**Details of bank**

**A MiniProject Report Submitted in fulfillment of the requirements for the award of the degree of**

**BACHEL0R OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**BY**

**Y. Sajeev kumar -18221A05B5**

**V. Sandeep -18221A05A9**

**M.B.V.V.Satya sai -. 18221A0573**

**V.leela teja sree - 18221A05B0**

**Under the Esteemed Guidance of**

**Mrs.Ganga Bhavani,**

**Associate Professor**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE**

**(Approved by AICTE, NewDelhi & Accredited by NAAC with “A” Grade)**

**(AUTONOMUS)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE**

**(Approved by AICTE, NewDelhi & Accredited by NAAC with “A” Grade)**

**(AUTONOMUS)**

****

**DECLARATION BY THE CANDIDATES**

This is bonafied and caried by us , **Y.sajeev, v.sandeep, M.satyasai, V.leela** as the regd.nos **18221A05B5 ,18221A05A9, 18221A0573, 18221A05B0** here by declared that the project report entitled “**details of banks”** guidance of **Mrs.GangaBhavani , Associate Professor, Department of CSE,** is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of technology** in **Computer Science Engineering.**

This is a record of bonafide work carried out by me and the results embodied in this project have not been reproduced or copied from any other source and have not been submitted to any other university or institute for the award of any other degree or diploma

**PROJECT GROUP**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE**

**(Approved by AICTE, NewDelhi & Accredited by NAAC with “A” Grade)**

**(AUTONOMUS)**

****

**CERTIFICATE**

This is to certify that the project work entitle “**details of bank”** being submitted by **y.sajeevkumar(18221A05B5) , V.sandeep(18221A05A9) , M.B.V.V.satya sai (18221A0573), V.leela teja sree (18221A05B0)** in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science Engineering, B.V.C Engineering College, Odalarevu,** is a record of work carried out by them under my guidance during the academic year 2018-2019 and it has been found suitable for acceptance according to the requirement of the University.

**Mrs.GangaBhavani, Mr.Gunamani Jena, M.tech M.tech(PHD)**

**Project Supervisor Head of the Department** Associate Professor, Professor, Department of CSE. Department of CSE.

**EXTERNAL EXAMINER**

**Abstract**

In the present modern era there are so many people who don’t know the details of the banks. By our works the people can easily found and contact the bank which they want. These also help to contact the bank from your choice.

Here we are go to given some banks that are in Andhra pradesh and TELANGANA . These are helpful to Every of the worker and every person from a bank employee to the farmer. The things that no one had no idea like IFSC code is the main code to transfer money through the NEFT and RGTS transaction.

Contents

Abstract\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_02

Contents\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_03

INTRODUCTION

Overview of C++4-9

Strings \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_9-13

Files\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_13-17

DESIGN AND DEVELOPMENT

Implementation17-21

System Design22-23

TESTING AND ANALYSIS

Test Results24-27

**SOURCE code\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 28-30**

**Future works \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_31**

**Introduction**

**Introduction to C++ :-**

“C++ is a statically-typed, free-form, (usually) compiled, multi-paradigm, intermediate-level general-purpose middle-level programming language.”

In simple terms, C++ is a sophisticated, efficient and a general-purpose programming language based on C. It was developed by Bjarne Stroustrup in 1979.

Many of today’s operating systems, system drivers, browsers and games use C++ as their core language. This makes C++ one of the most popular languages today.

Since it is an enhanced/extended version of C programming language, C and C++ are often denoted together as C/C++.

**Reasons for learning C++**

This is one of those questions you need to ask before starting any programming language. It helps you understand the scope of the language, the real world usability and how far you can get with it in terms of support. Here are 5 reasons why you should learn C++.

C++ is irreplaceable With the use of C++ in development of modern games, operating systems, browsers, and much more, it is safe to say that C++ is irreplaceable. Many major applications like

Adobe Products like Photoshop, Illustrator, InDesign

Amazon - one of the biggest e-commerce sites

Autodesk products for Computer Aided Design

Facebook - social networking site are heavy C++ centric products.

**•C++ is irreplaceable**

With the use of C++ in development of modern games, operating systems, browsers, and much more, it is safe to say that C++ is irreplaceable.

Many major applications like

Adobe Products like Photoshop, Illustrator, InDesign

Amazon - one of the biggest e-commerce sites

Autodesk products for Computer Aided Design

Facebook - social networking site are heavy C++ centric products.

Moreover, the fact that there’s a huge community improving C++ on every iteration means that it is only expected to be used even more in the coming future.

**•You learn the internal architecture of a computer**

Since, C++ is a middle level language, you will write code that interacts directly with the internal hardware of the computer.

You’ll learn how the computer memory really works, how information is stored in them, how you can retrieve them and so on.

It is sure to expand your knowledge on the architecture of the computer.

**•Over 600,000 C++ repositories on Github**

Github, the leading open source collaboration platform, has over 600,000 repositories for C++ alone.

This metric itself proves the worth of C++ in the open source community as well.

Be it gaming, graphics, windows applications, you can find tons of great open source projects extensively used today. And, you can always create your own.

**•60% StackOverflow Answer rate and active community**

Likewise, with over 400,000 C++ questions asked on StackOverflow, the number one Q&A platform for developers, more than 60% questions have been answered.

The number of questions asked and the percentage of them answered shows the interest and active support for C++ today.

So, you can expect many great developers to help you solve real-life problems using C++.

**•C++ job opportunities and salary**

C++ developers can expect an average of yearly $100,000 salary with over 7,700 jobs advertised every month.

The requirement of jobs comes mostly from game development, rendering engines and the windows applications.

***Strings:-***

In computer programming, a string is traditionally a sequence of characters, either as a literal constant or as some kind of variable. The latter may allow its elements to be mutated and the length changed, or it may be fixed (after creation). A string is generally considered as a data type and is often implemented as an array data structure of bytes (or words) that stores a sequence of elements, typically characters, using some character encoding. String may also denote more general arrays or other sequence (or list) data types and structures.

Depending on the programming language and precise data type used, a variable declared to be a string may either cause storage in memory to be statically allocated for a predetermined maximum length or employ dynamic allocation to allow it to hold a variable number of elements.

When a string appears literally in source code, it is known as a string literal or an anonymous string

In formal languages, which are used in mathematical logic and theoretical computer science, a string is a finite sequence of symbols that are chosen from a set called an alphabet.

**String datatypes**

A string datatype is a datatype modeled on the idea of a formal string. Strings are such an important and useful datatype that they are implemented in nearly every programming language. In some languages they are available as primitive types and in others as composite types. The syntax of most high-level programming languages allows for a string, usually quoted in some way, to represent an instance of a string datatype; such a meta-string is called a literal or string literal.

).

**Implementations**

Some languages like C++ implement strings as templates that can be used with any datatype, but this is the exception, not the rule.

Some languages, such as C++ and Ruby, normally allow the contents of a string to be changed after it has been created; these are termed mutable strings. In other languages, such as Java and Python, the value is fixed and a new string must be created if any alteration is to be made; these are termed immutable strings.

Strings are typically implemented as arrays of bytes, characters, or code units, in order to allow fast access to individual units or substrings—including characters when they have a fixed length. A few languages such as Haskell implement them as linked lists instead.

Some languages, such as Prolog and Erlang, avoid implementing a dedicated string datatype at all, instead adopting the convention of representing strings as lists of character codes.

**Representations**

Representations of strings depend heavily on the choice of character repertoire and the method of character encoding. Older string implementations were designed to work with repertoire and encoding defined by ASCII, or more recent extensions like the ISO 8859 series. Modern implementations often use the extensive repertoire defined by Unicode along with a variety of complex encodings such as UTF-8 and UTF-16.

The term byte string usually indicates a general-purpose string of bytes, rather than strings of only (readable) characters, strings of bits, or such. Byte strings often imply that bytes can take any value and any data can be stored as-is, meaning that there should be no value interpreted as a termination value.

Most string implementations are very similar to variable-length arrays with the entries storing the character codes of corresponding characters. The principal difference is that, with certain encodings, a single logical character may take up more than one entry in the array. This happens for example with UTF-8, where single codes (UCS code points) can take anywhere from one to four bytes, and single characters can take an arbitrary number of codes. In these cases, the logical length of the string (number of characters) differs from the physical length of the array (number of bytes in use). UTF-32 avoids the first part of the problem.

***Files:-***

File streams are a lot like cin and cout

In Standard C++, you can do I/O to and from disk files very much like the ordinary console I/O streams cin and

cout. The object cin is a global object in the class istream (input stream), and the global object cout is a member

of the class ostream (output stream). File streams come in two flavors also: the class ifstream (input file stream)

inherits from istream, and the class ofstream (output file stream) inherits from ostream. Thus all of the member

functions and operators that you can apply to an istream or ostream object can also be applied to ifstream and

ofstream objects. However, file streams have some additional member functions and integrated in a way determined by your computer's Operating System

(OS), which is responsible for keeping track of the information. If the file is deleted, moved, expanded, contracted,

etc., the OS keeps track of exactly where it is on the disk and how much of it there is. The C/C++ facilities for

working with disk files actually call OS subroutines to do the work.

Instead of creating and then opening the file streams in separate statements, you can use a constructor that takes the

file name as an argument; after doing the normal initializations, the constructor completes the initialization by

opening the named file. The above four statements would then condense down to two:

ifstream my\_input\_file("input\_data"); // create and open

ofstream my\_output\_file("output\_data");

In both ways of opening a file, you can specify the file path or file name either with a C-string array or literal (as the

above examples do), or in C++11 with a std::string. For example:

string filename;

cin >> filename;

ifstream my\_input\_file(filename);

When opening files, especially input files, is it critical to test for whether the open operation succeeded. File stream

errors are discussed in more detail below. But for now, here is one way of doing this test using a member function

that returns true if the file was successfully opened:

if (my\_input\_file.is\_open()) {

! ! // can continue, file opened correctly

! ! }

Now that the file streams are open, using them could not be simpler. We can read and write variable values from/to

the streams using the stream input and output operators just like with cin and cout. For example, to read an integer

and a double from the input file:

my\_input\_file >> int\_var >> double\_var;

To output to the file:

my\_output\_file << "The integer is " << int\_var << endl;

The contents of the input file are processed just like you were typing them in via cin, and the output going into the

file looks identical to what is written on your display with cout. You can read or write as much information from the

file as is appropriate. Since ifstream and ofstream inherit from istream and ostream, your definitions of overloaded

operators << or >> for ostream and istream will automatically work for file streams.

**Implementation**

**Files:-**

It consists of two major components:

• algorithm and

• data structures.

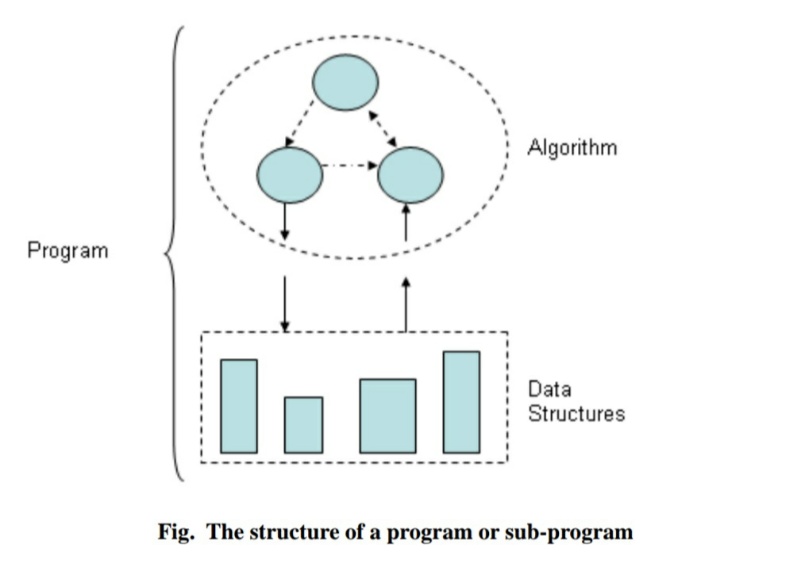
• The algorithm takes care of the rules and

procedures required for solving the problem

and the data structures contain the data.

• The data is manipulated by the procedures for

achieving the goals of the program as shown

in Fig

A data structure is volatile by nature in the

sense that its contents are lost as soon as the

execution of the program is over.

• Similarly, an object also loses its states after

the program is over.

If we want to permanently store our data or

want to create persistent objects then it

becomes necessary to store the same in a

special data structure called file.

• The file can be stored on a second storage

media such as hard disk. In fact, vary large

data is always stored in a file.

**File**

“A file is a logical collection of records where each record

consists of a number of items known as fields”.

The records in a file can be arranged in the following three ways:

• Ascending/Descending order: The records in the file can be

arranged according to ascending or descending order of a key field..

• Alphabetical order: If the key field is of alphabetic type then the

records are arranged in alphabetical order.

**FILES AND STREAMS**

In C++, a stream is a data flow from a source to a sink. The

sources and sinks can be any of the input/output

devices or files.

For input and output, there are two different streams

called input stream and output stream.

Stream Description

cin standard input stream

cout standard output stream

cerr standard error stream

The standard source and sink are keyboard and monitor

screen respectively

**ifstream**: It is the input file stream class. Its member

function open( ) associates the stream with a specified

file in an input mode.

In addition to open(), ifstream class inherits the following functions

from istream class.

(i) get( ) (ii) getline( ) (iii) read( ) (iv) seekg( ) (iv) tellg( )

**ofstream** : It is the output file stream class. Its member

function open( ) associates the stream with a specified

file in an output mode.

In addition to open(), ofstream inherits the following functions from

ostream class

(i) put( ) (ii) write( ) (iii) seekp( ), (iv) tellp( )

**fstream** : It supports files for simultaneous input and

output. It is derived from ifstream, ofstream and

iostream classes.

The functions associated with this stream are

1. open : This associates the stream with a specified file.

2. close : It closes the stream.

3. close all : It closes all the opened streams

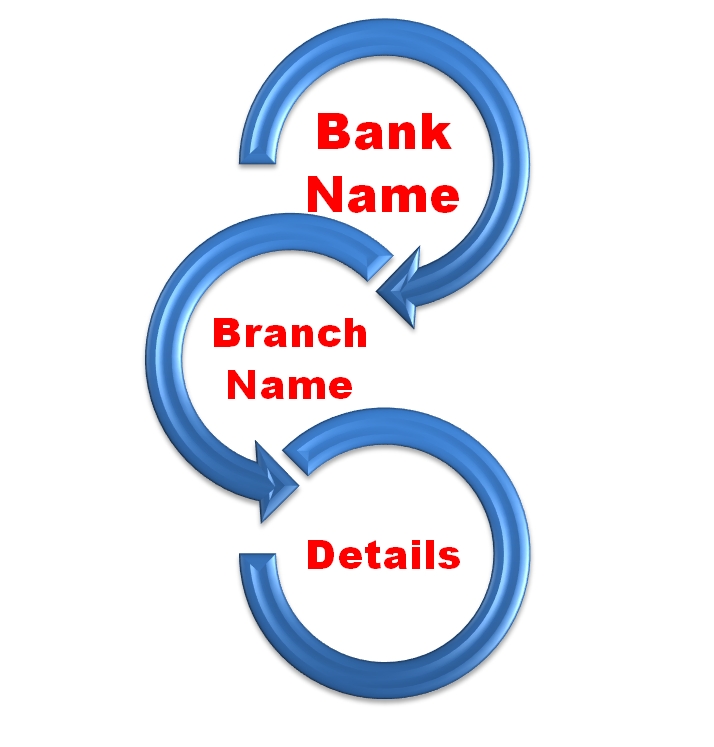
4. seekg : Sets current `get' position in a stream

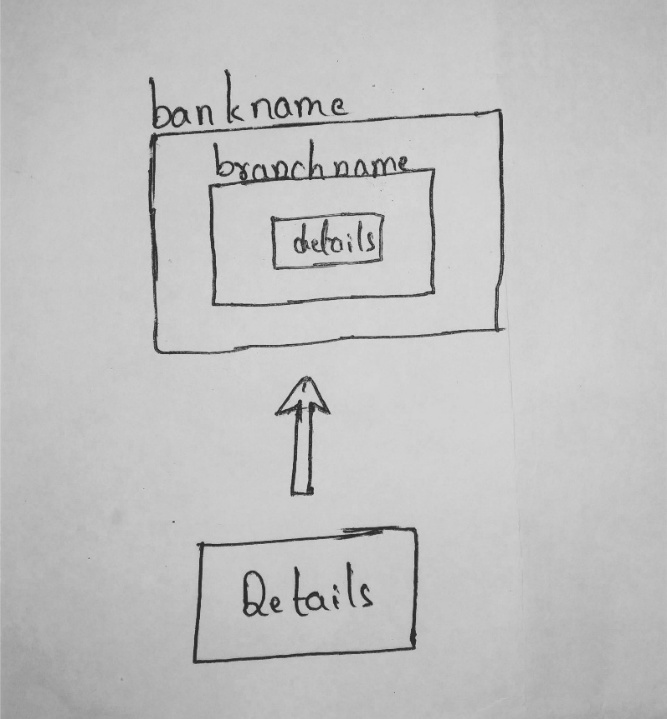
5. seekp : Sets current `put' position in a stream

6. tellg : Returns the current `get' position in a stream

7. tellp : Returns the current `put' position in a stream

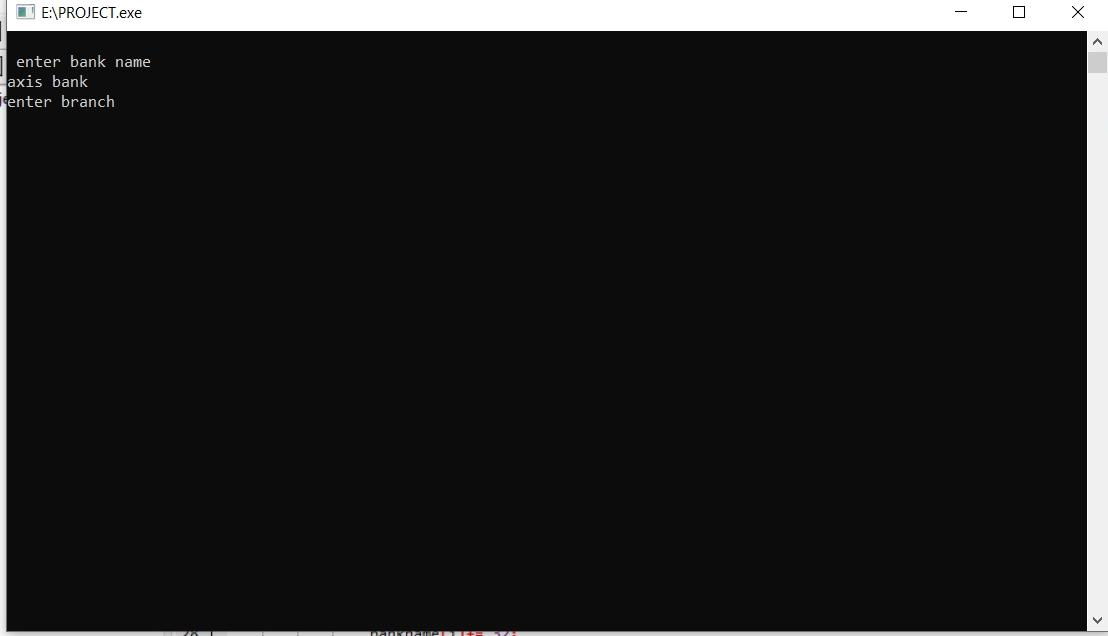
**Design:--Flow of path**

**Back end development**

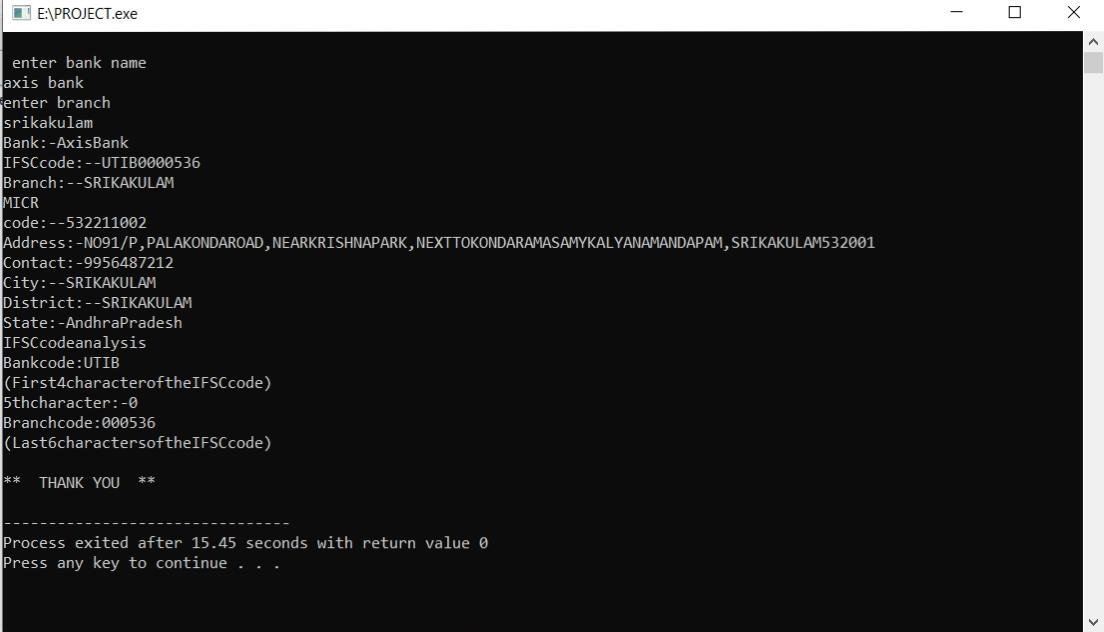
****

**Test cases:--**

This is the page that you can have on the execution path

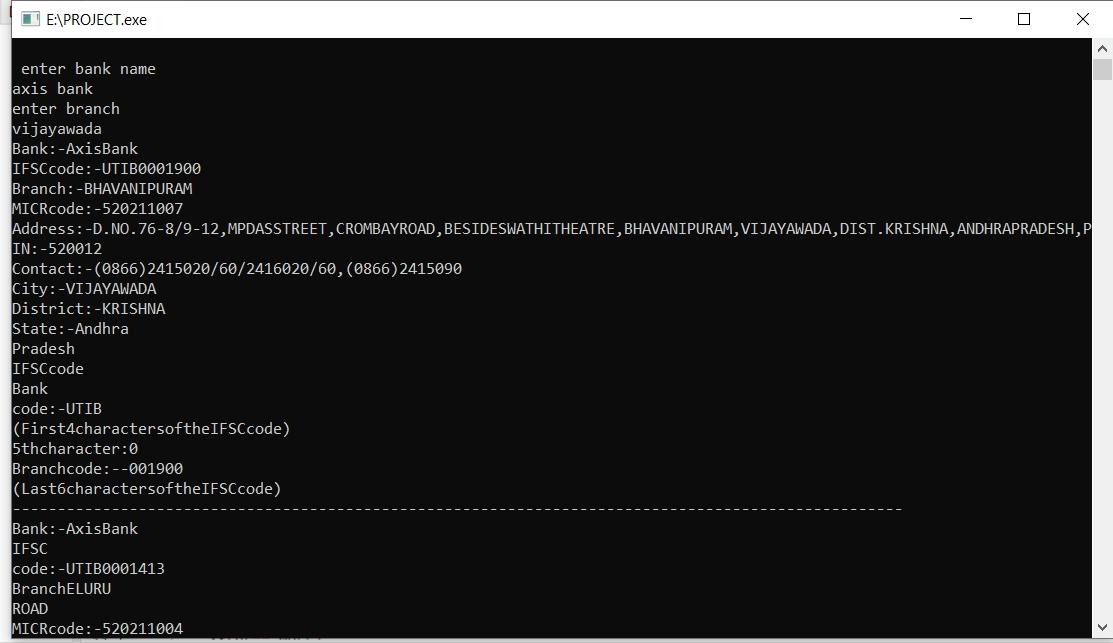


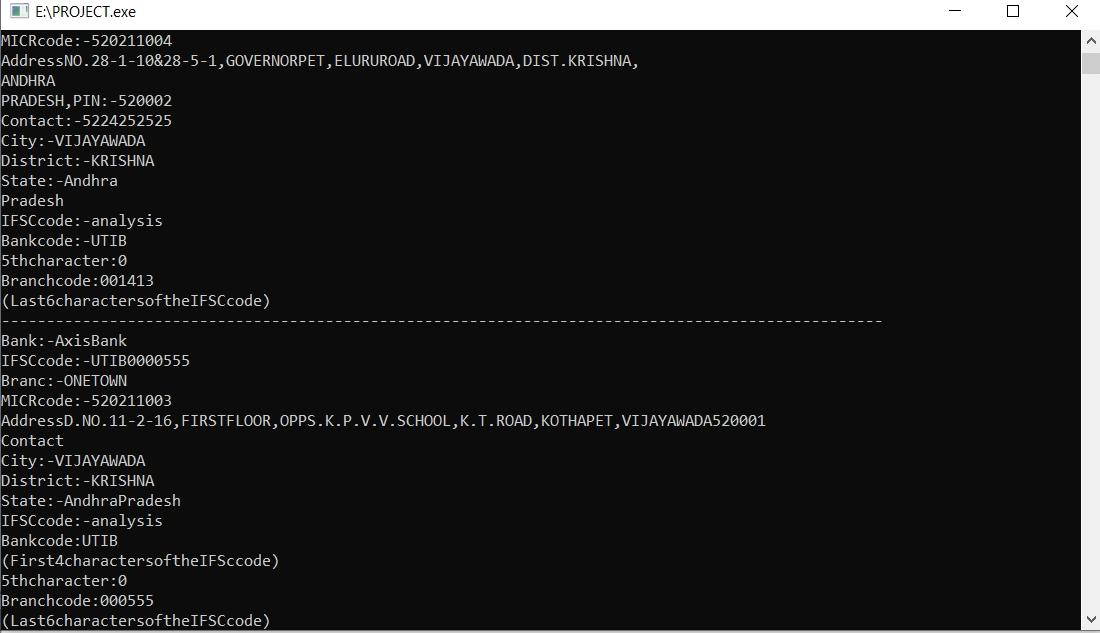
After entering your detailed information it can directly print the details of the bank

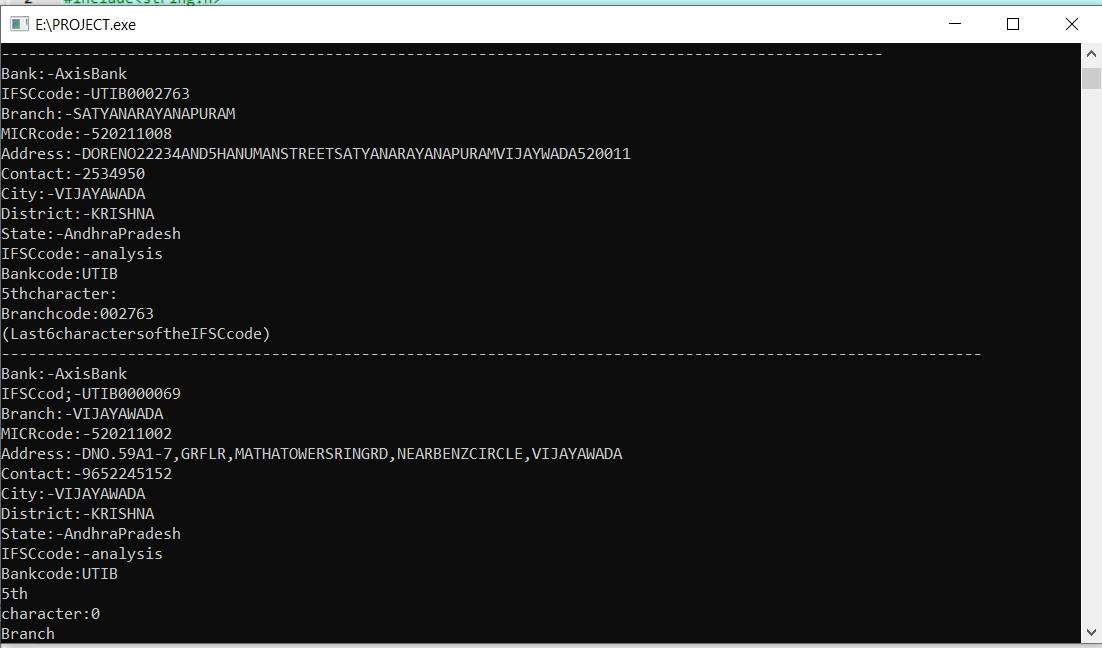


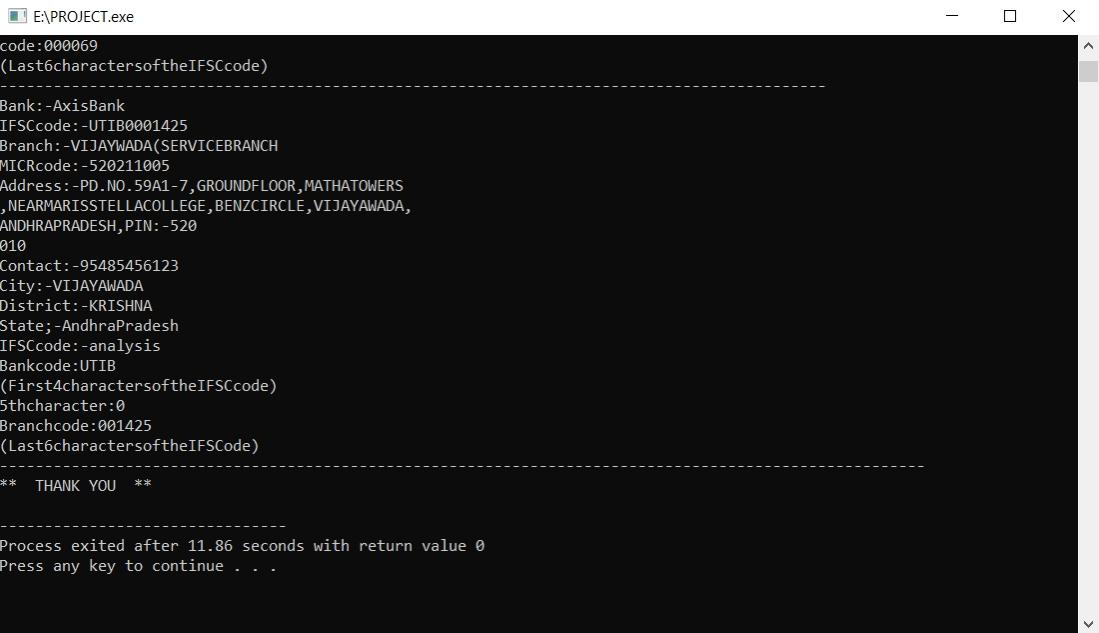
If there are more than a bank in particular area then it will print that more information also

See that also









As everyone can observance it can be separated with the symbols as a line to identify the different things or places in that place

**Source code:--**

#include<iostream>

#include<string.h>

#include<fstream>

#include<dirent.h>

#include<stdio.h>

#include<conio.h>

using namespace std;

ifstream details;

int flag,c;

static int count=0;

int main()

{

do

{

char s1[100]= "E:/clg/sem 1/c++/bank/";

char s3[100]="\0";

char bankname[100];

cout<<"\n enter bank name \n";

gets(bankname);

char branchname[100];

cout<<"enter branch"<<endl;

gets(branchname);

strcat(branchname,".txt");

for(int i=0;i<sizeof(bankname);i++)

{

if(bankname[i]>=65 && bankname[i]<=90)

{

bankname[i]+= 32;

}

}

struct dirent \*d,\*d1;

DIR \*dr = opendir(s1);

if(dr== NULL)

{

cout<<"could not open";

}

while((d= readdir(dr))!= NULL)

{

if(strcmp((d->d\_name),bankname) == 0)

{

flag=1;

strcat(s1,bankname);

DIR \*dr1 = opendir(s1);

if(dr1 == NULL)

{

cout<<"could not open";

}

while((d1= readdir(dr1))!= NULL)

{

if(strcmp(d1->d\_name,branchname)==0)

{

flag=2;

count=2;

strcat(s1,"/");

strcat(s1,branchname);

strcat(s3,s1);

details.open(s3);

while(!details.eof())

{

string g;

details >> g;

cout << g<<endl;

}

}

if(count==2)

{

break;

}

}

closedir(dr1);

}

}

if(flag == 1)

{

cout<<"\*\*\*\*\*\* SORRY WE DONT SERVE IN THE BRANCH YOU ENTERED\*\*\*\*\*\*\*\*\*\*"<<endl;

}

else if(flag == 2)

{

cout<<"\*\*\*\*\*\* THANK YOU \*\*\*\*\*\*"<<endl;

}

else

{

cout<<"\*\*\*\* ENTER CORRECT INFO \*\*\*\*\*\*"<<endl;

}

closedir(dr);

c++;

if(count == 1)

{

cout<<"enter correct bank and branch details again"<<endl;

}while(count!=2 && c<=3);

}

**Future works**

**•**Extend to the every bank in the AP and telangana

•To search a bank by its IFSC code given by user or printing details of bank by IFSC code