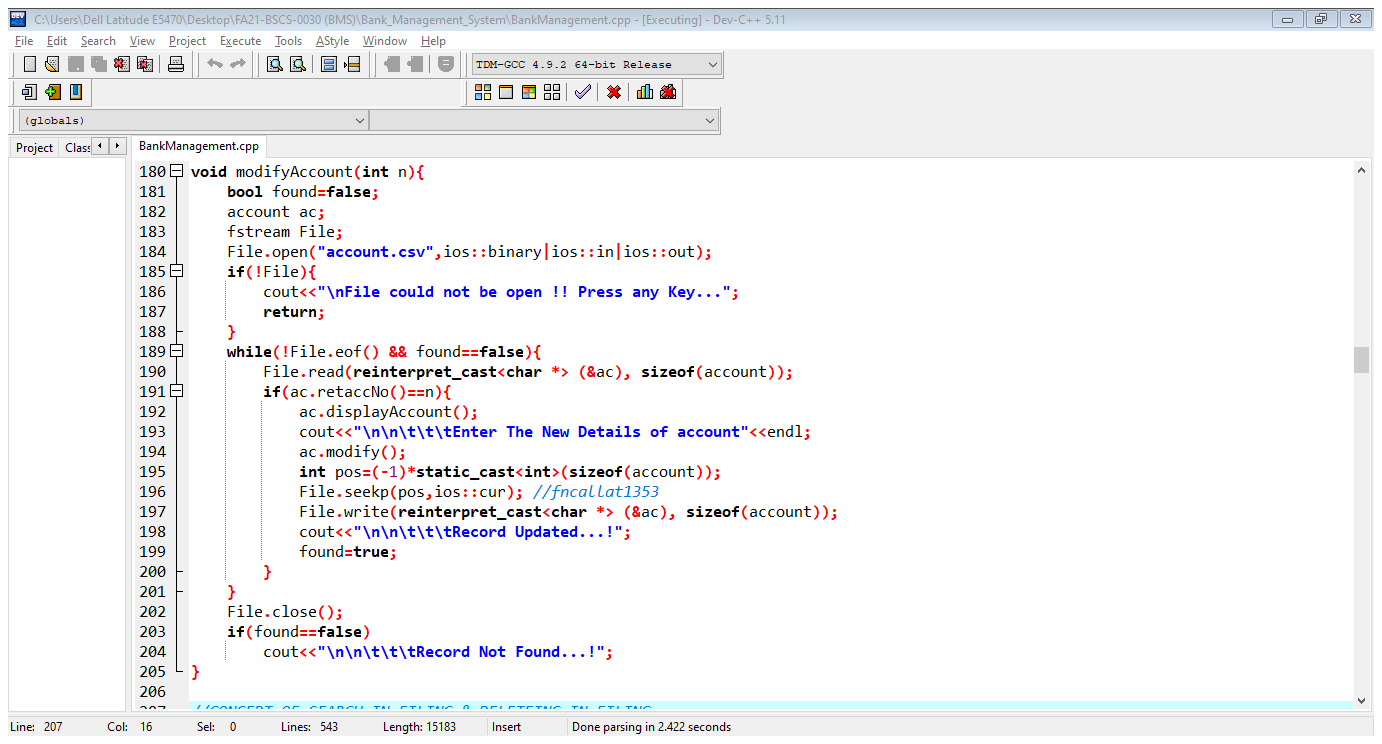
This feature is one of the most important in my project in which those users whose account is already created the admin can able to delete any account of the account holders.

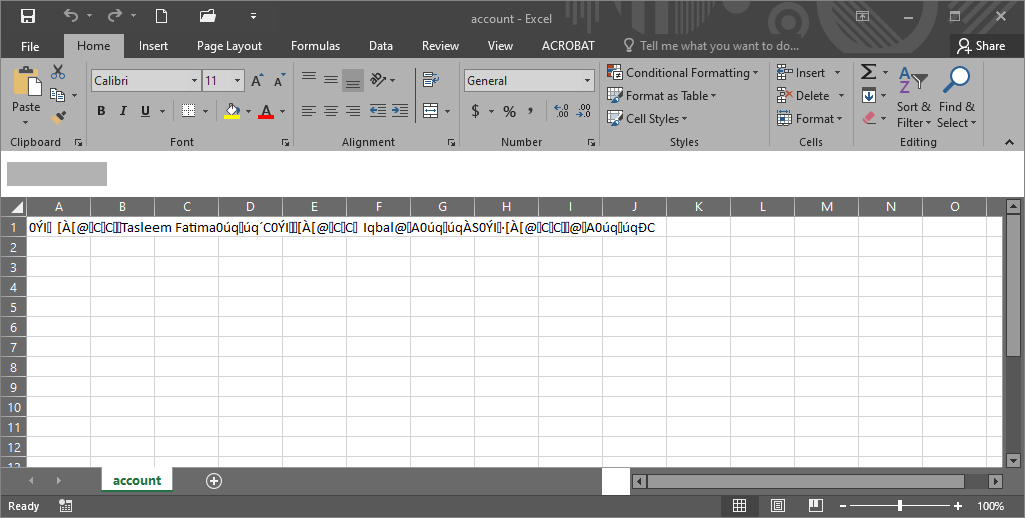


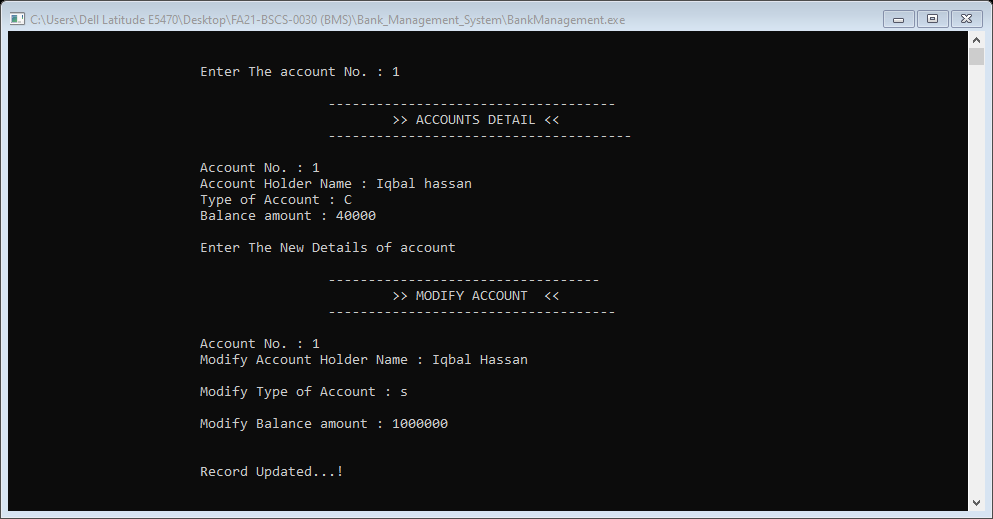


**8. Modify the existing Account:**

Modify the existing Account this feature is one of the most important in my project in which those users whose account is already created the admin can able to Modify any account of the account holders.

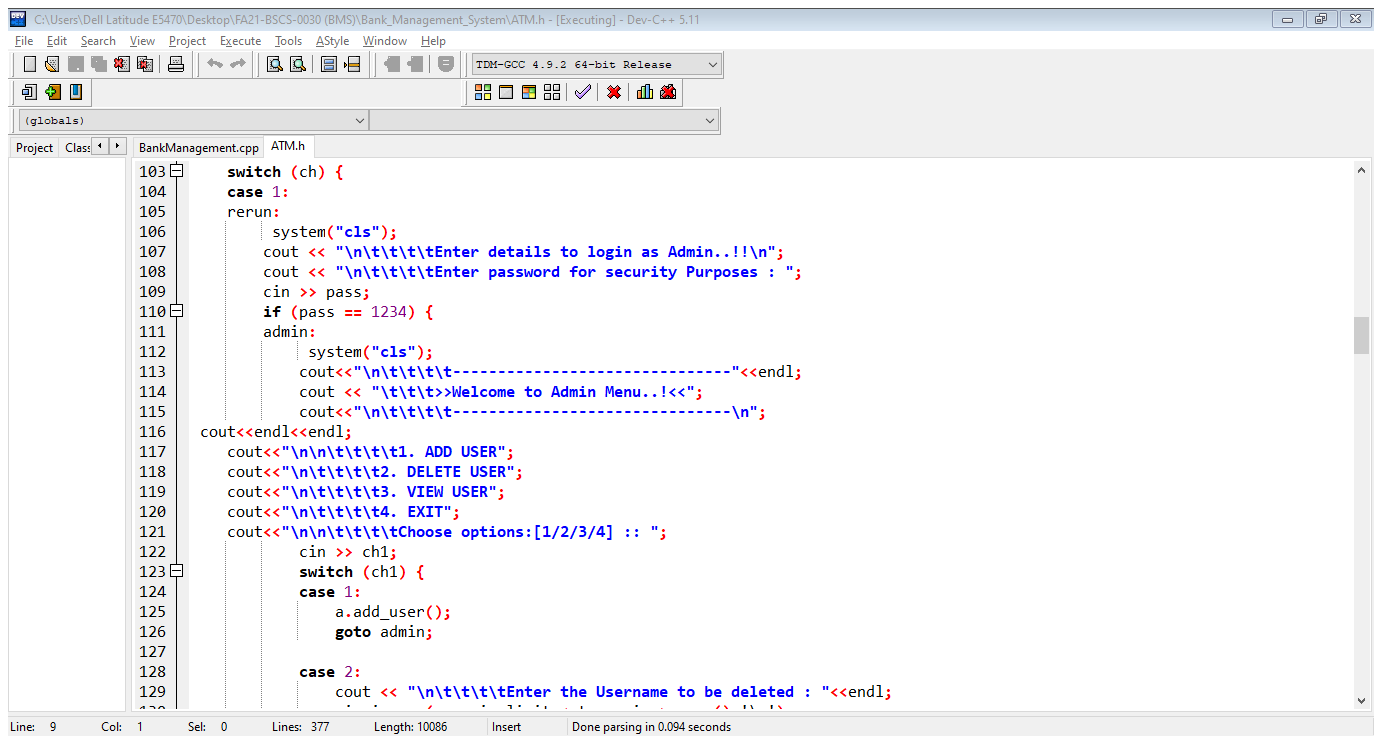


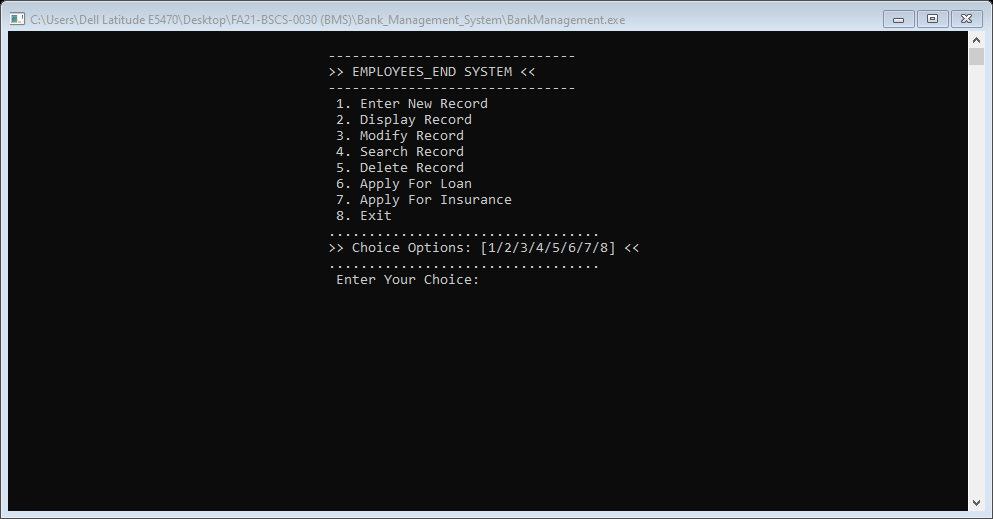


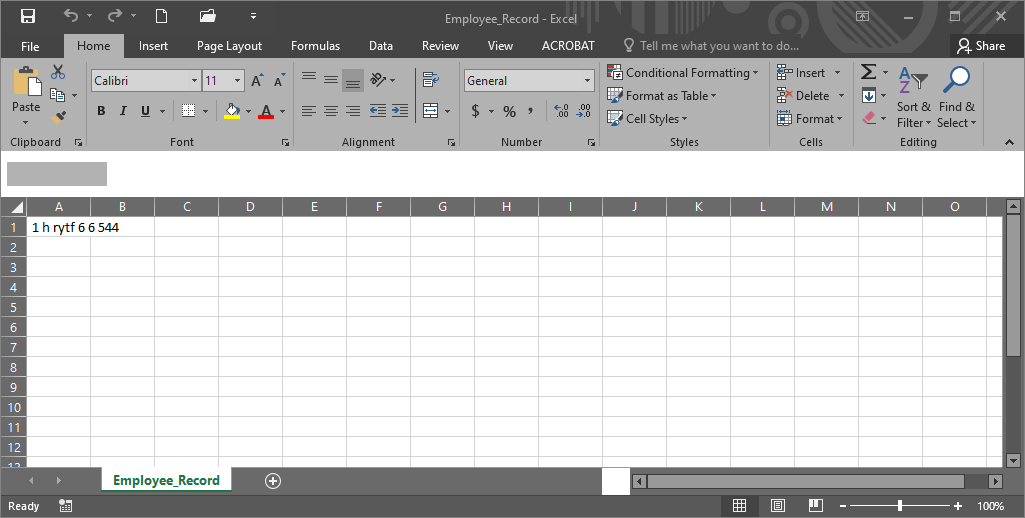


**9. Check Employees Management:**

This is total different file firstly it another header file which is included in bankmanagement.cpp file it having file handling with user to access employee’s data admin will again give the password for verification of admin then admin can able to access employee’s data and perform the following functionality.

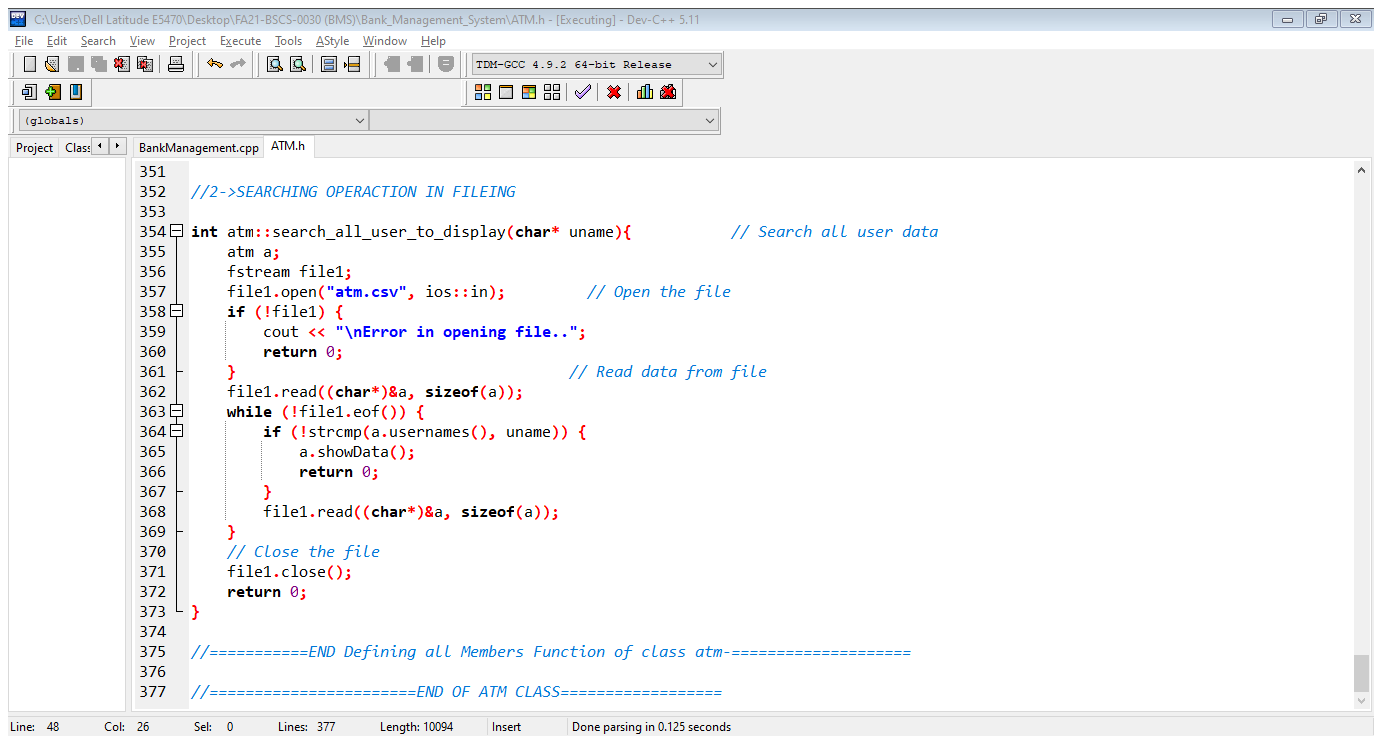


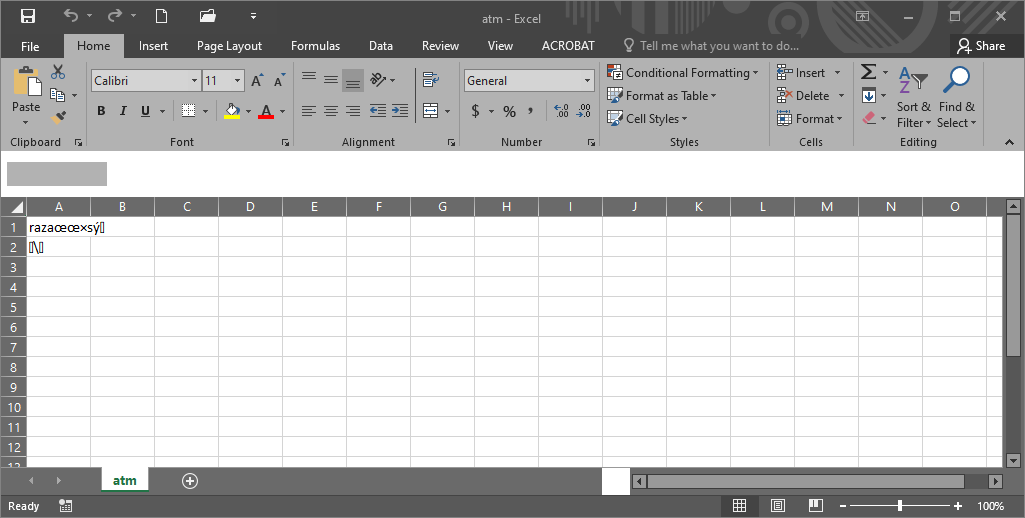




**10. Go to ATM:**

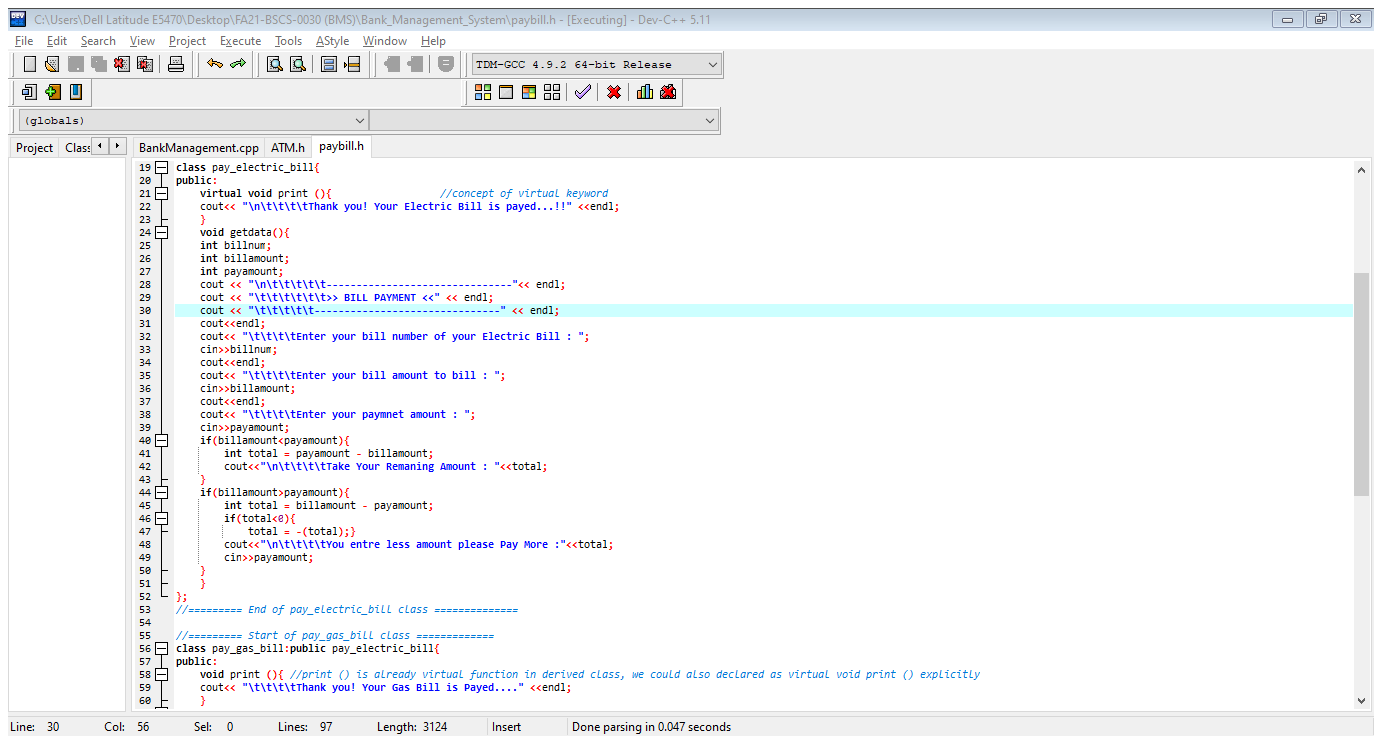
This feature is one of the most important in my project in which those users whose account is already created the admin can able to delete any account of the account holders it having file handling with user to access employee’s data admin will again give the password for verification of admin then admin can able to access ATM data and perform the following functionality.

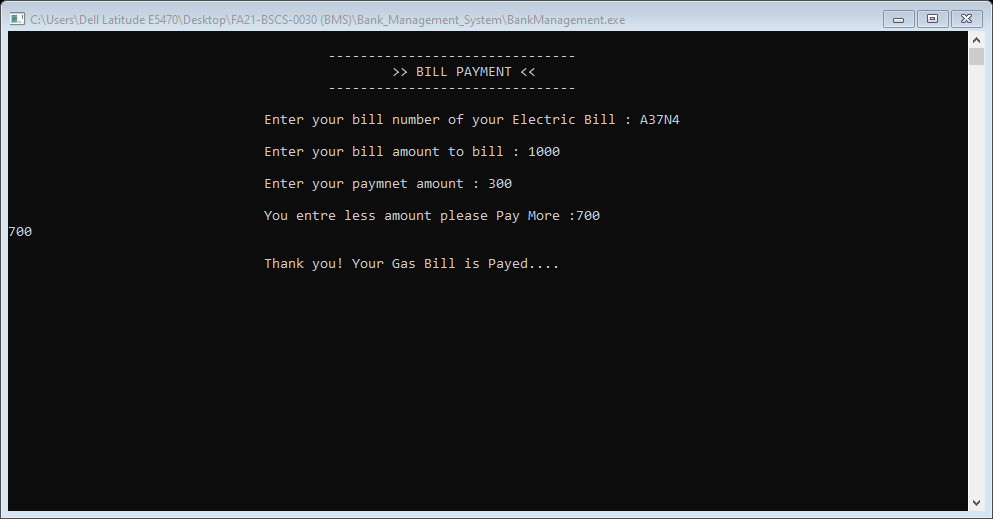




**10. Pay Bill:**

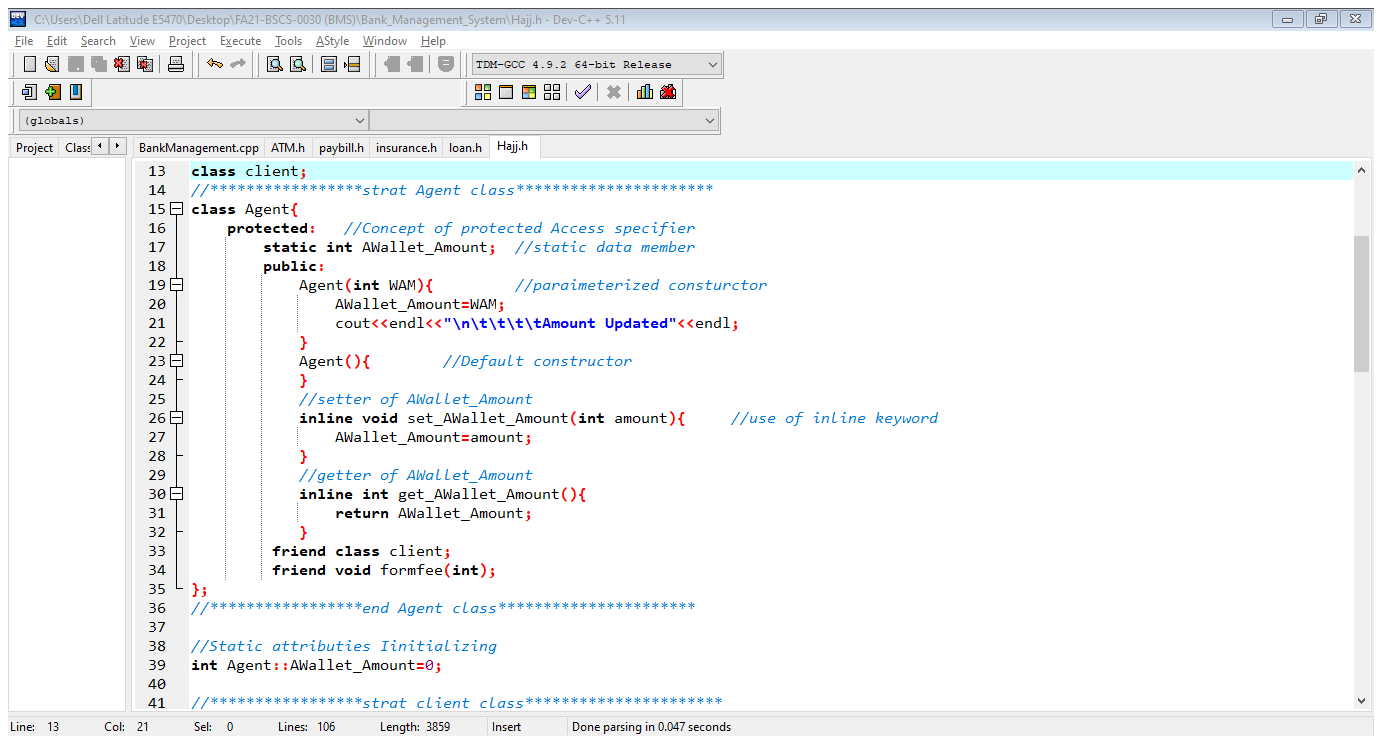
Run Time Polymorphism, In this program there are two class pay\_electric\_bill (base class) & pay\_gas\_bill (Derived class), in base class there is a function void print () this function is also in derived class with different work in calling process compiler confused to display which print() function so to resolve this problem we used virtual keyword with that function then it’s become virtual function.

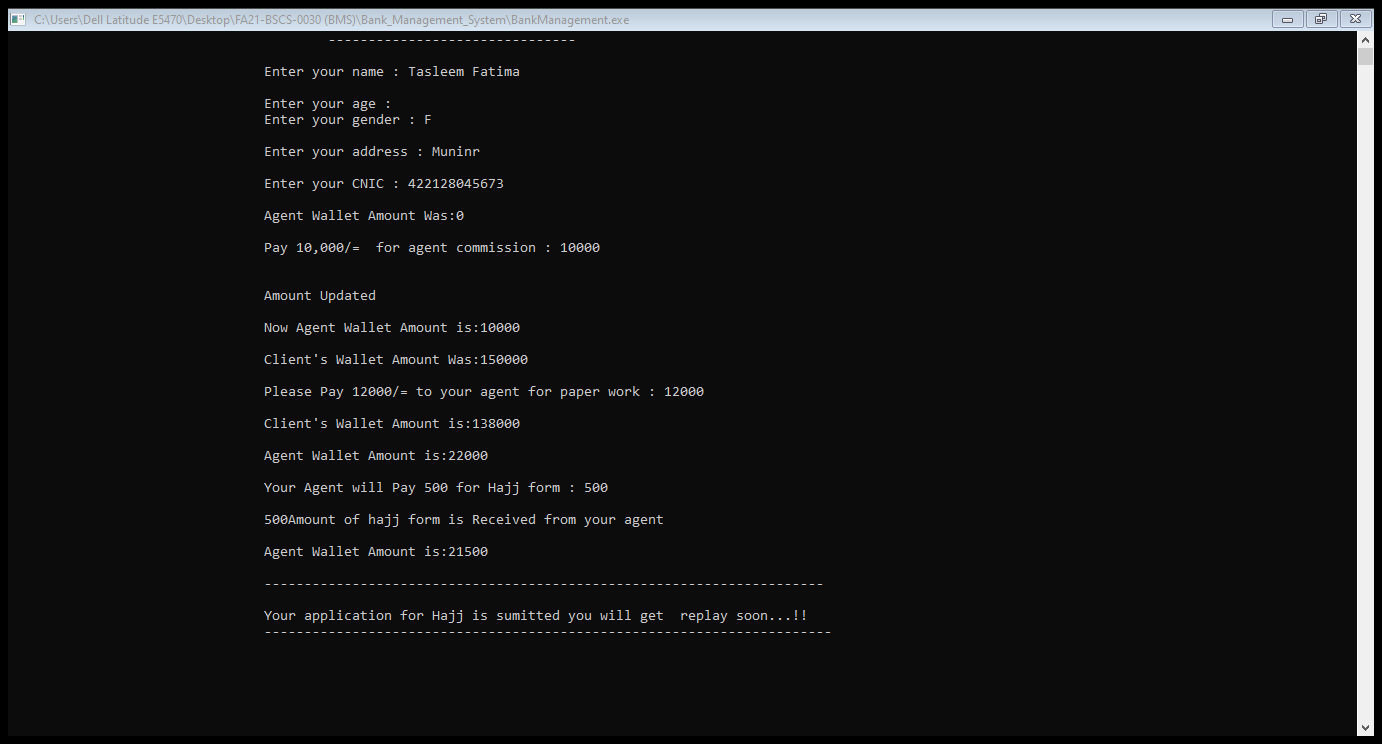




**11. Hajj Application Form:**

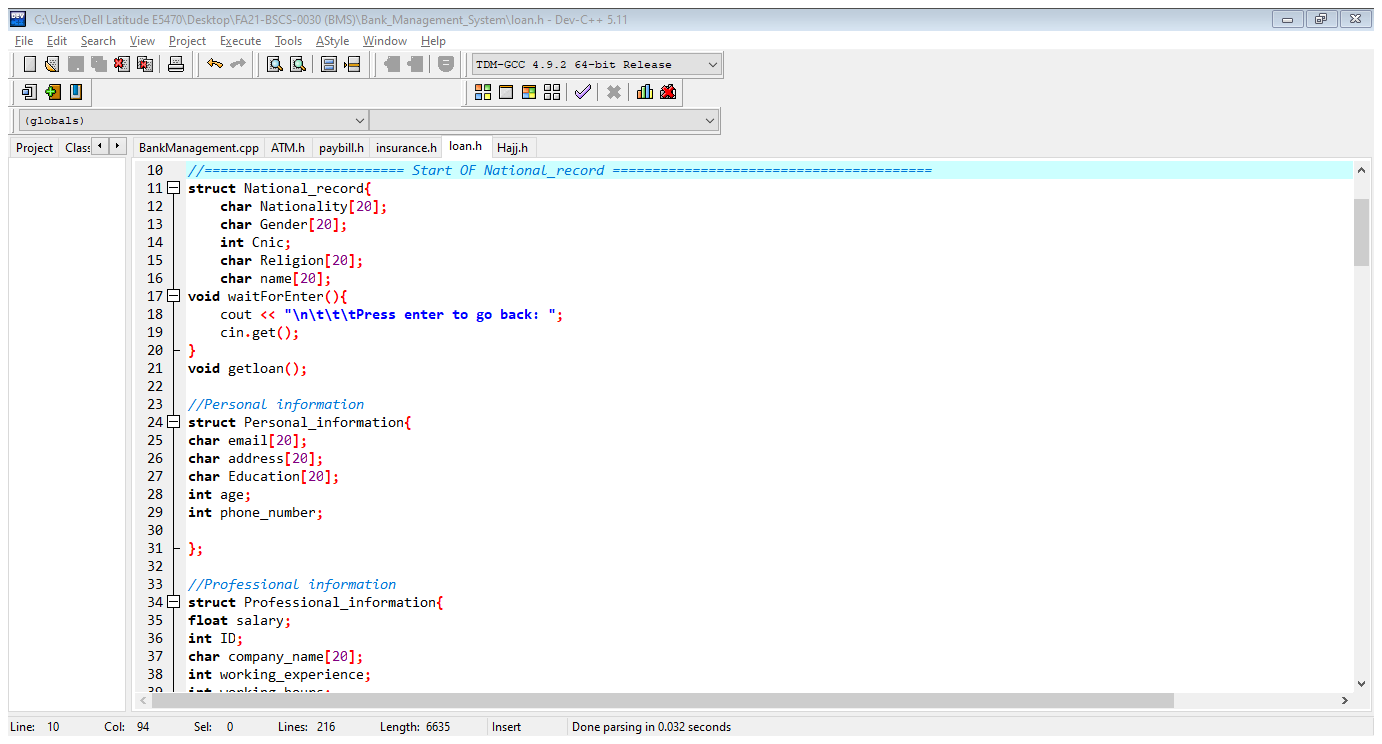
Concept Of Frind Class & Friend Function, in this program there are two class one is agent and other is client class client class is a friend of class agent. This is total different file firstly it another header file which is included Bankmanagement.cpp file it having file handling with user to access employee’s data admin will again give the password for verification of admin then admin can able to access data and perform the following functionality.

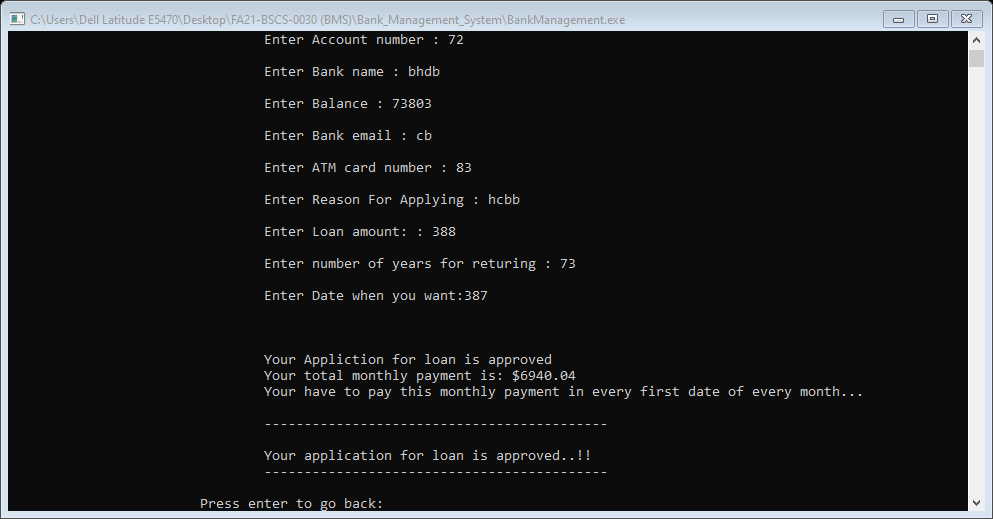


****

**12. Apply for loan:**

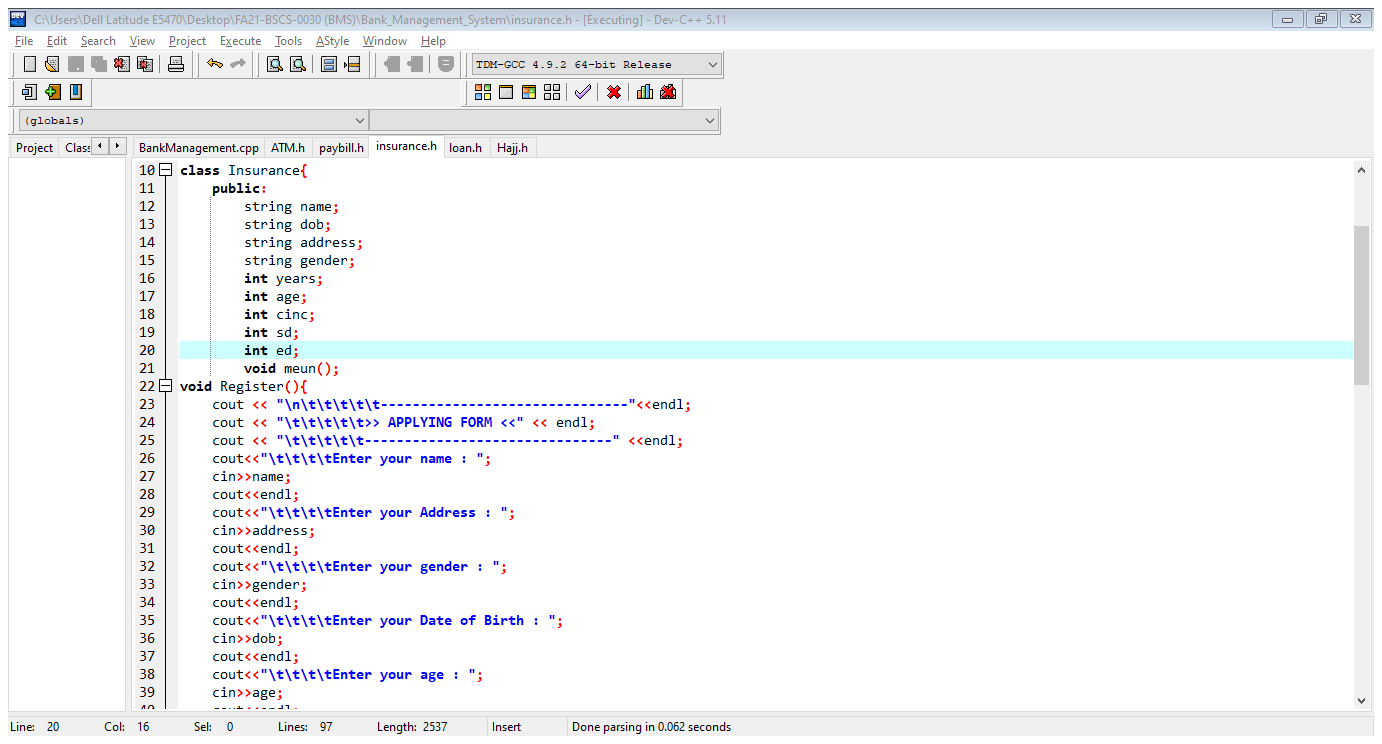
CONCEPT OF NESTED STRUCTURE, in this program there is a National\_record in which it member data & function are present in this struct then there are three diffrerent struct named Personal\_information, Professional\_information & Bank\_information is nested inside the National\_record.

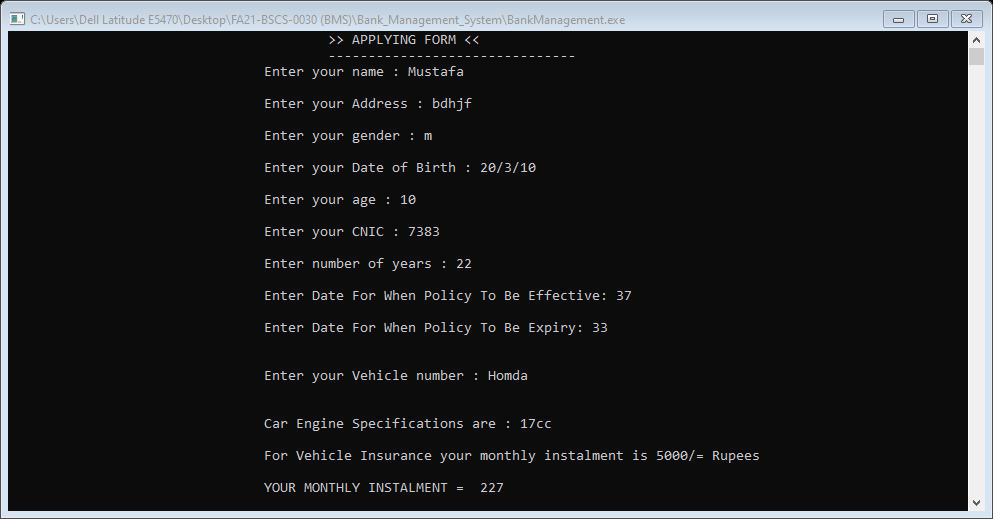


****

**13. Apply for Insurance:**

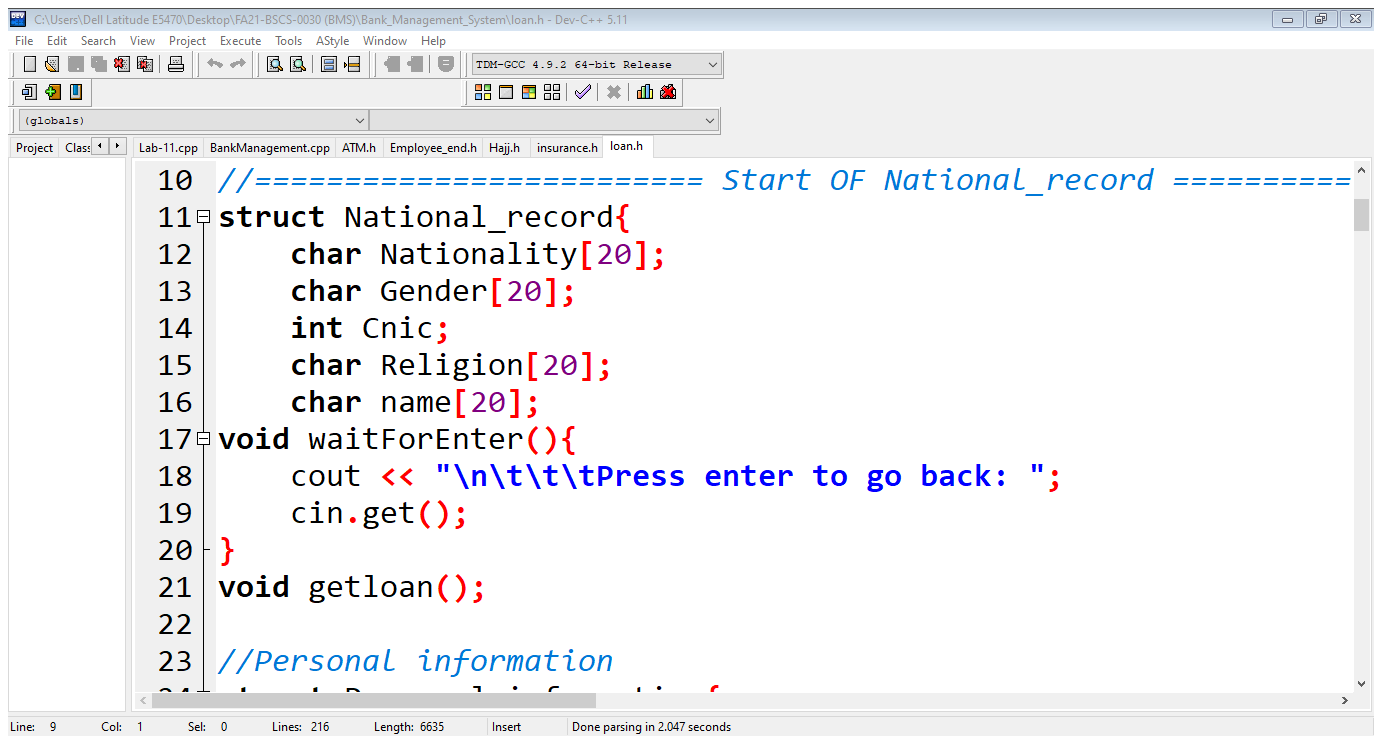
CONCEPT OF NESTED CLASS, in this program there is a class Insurance which is a base class and other class called Vehicle\_Insurance which is the derived class of Insurance in this derived class there is another class Engine which is the nested of Vehicle\_Insurance class.



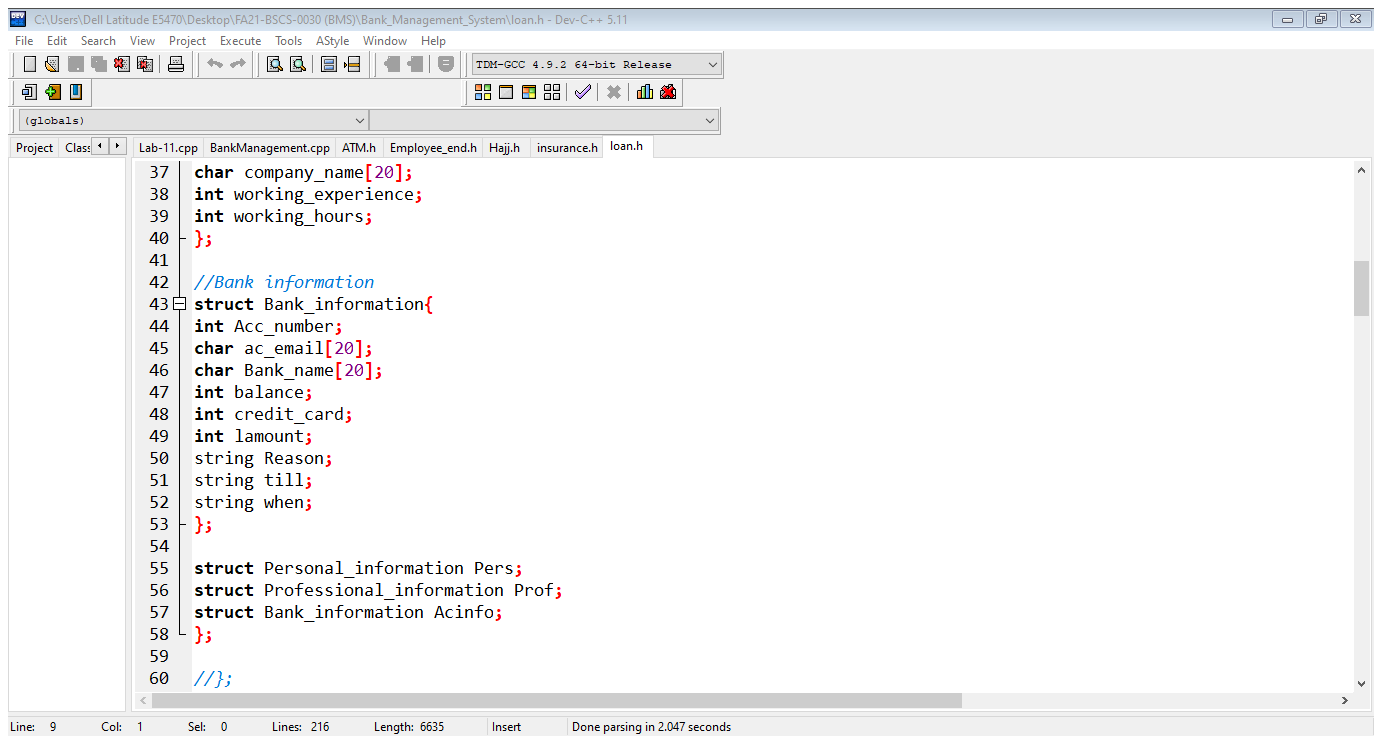
****

**IMPLIMANTATION OF OOP IN PROJECT:**

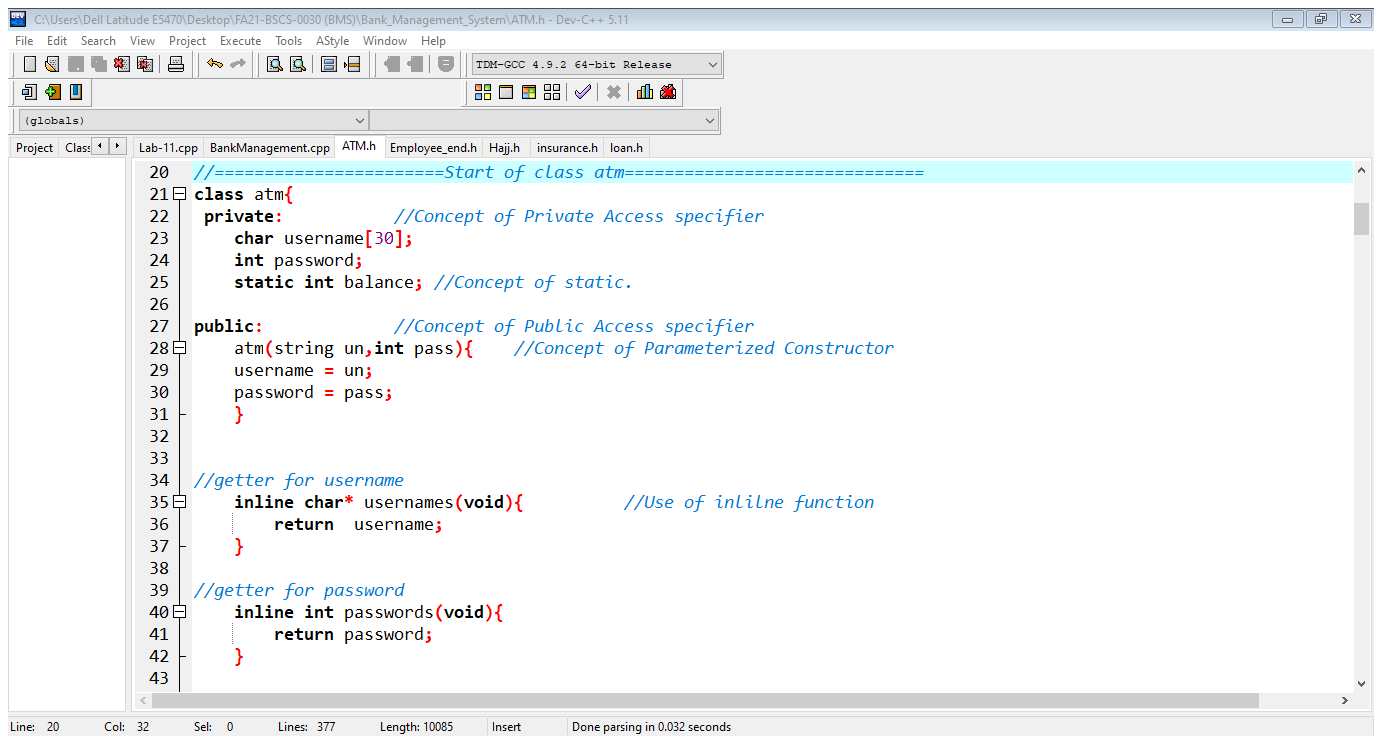
1. Structure



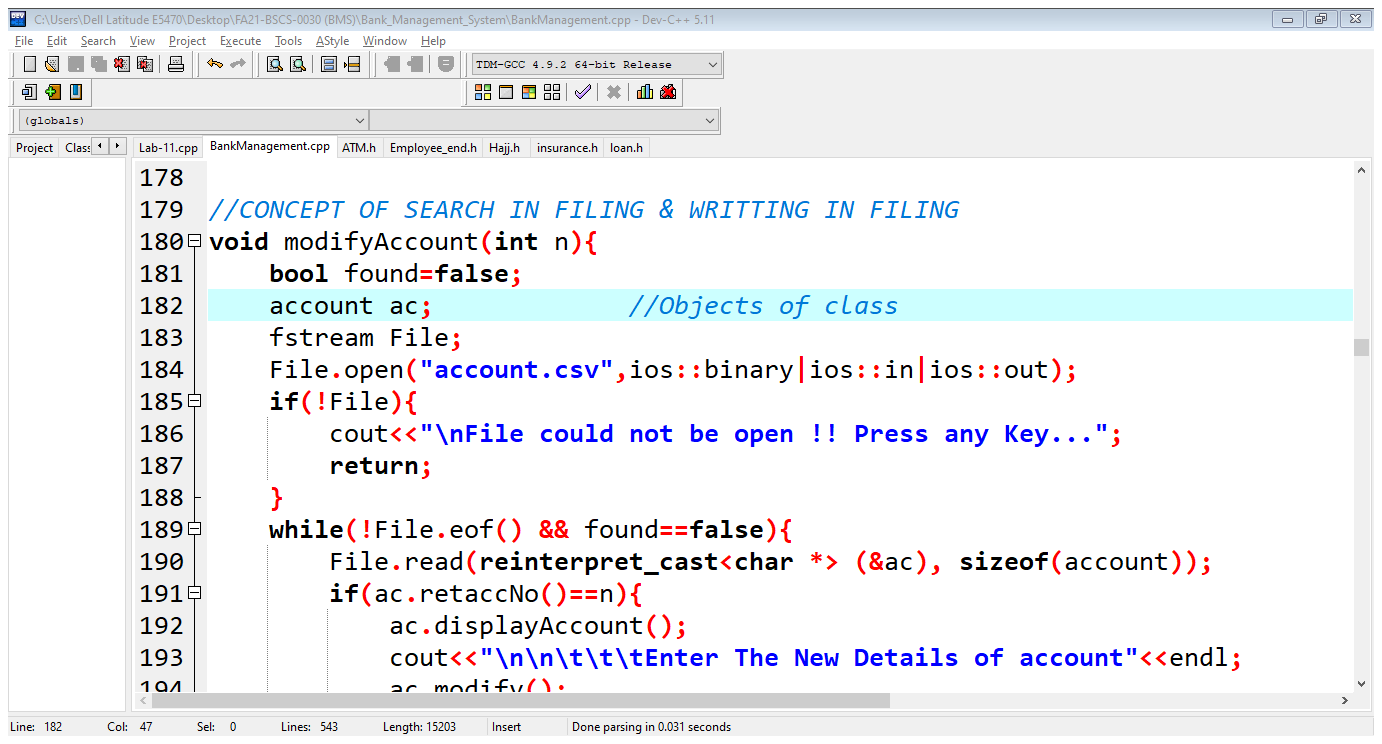
1. Nested Structure



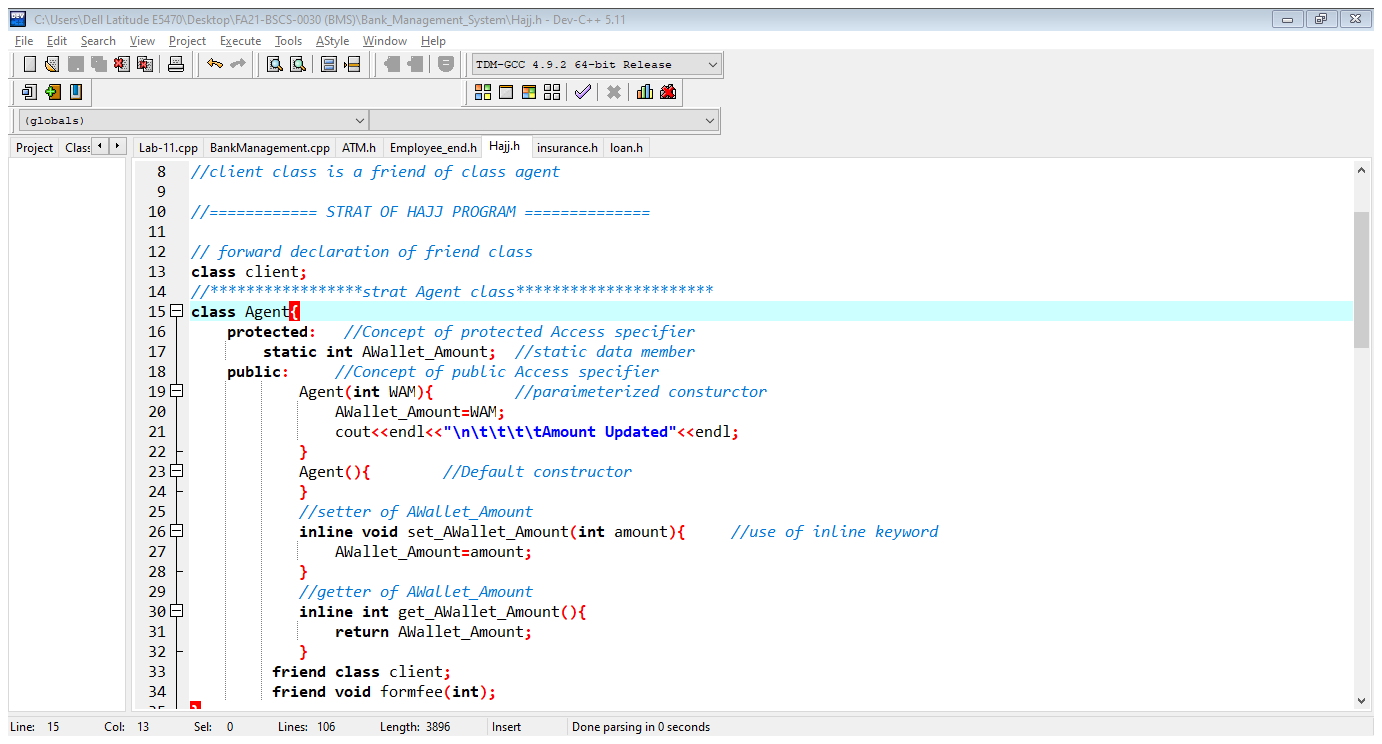
1. Classes



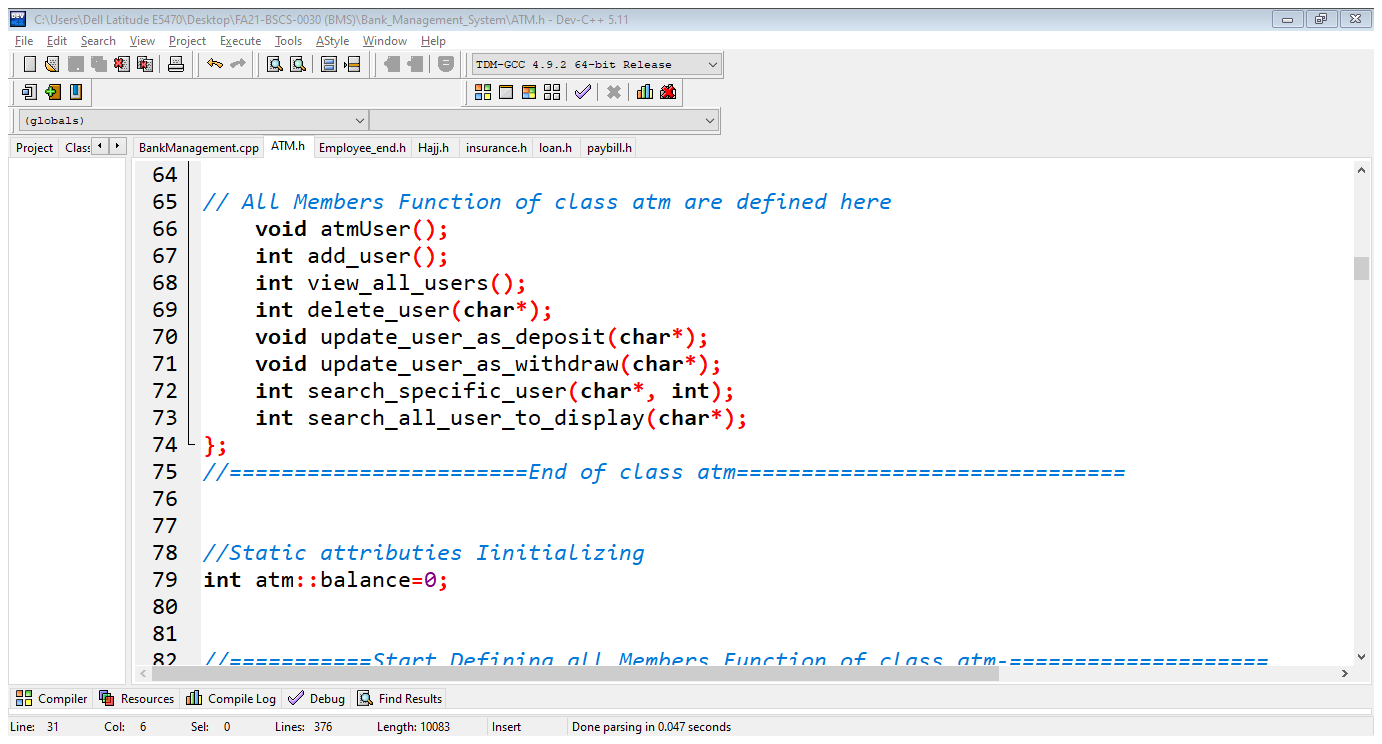
1. Objects



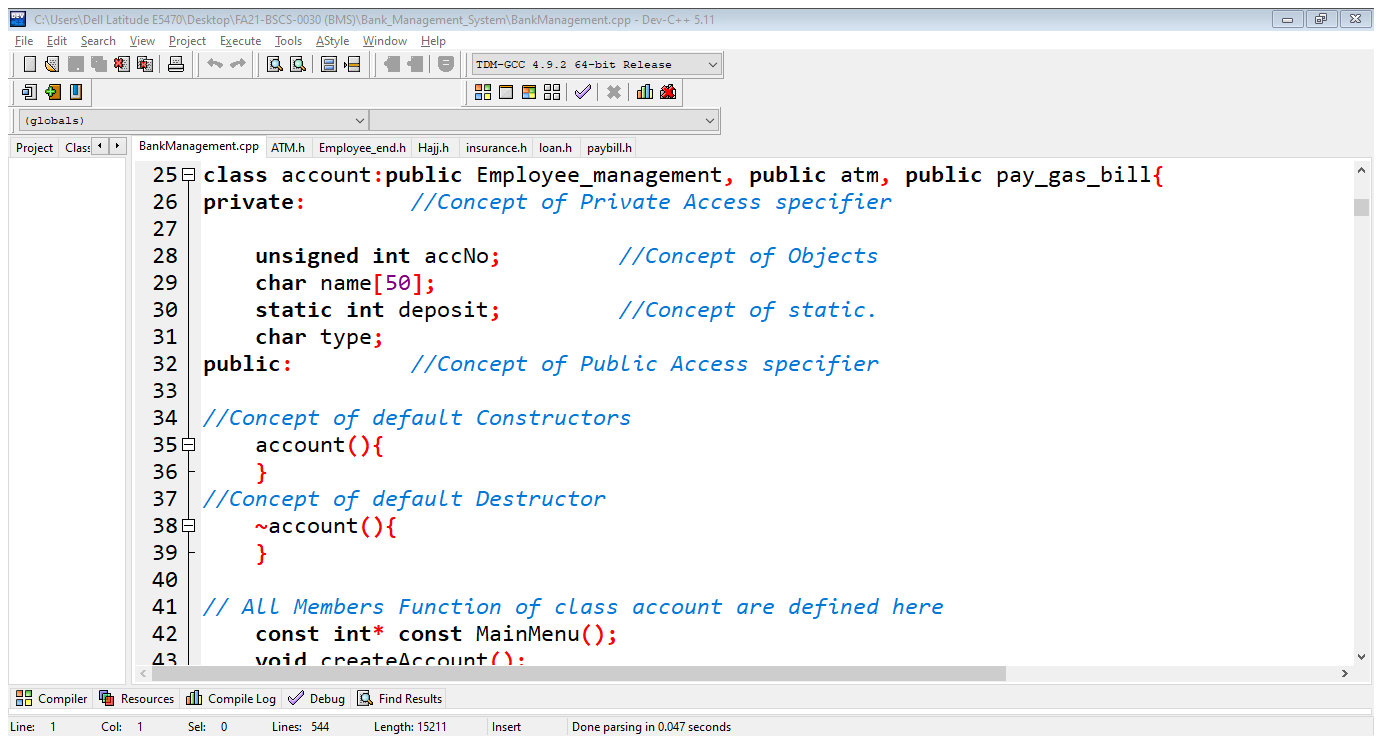
1. Access specifier (public, private, protected)



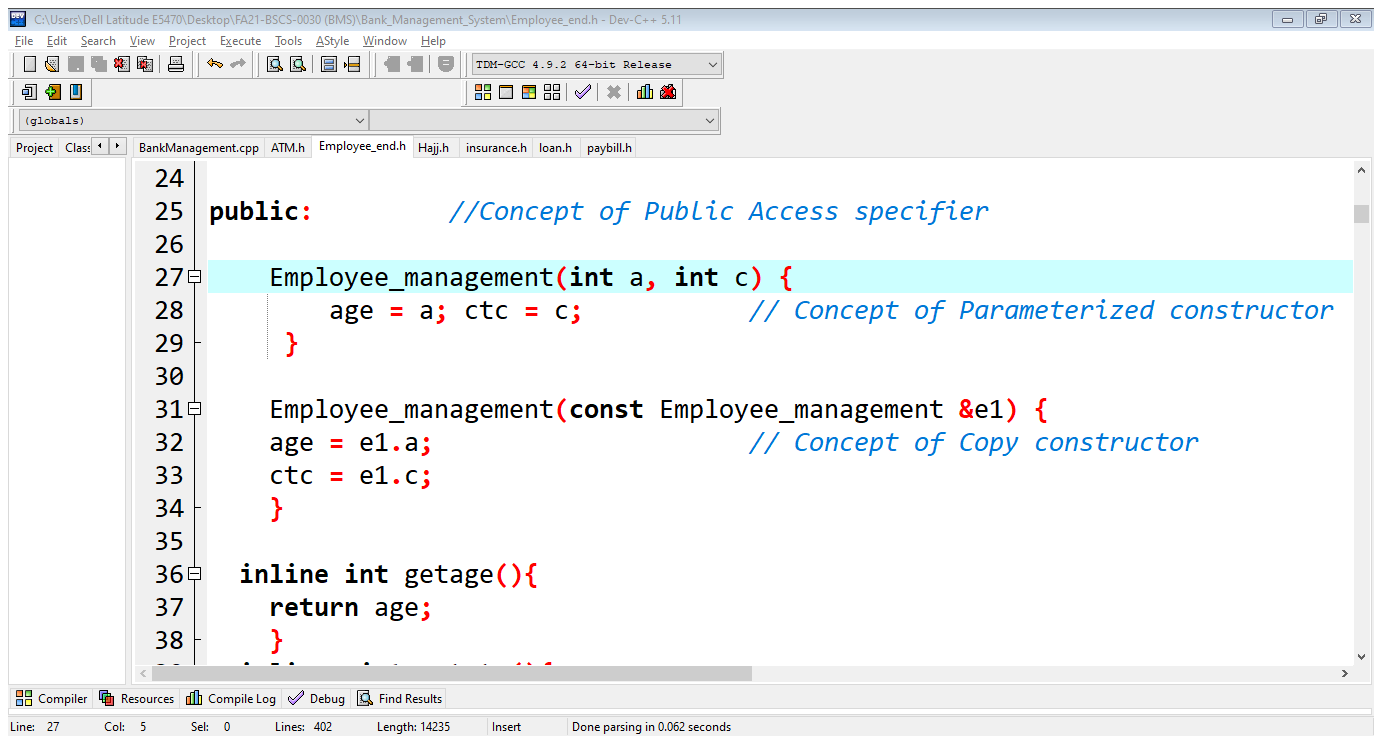
1. Members Function



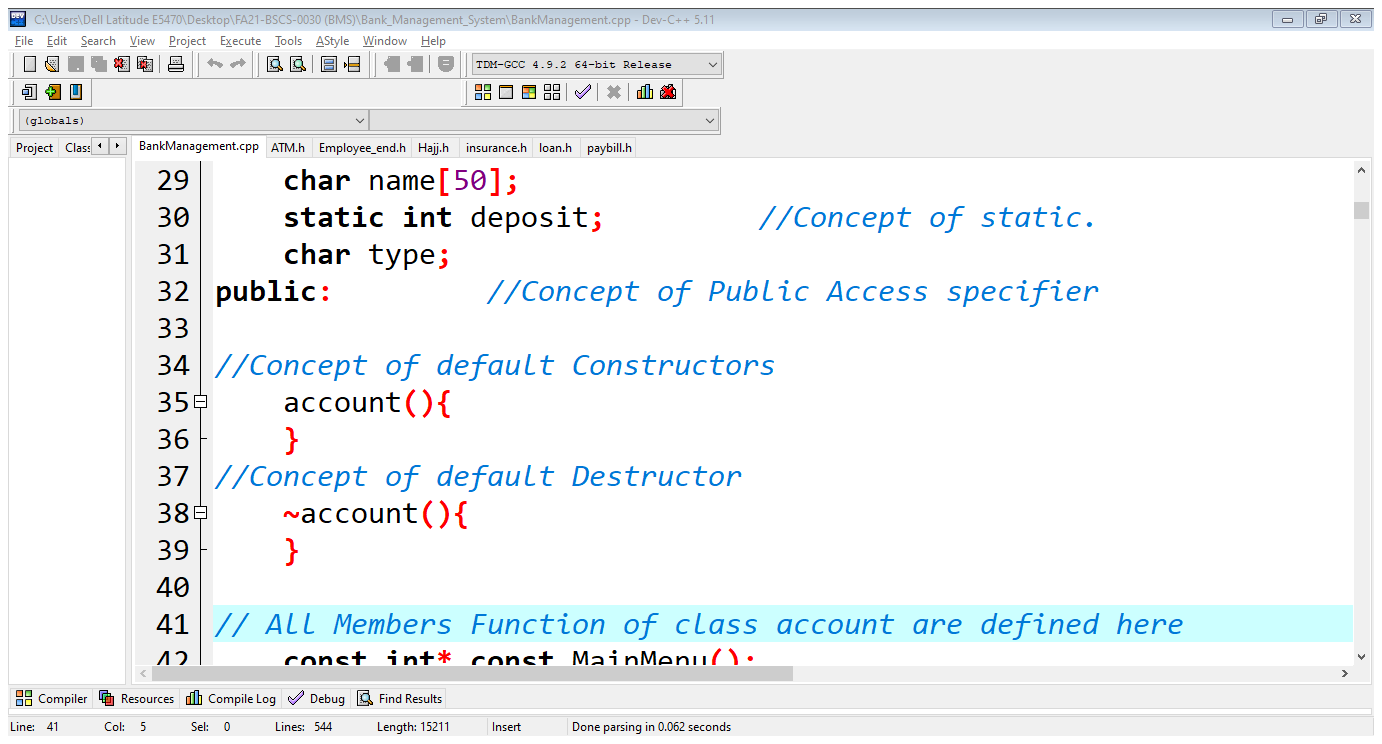
1. Members Data



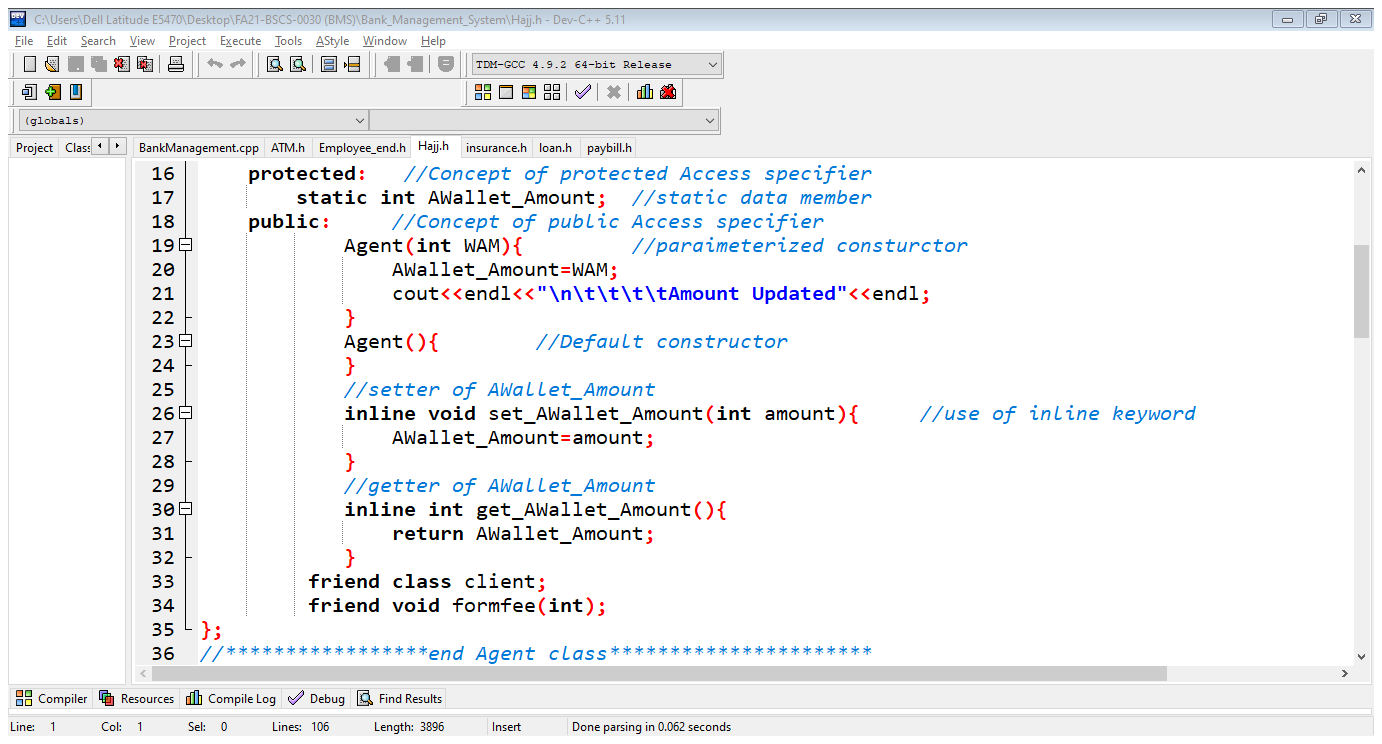
1. Constructor (No-argument, Default, Parameterized Constructor)



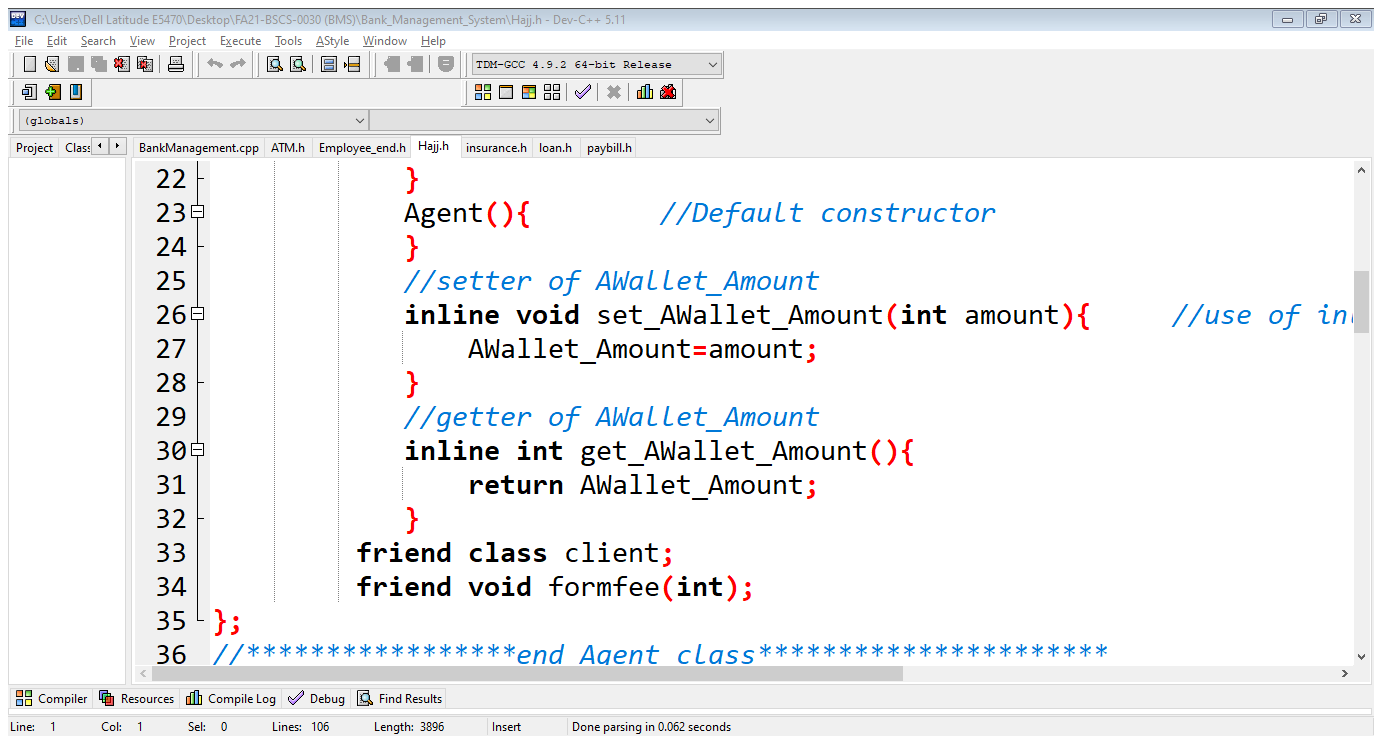
1. Destructor



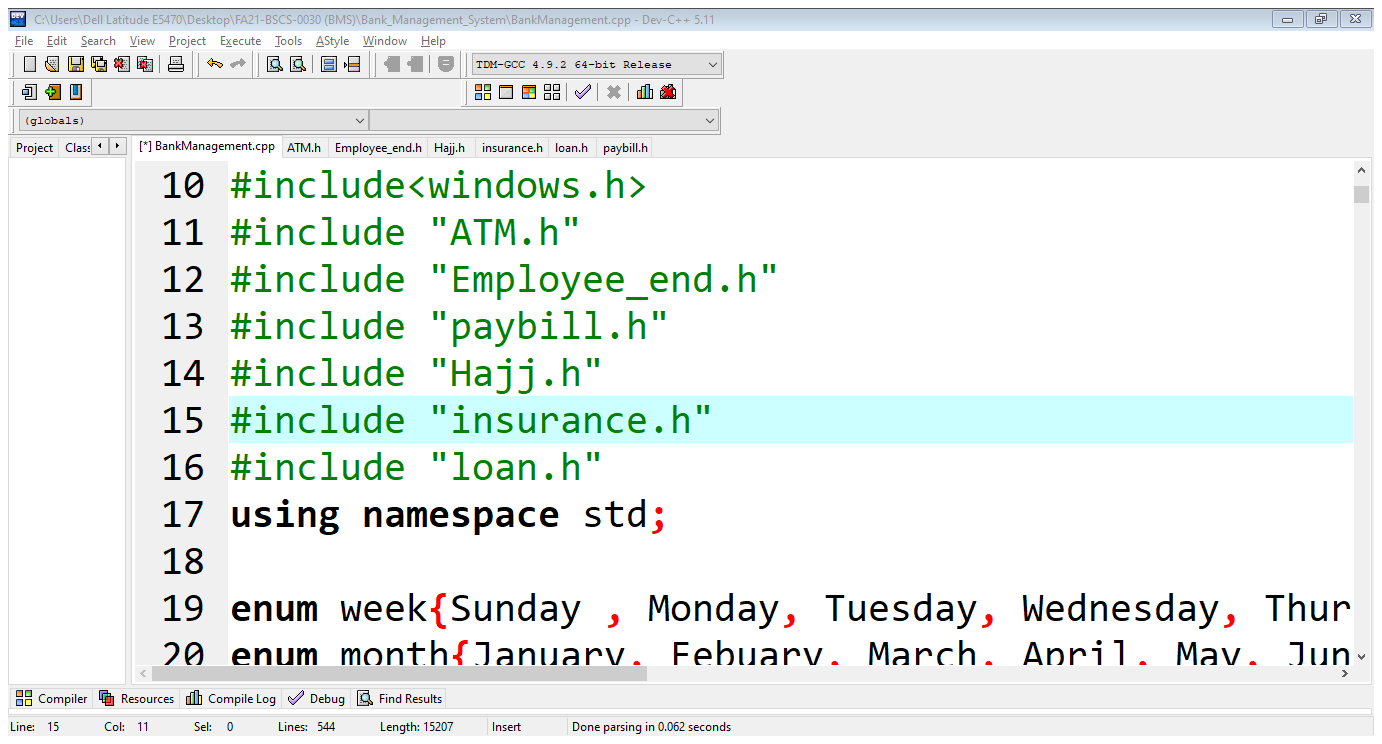
1. Setter



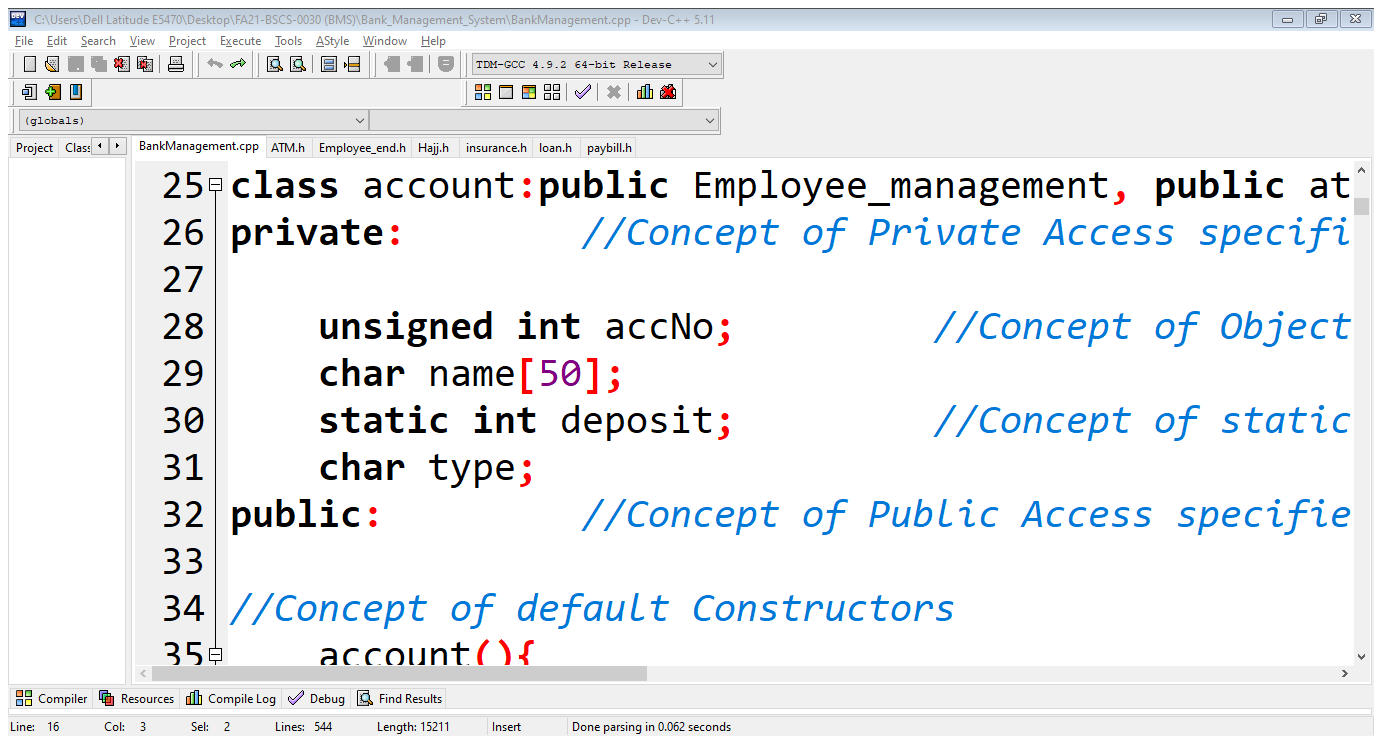
1. Getter



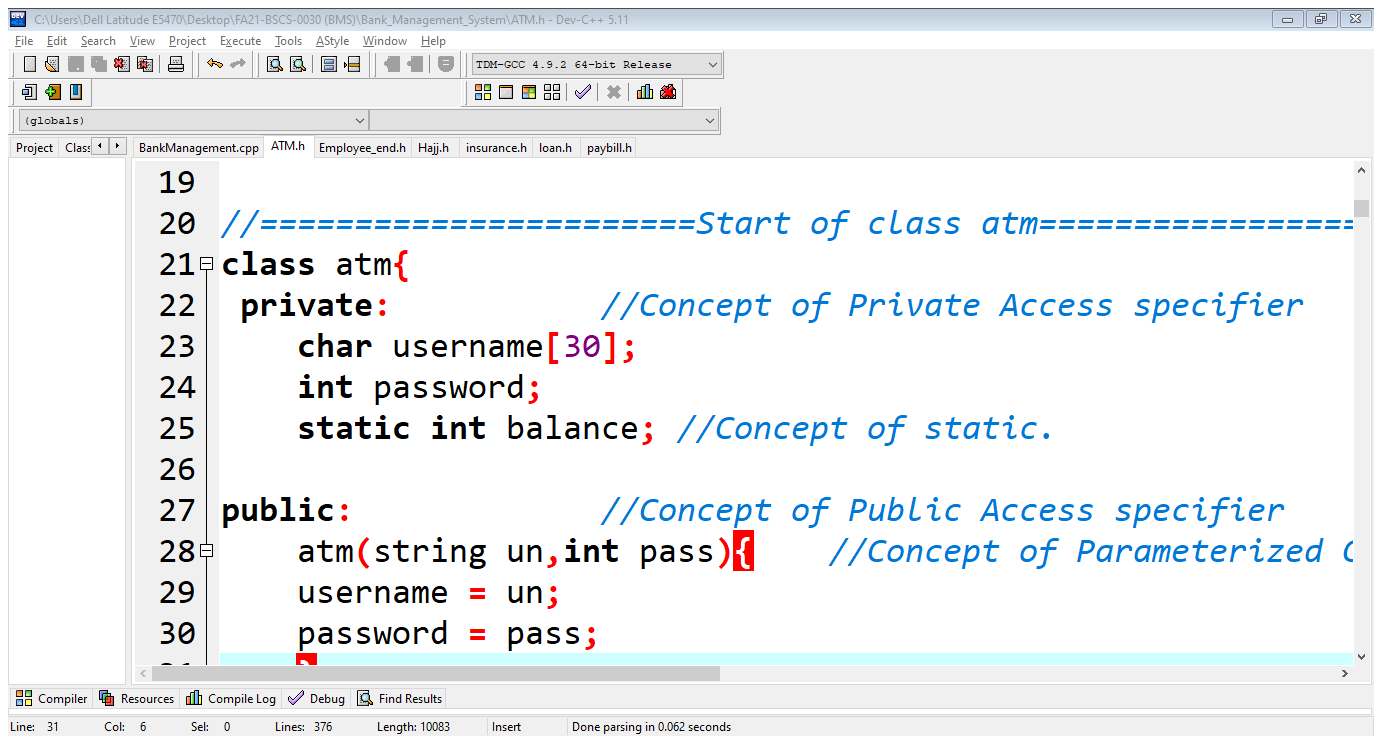
1. Header files



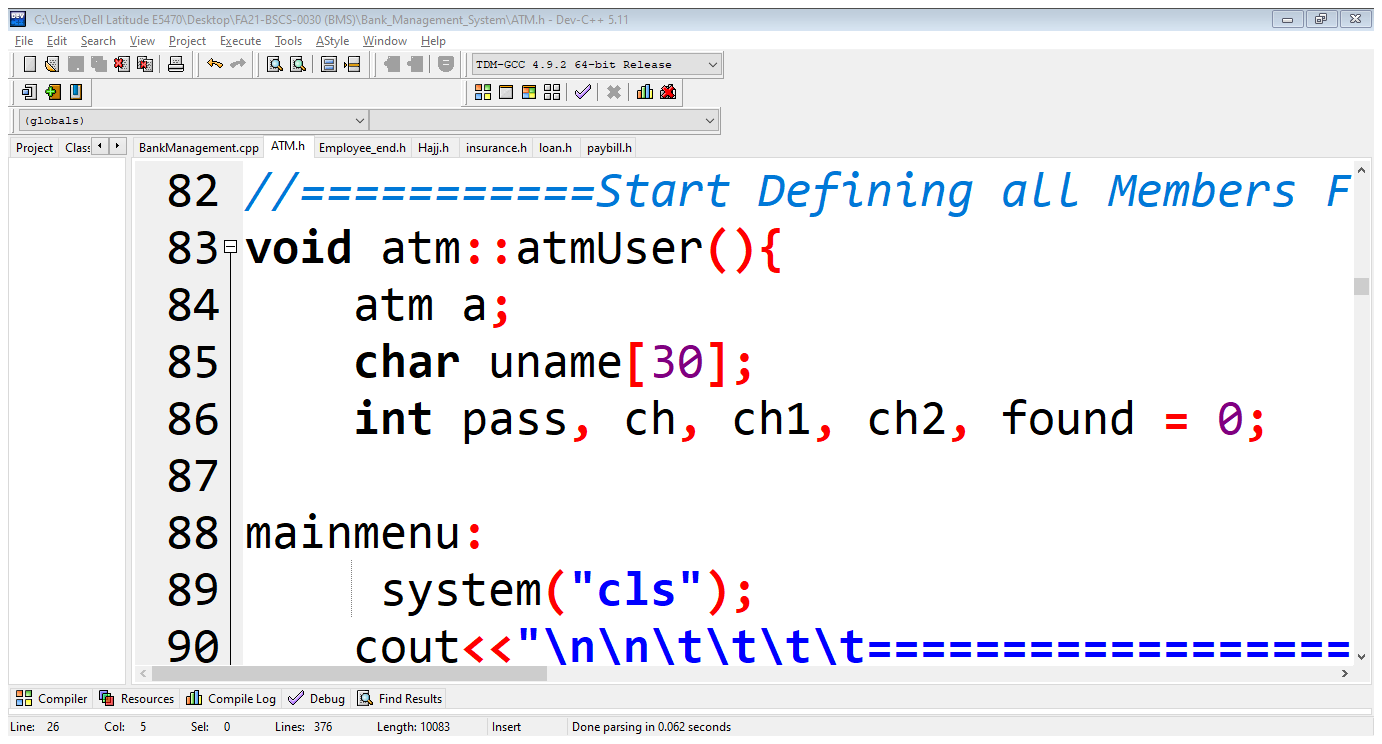
1. Static and Dynamic Arrays



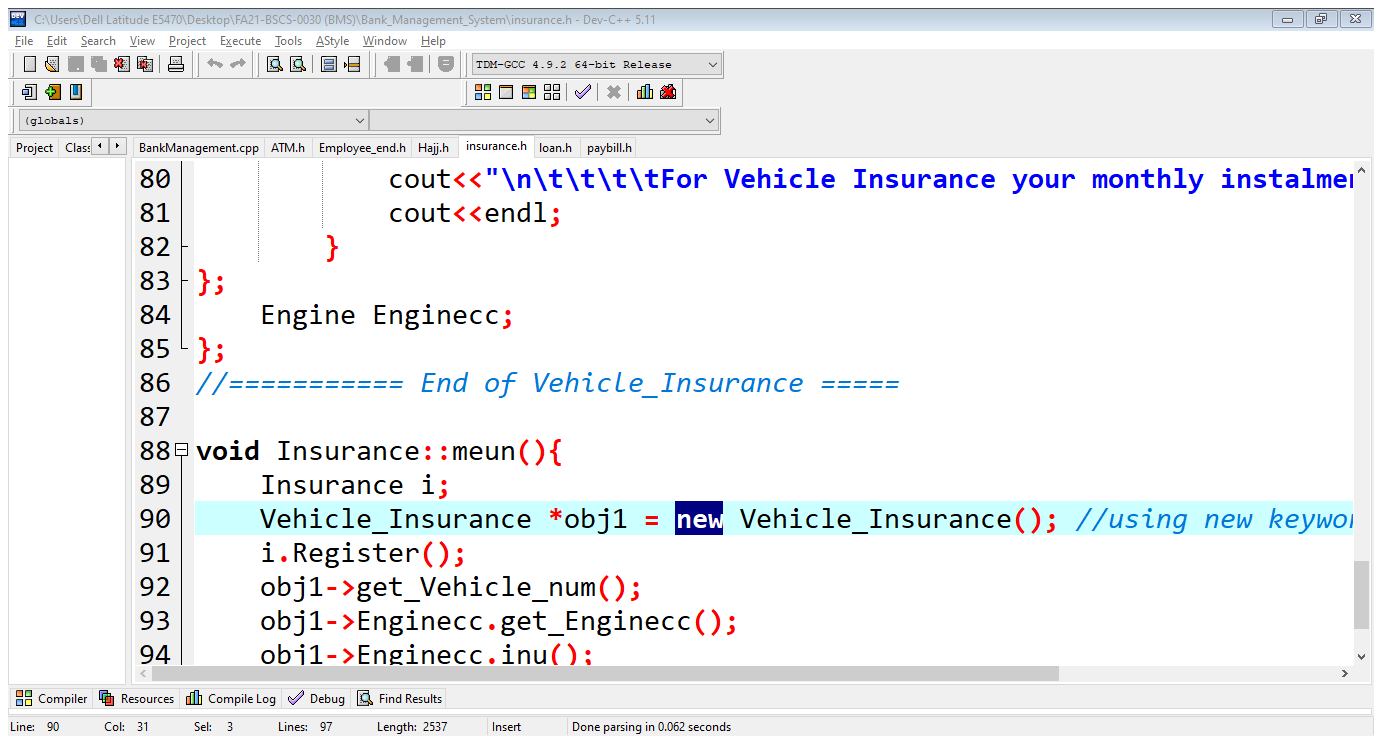
1. Static data and member functions



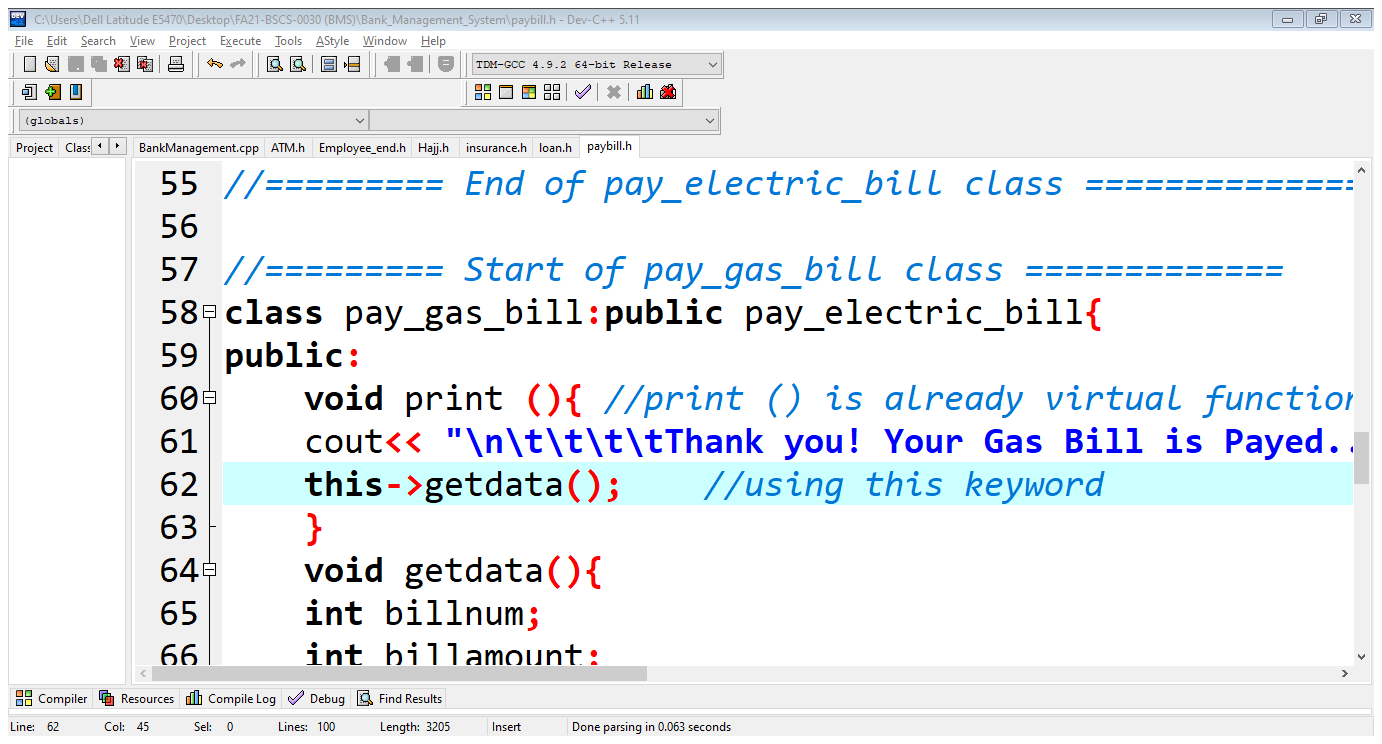
1. Array of Objects



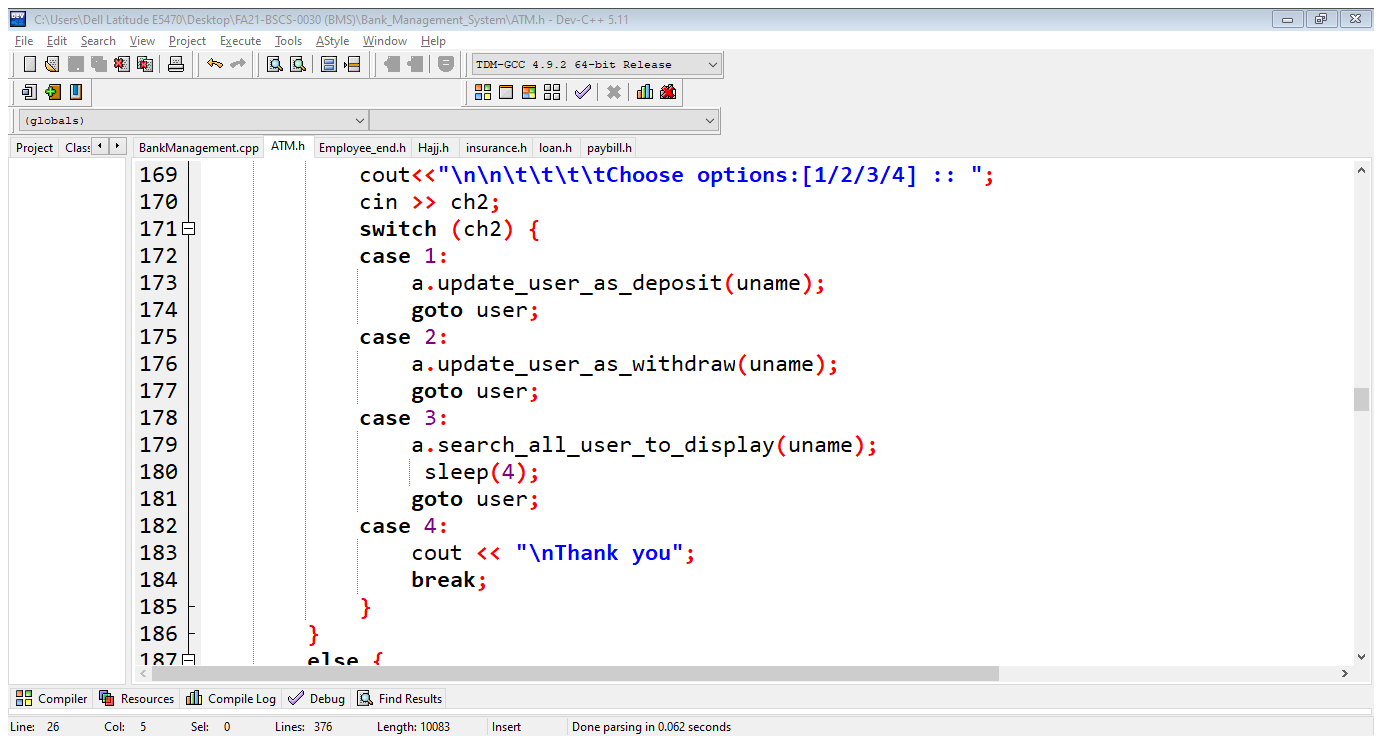
1. New Keyword



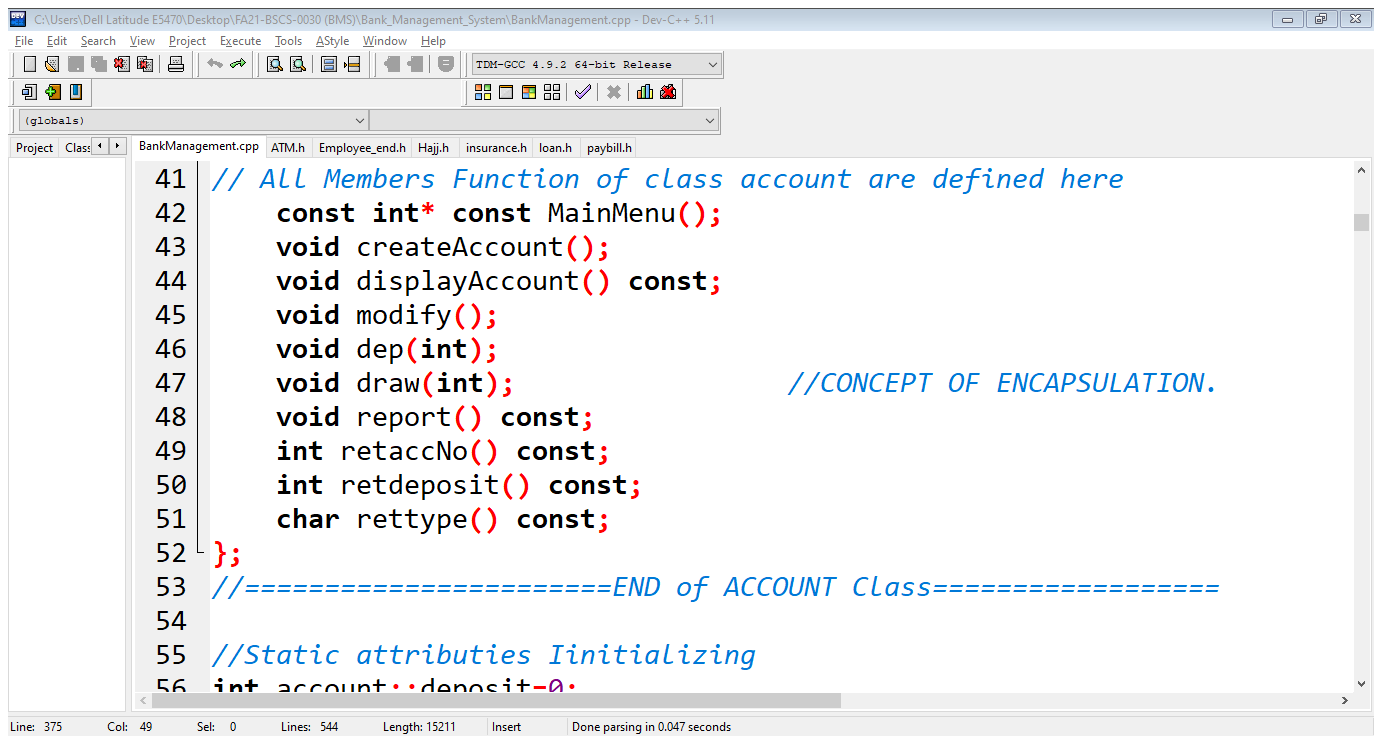
1. This Keyword



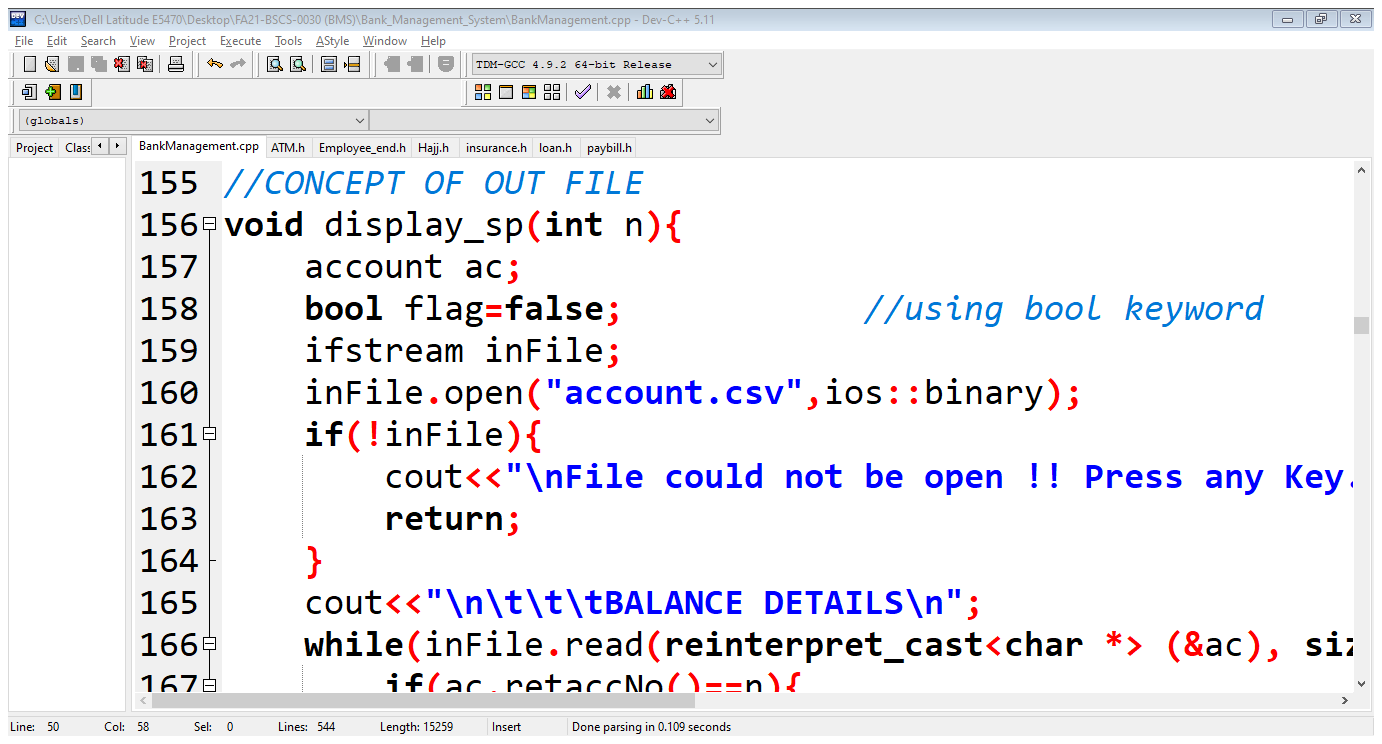
1. Goto Keyword



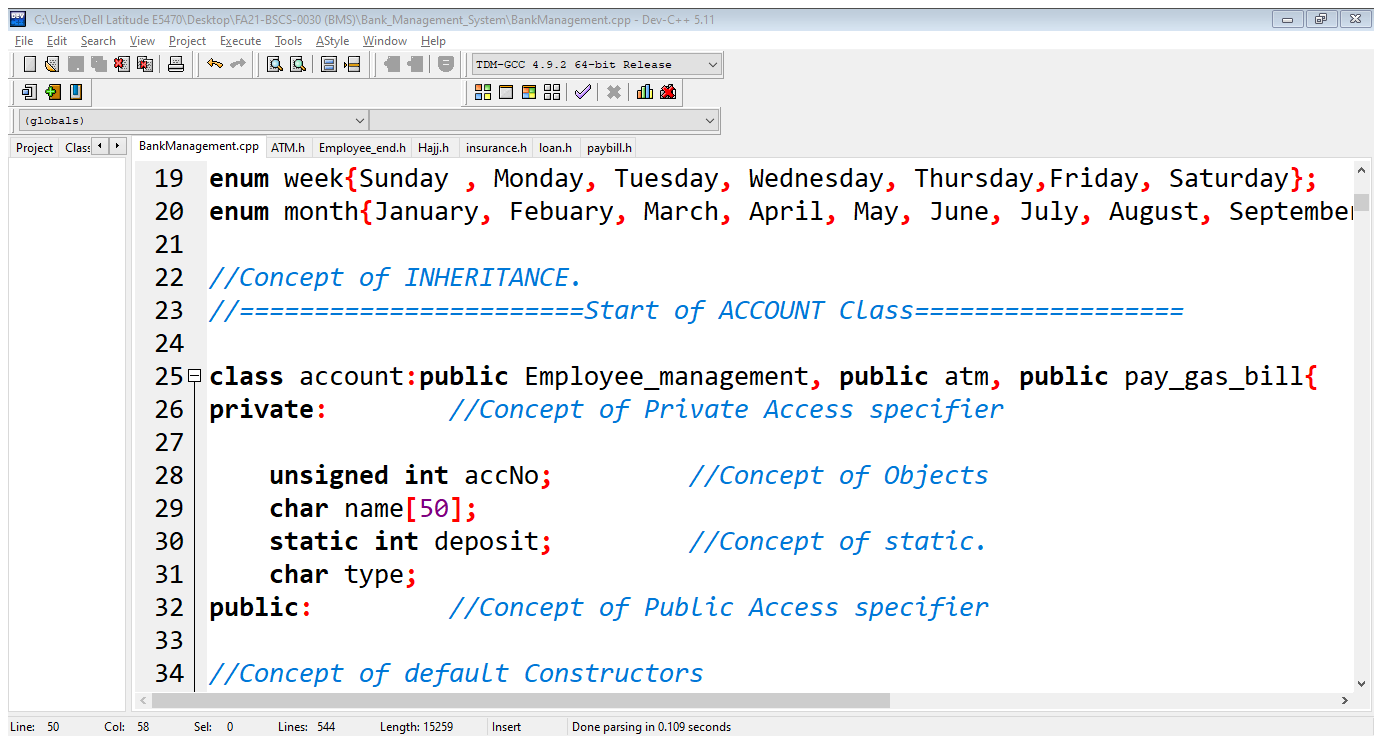
1. Constant & its types



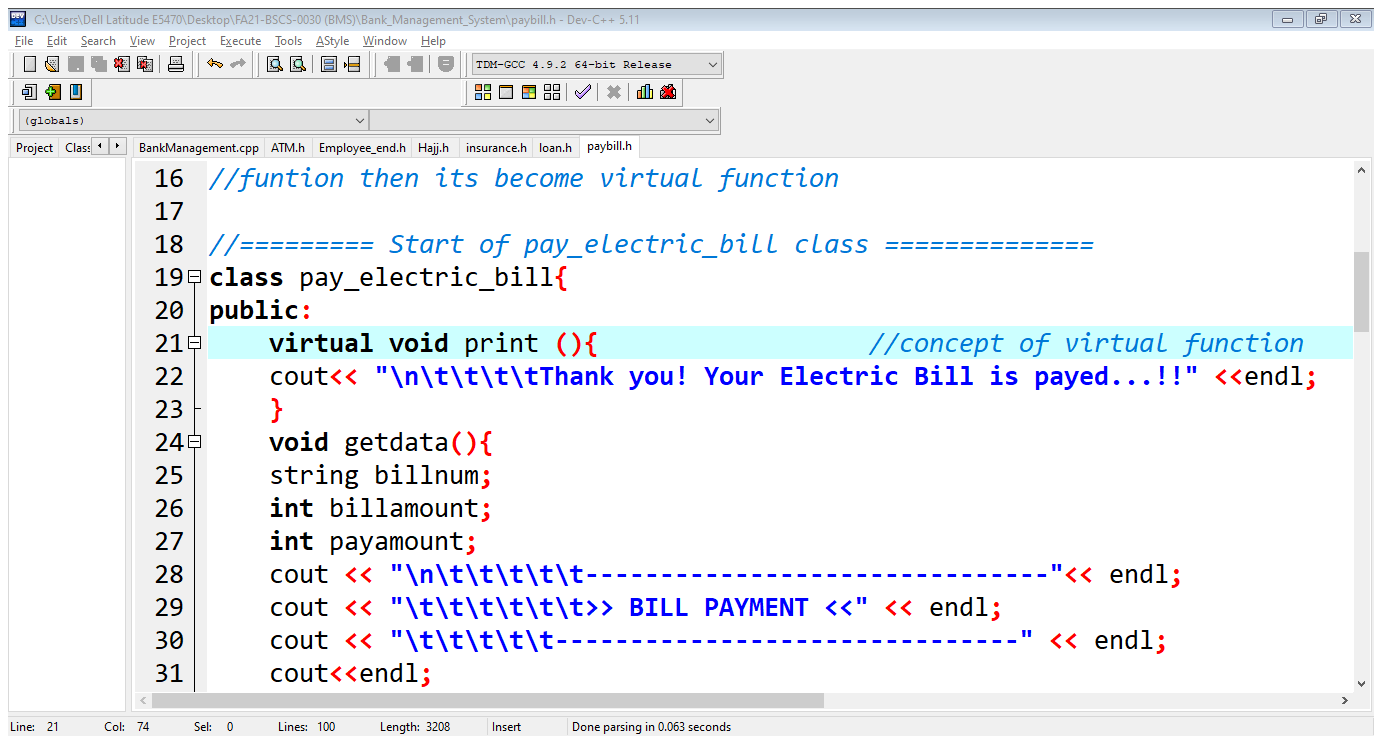
1. Bool Keyword



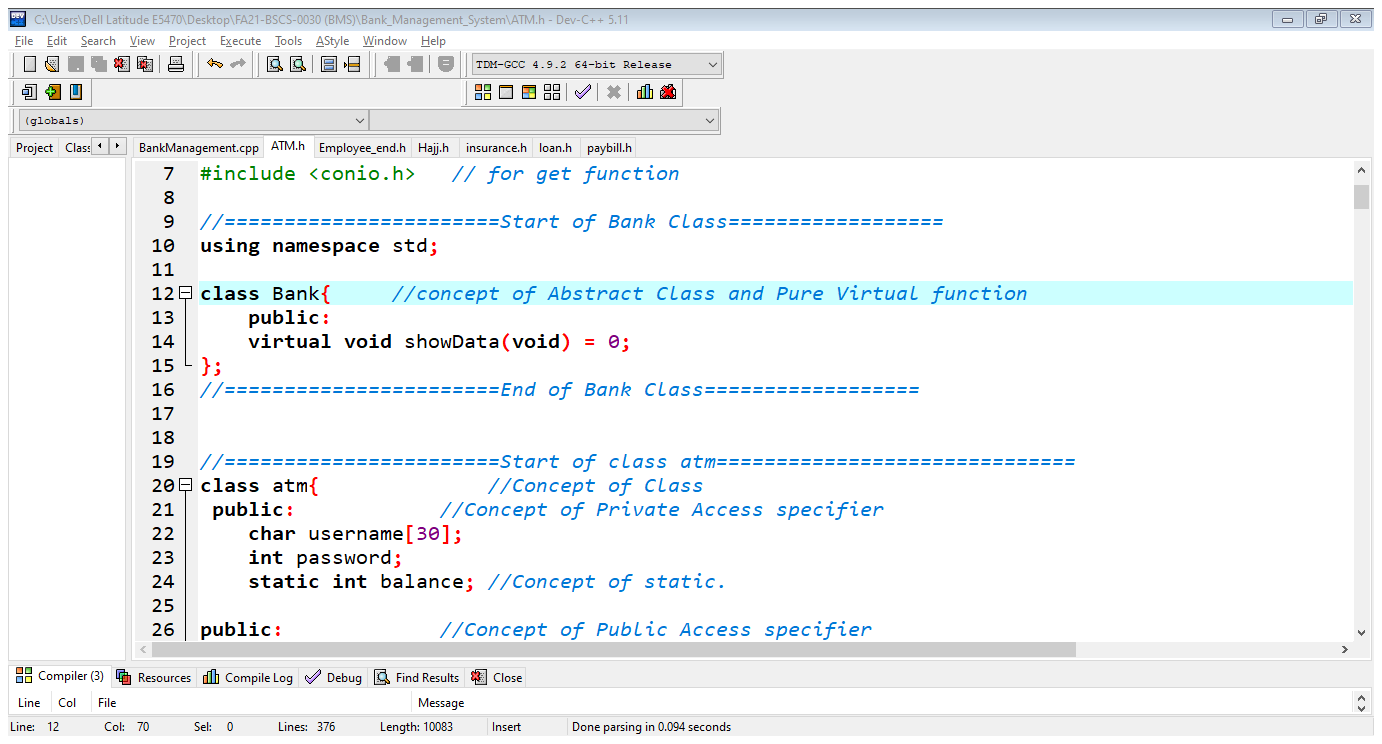
1. Multiple Inheritance



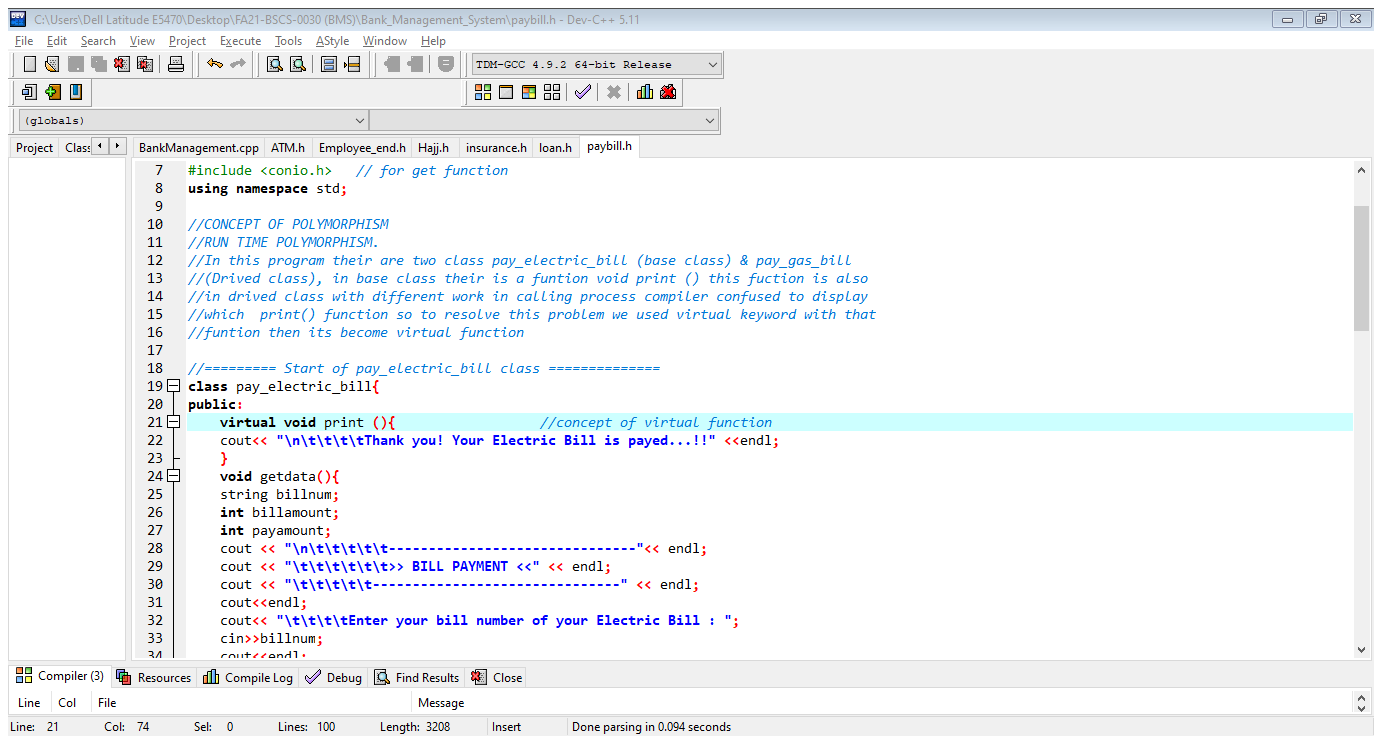
1. Virtual Function



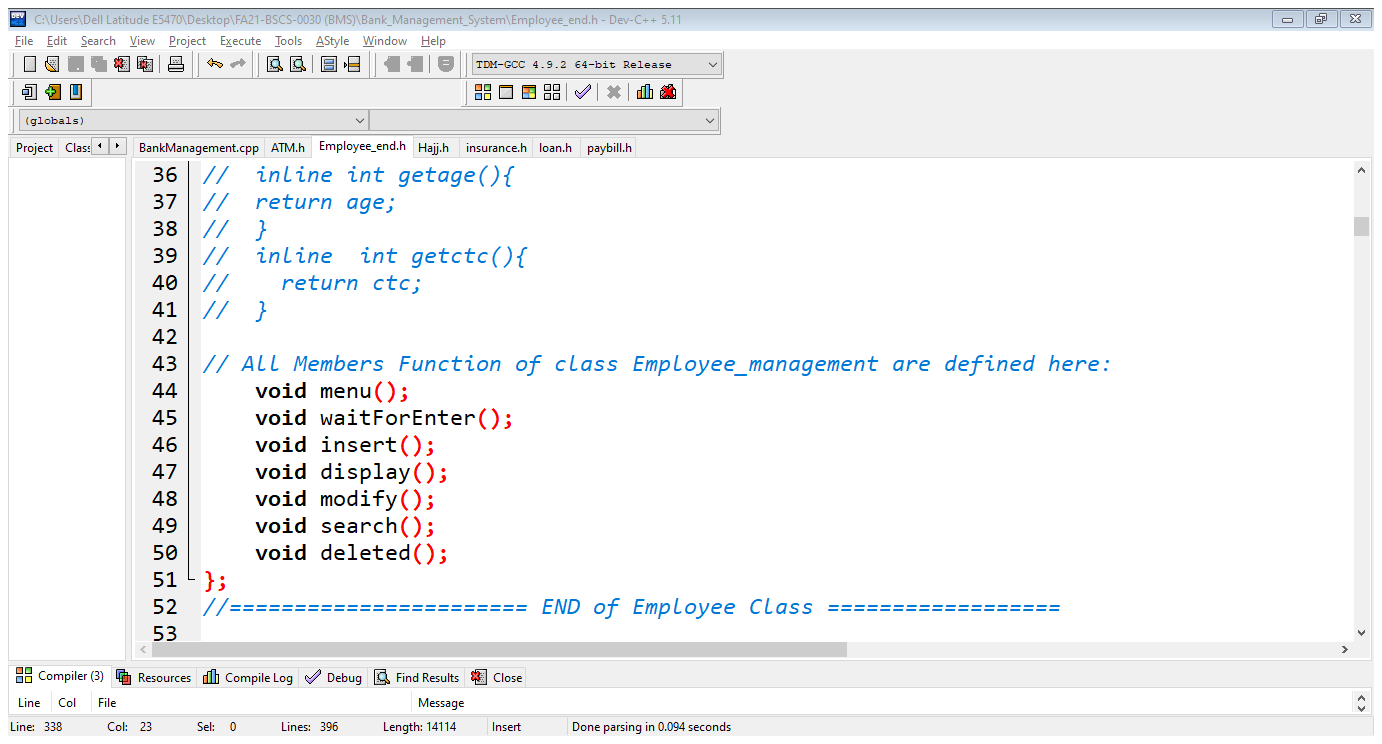
1. Abstract class & Pure Virtual



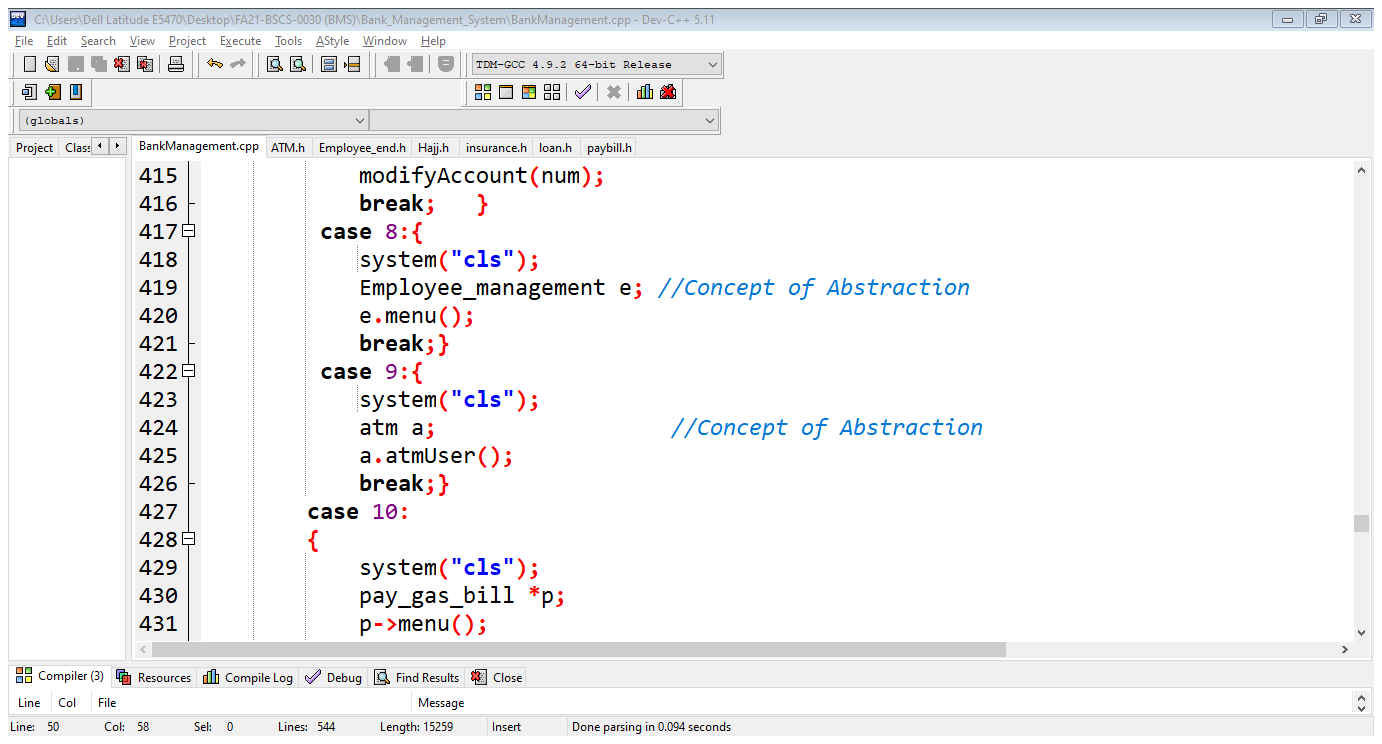
1. Run Time Polymorphism



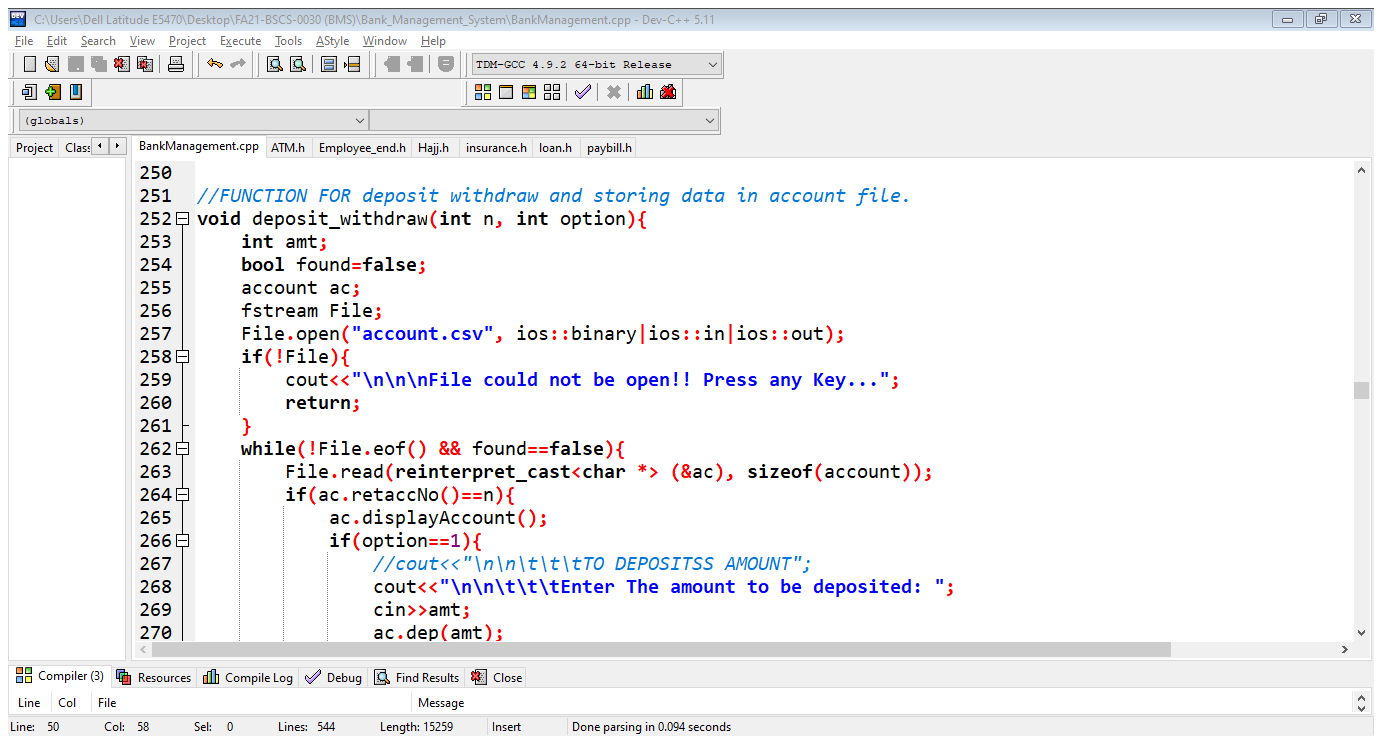
1. Encapsulation



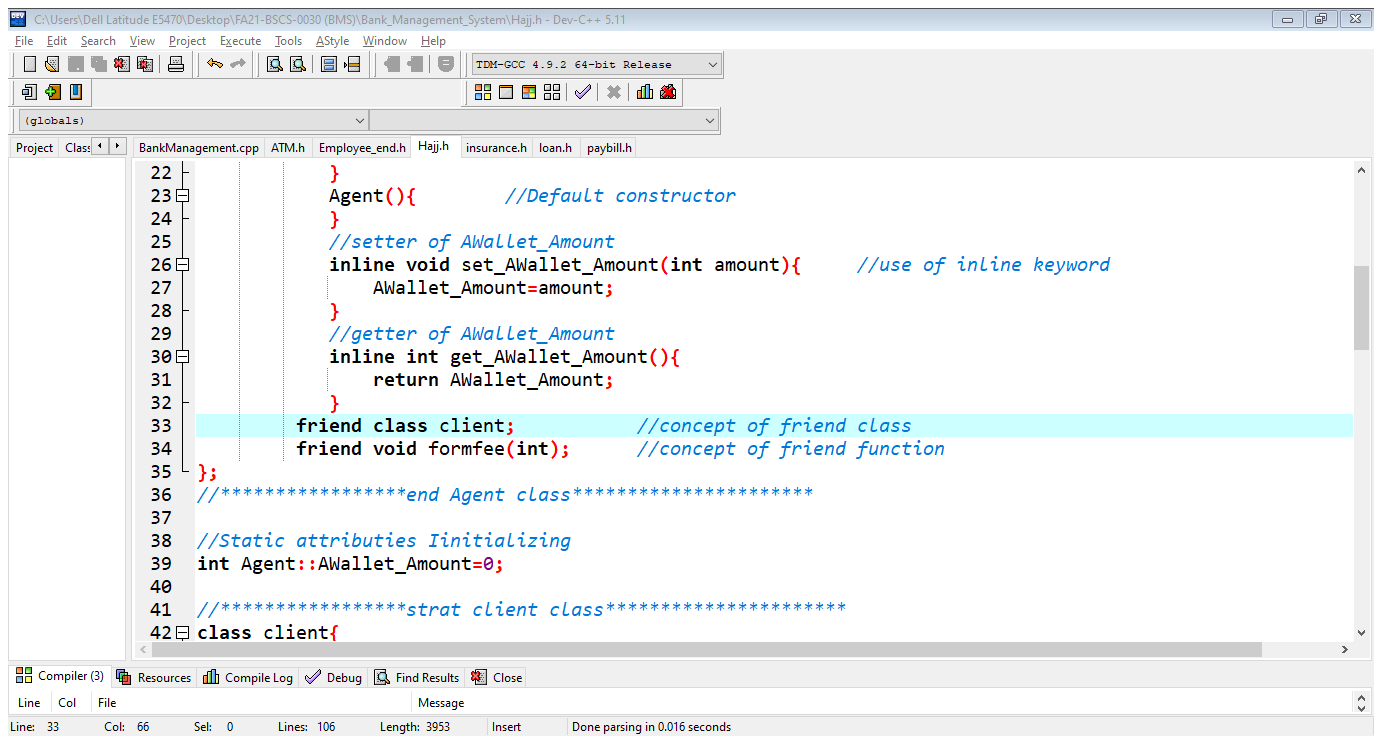
1. Abstraction



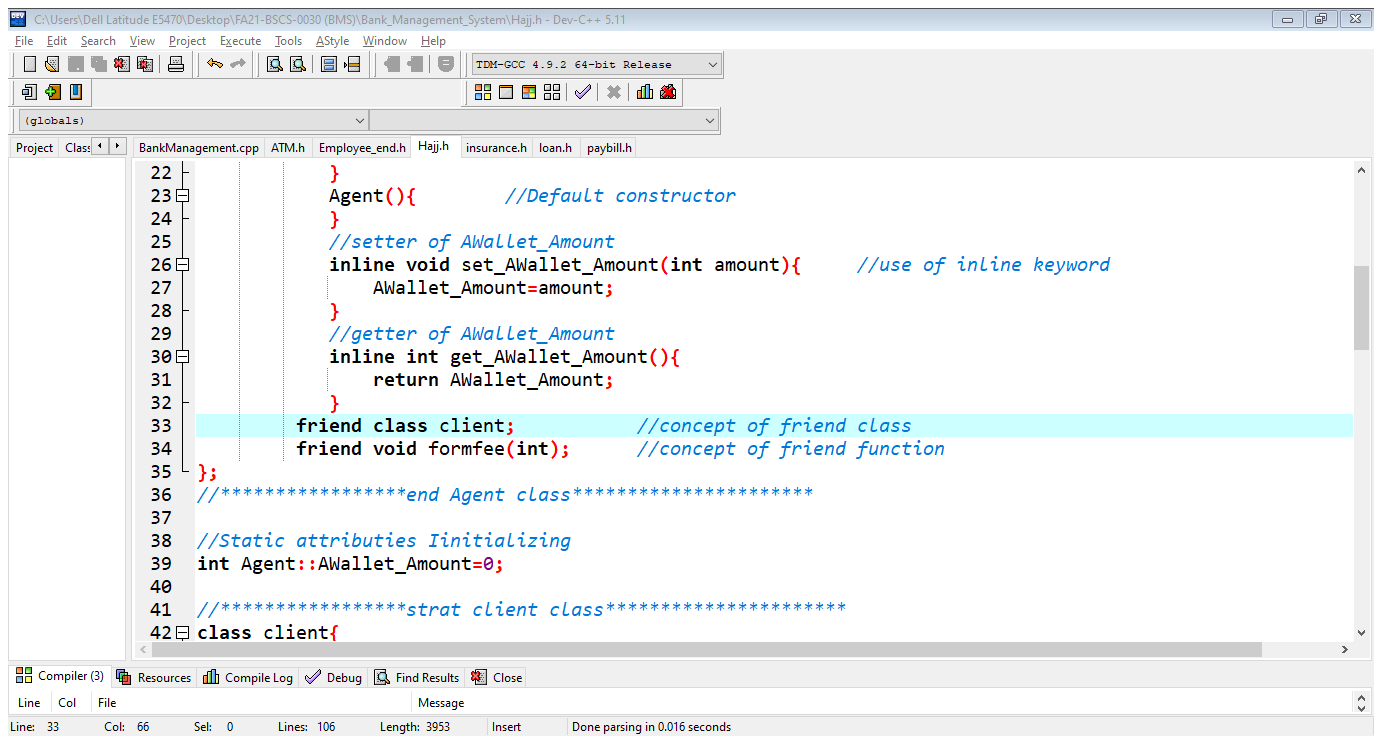
1. File Handing (ifstream, ofstream, fstream)



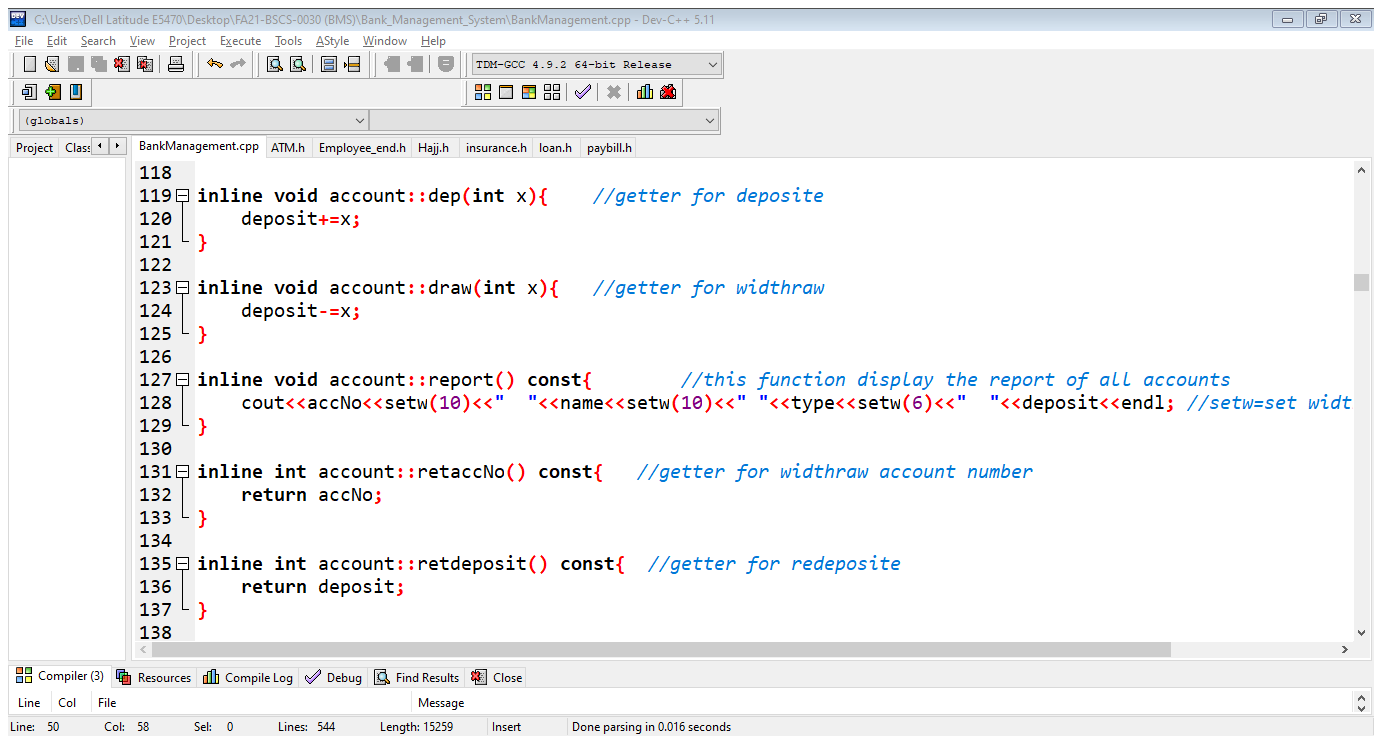
1. Friend Function



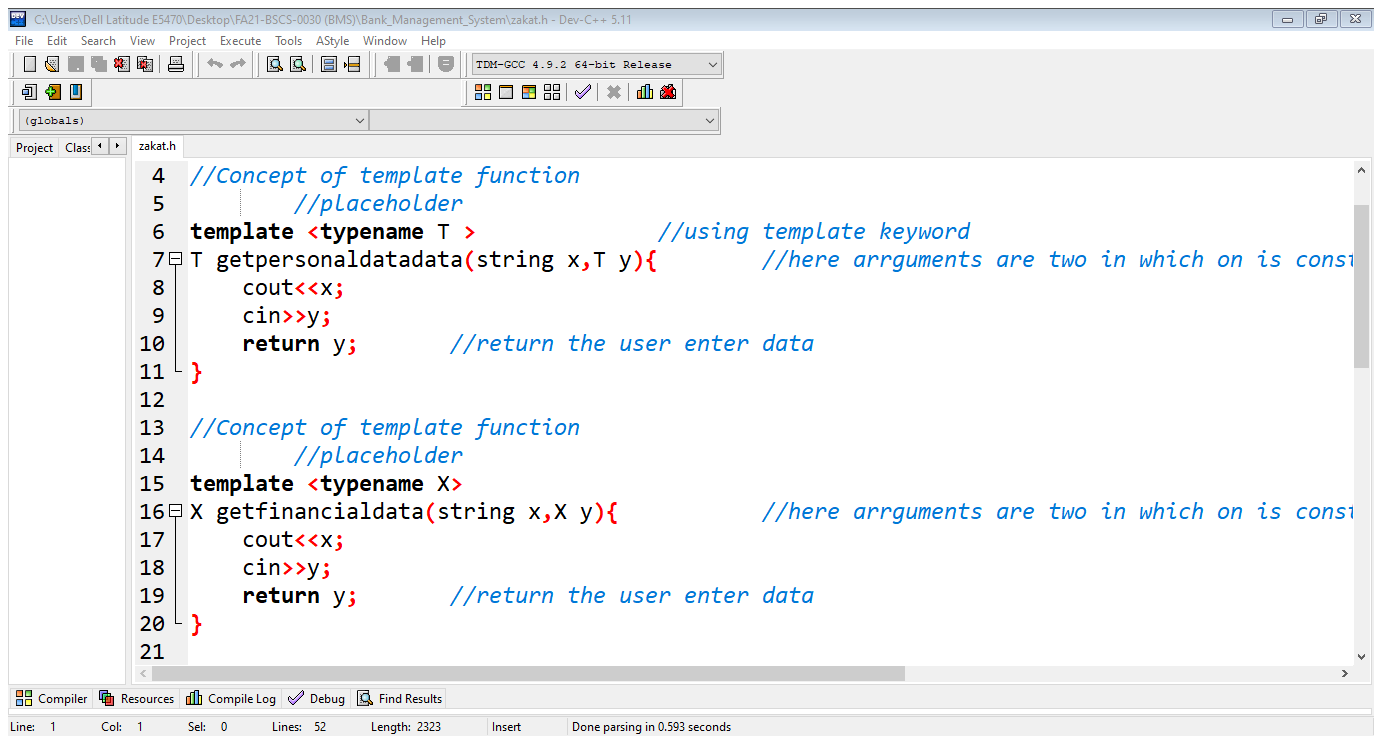
1. Friend Class



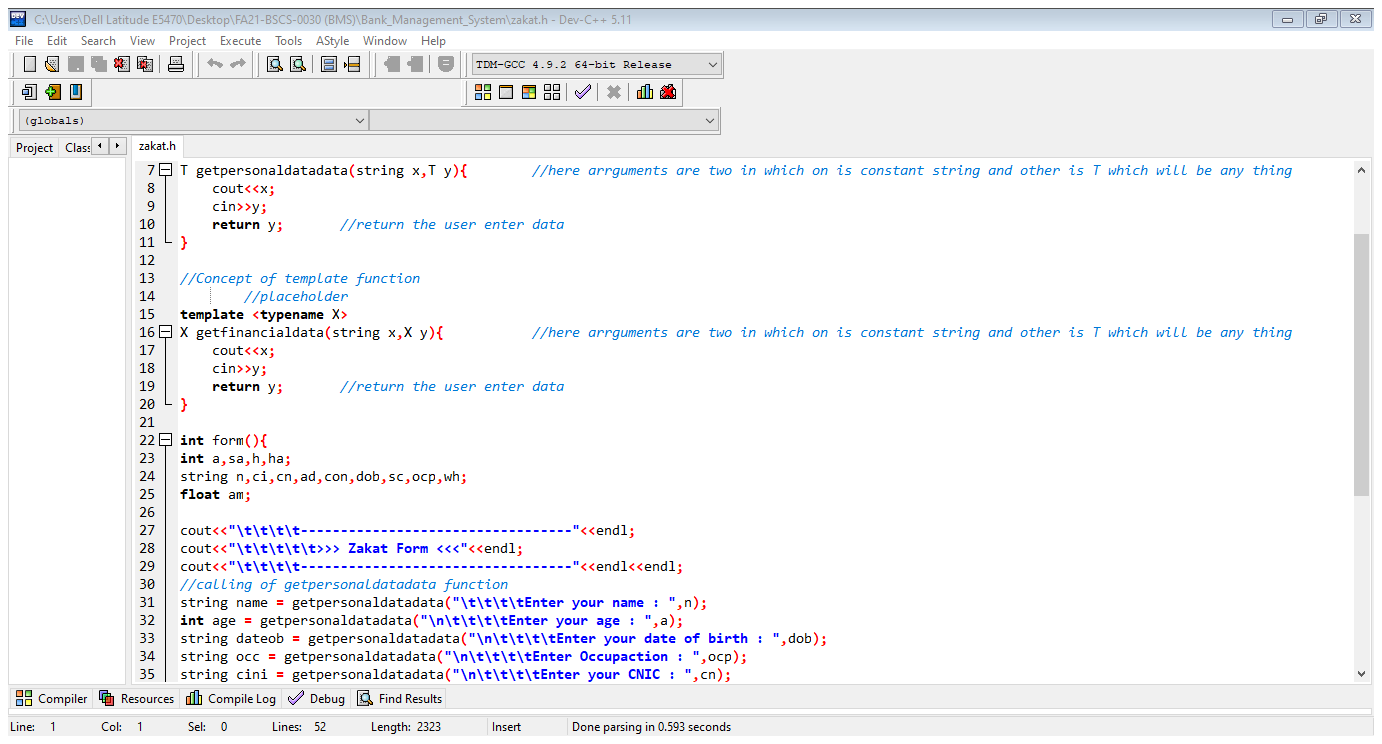
1. Inline Function



1. Templates / Generic Functions



1. Templates / Generic Class



END OF PROJCT REPORT