qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm

|  |
| --- |
| [Type the document title]  [Pick the date] |

## BANK MANAGEMENT SYSTEM

##### PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

##### BACHELOR OF TECHNOLOGY

##### (COMPUTER SCIENCE AND ENGINEERING)

###### SUBMITTED BY

(Divya Dubey – 2100460100048) September 2022



##### Dr. APJ Abdul Kalam Technical University, Uttar Pradesh

###### LUCKNOW, INDIA

**CERTIFICATE**

This is to certify that the Project Report on the topic of "**BANK MANAGEMENT SYSTEM**" is submitted by **Divya Dubey CSE 2nd year (2100460100048)** in fulfillment for the award of degree of **BACHELOR OF TECHNOLOGY** in **Computer Science and Engineering** has been found satisfactory and is approved for submission.

1. Embodies the work of the candidates themselves,
2. Has duly been completed, and
3. Is up to the desired standard both in respect of contents and language for being referred to the examiners.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Mr. Millind Bhatt** **Dr. Saurabh Singh**

Associate Professor Head of Department CSE

(CSE) Department MPEC

MPEC

**ACKNOWLEDGEMENT**

It gives us the great sense of pleasure to present the report of the B.Tech project undertaken during B.Tech 2ndyear. I owe special debt of gratitude to **Dr. Saurabh Singh** (Head of Department in Computer Science and Engineering).

The satisfaction that accompanies the successful completion of any task would be incomplete without mention of people whose ceaseless cooperation made in possible, whose constant guidance and encouragement crown all effort with success.

We are grateful to our project guide **Mr. Abhishek Singh Sengar** for the guidance, inspiration and constructive suggestions that help us in the preparation in this project.

I also thank my college and colleagues who helped us in successful completion of the project.

**DIVYA DUBEY**

**(2100460100048)**

**TABLE OF CONTENTS**

#### TOPIC

1.Introduction of the Problem

2.Objective (problem statement)

3.H/W, S/W Requirements

4.Flowchart

5.Literature Review

6.Technology Use

7.Output Screen shots

8.Conclusion

9.Future Scope

10.References

11.Appendix

**INTRODUCTION**

The **Bank Management System project in C** is a consoled based application and created using c programming language. This project keeps track of a client’s bank account. This project demonstrates the operation of a banking account system and covers the essential functions of bank management software like:

1.Create Account

2.Update information of existing account

3.For transactions

a) deposit

b) withdraw

4.Check the details of existing account

5.Removing existing account

6.View customer’s list

**C PROGRAMMING LANGUAGE**

**C** is a general-purpose programming language that is extremely popular, simple, and flexible to use. It is a structured programming language that is machine-independent and extensively used to write various applications, Operating Systems like Windows, and many other complex programs like Oracle database, Git, Python interpreter, and more.

It is said that ‘C’ is a god’s programming language. One can say, C is a base for the programming. If you know ‘C,’ you can easily grasp the knowledge of the other programming languages that uses the concept of ‘C’

It is essential to have a background in computer memory mechanisms because it is an important aspect when dealing with the C programming language.

In 1972, a great computer scientist Dennis Ritchie created a new programming language called ‘C’ at the Bell Laboratories.

 ‘C’ programming language contains all the features of these languages and many more additional concepts that make it unique from other languages.

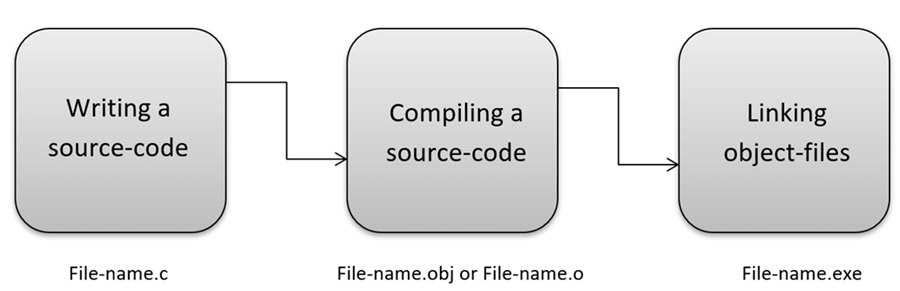
‘C’ is a powerful programming language which is strongly associated with the UNIX operating system. Even most of the UNIX operating system is coded in ‘C’.

Initially ‘C’ programming was limited to the UNIX operating system, but as it started spreading around the world, it became commercial, and many compilers were released for cross-platform systems. Today ‘C’ runs under a variety of operating systems and hardware platforms. As it started evolving many different versions of the language were released. At times it became difficult for the developers to keep up with the latest version as the systems were running under the older versions.

To assure that ‘C’ language will remain standard, American National Standards Institute (ANSI) defined a commercial standard for ‘C’ language in 1989. Later, it was approved by the International Standards Organization (ISO) in 1990. ‘C’ programming language is also called as ‘ANSI C’.

## How C Programming Language Works?

C is a compiled language. A compiler is a special tool that compiles the program and converts it into the object file which is machine readable. After the compilation process, the linker will combine different object files and creates a single executable file to run the program. The following diagram shows the execution of a ‘C’ program

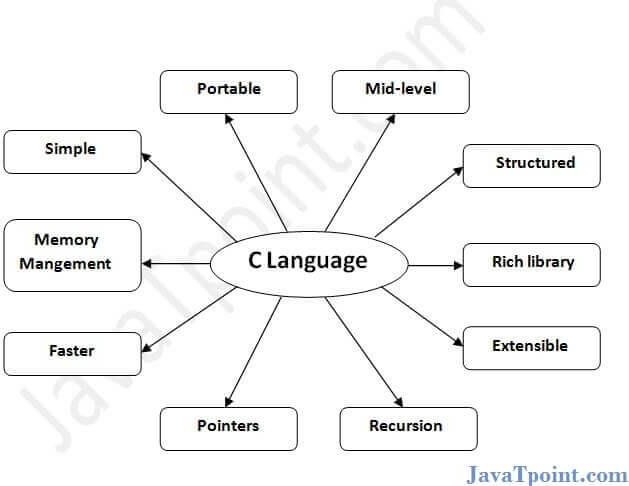


Nowadays, various compilers are available online, and you can use any of those compilers. The functionality will never differ and most of the compilers will provide the features required to execute both ‘C’ and ‘C++’ programs.

Following is the list of popular compilers available online:

* Clang compiler
* MinGW compiler (Minimalist GNU for Windows)
* Portable ‘C’ compiler
* Turbo C

##### Some Features of C Programming



###### Simple

C is a simple language in the sense that it provides a structured approach(to break the problem into parts), the rich set of library functions, data types, etc.

1. Machine Independent or Portable

Unlike assembly language, c programs can be executed on differentmachines with some machine specific changes. Therefore, C is a machine independent language.

1. Mid-level programming language

Although, C is intended to do low-level programming. It is used to develop system applications such as kernel, driver, etc. It also supports the features of a high-level language. That is why it is known as mid-level language.

1. Structured programming language

C is a structured programming language in the sense that we can break the program into parts using functions. So it is easy to understand and modify. Functions also provide code reusability.

1. Rich Library

C provides a lot of inbuilt functionsthat make the development fast.

1. Memory Management

It supports the feature of dynamic memory allocation. In C language, we can free the allocated memory at any time by calling the free()function.

1. Speed

The compilation and execution time of C language is fast since there are lesser inbuilt functions and hence the lesser overhead.

1. Pointer

C provides the feature of pointers. We can directly interact with the memory by using the pointers. We can use pointers for memory, structures, functions, array, etc.

1. Recursion

In C, we can call the function within the function. It provides code reusability for every function. Recursion enables us to use the approach of backtracking.

1. Extensible

C language is extensible because it can easily adopt new features.

# **C Hello World! First Program:**

Here,is a Hello World program in C

#include<stdio.h> //Pre-processor directive

void main () //main function declaration

{

printf("Hello World"); //to output the string on a display

getch (); //terminating function

}

Here is the code explanation:

### Pre-processor directive

#include is a pre-processor directive in ‘C.’

**#include <stdio.h>**, stdio is the library where the function **printf**is defined**.**printfis used for generating output. Before using this function, we have to first include the required file, also known as a header file (.h).

You can also create your own functions, group them in header files and declare them at the top of the program to use them. To include a file in a program, use pre-processor directive

#include <file-name>.h

File-name is the name of a file in which the functions are stored. Pre-processor directives are always placed at the beginning of the program.

### The main function

The main function is a part of every ‘C’ program. We can represent the main function in various forms, such as:

* main()
* int main()
* void main()
* main(void)
* void main(void)
* int main(void)

The empty parentheses indicate that this function does not take any argument, value or a parameter. You can also represent this explicitly by placing the keyword void inside the parentheses. The keyword void means the function does not return any value, in this case, the last statement is always getch ().

#include<stdio.h> //Pre-processor directive

int main() //main function declaration

{

printf("Hello World"); //to output the string on a display

return 0; //terminating function

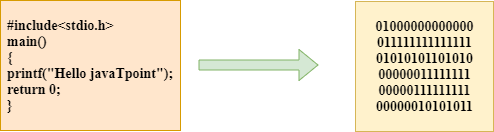
}

In the above example, the keyword int means the function will return an integer value. In this case, the last statement should always return 0.

# **Compilation process in c**

## What is a compilation?

The compilation is a process of converting the source code into object code. It is done with the help of the compiler. The compiler checks the source code for the syntactical or structural errors, and if the source code is error-free, then it generates the object code.

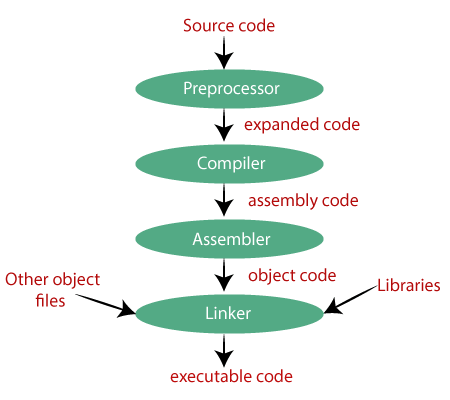


The c compilation process converts the source code taken as input into the object code or machine code. The compilation process can be divided into four steps, i.e., Pre-processing, Compiling, Assembling, and Linking.

The preprocessor takes the source code as an input, and it removes all the comments from the source code. The preprocessor takes the preprocessor directive and interprets it. For example, if **<stdio.h>,** the directive is available in the program, then the preprocessor interprets the directive and replace this directive with the content of the **'stdio.h'** file.

The following are the phases through which our program passes before being transformed into an executable form:

* **Preprocessor**
* **Compiler**
* **Assembler**
* **Linker**



### **Preprocessor**

The source code is the code which is written in a text editor and the source code file is given an extension ".c". This source code is first passed to the preprocessor, and then the preprocessor expands this code. After expanding the code, the expanded code is passed to the compiler.

### **Compiler**

The code which is expanded by the preprocessor is passed to the compiler. The compiler converts this code into assembly code. Or we can say that the C compiler converts the pre-processed code into assembly code.

### **Assembler**

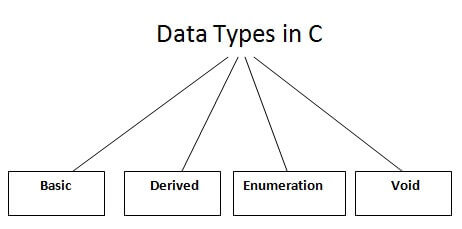
The assembly code is converted into object code by using an assembler. The name of the object file generated by the assembler is the same as the source file. The extension of the object file in DOS is '.obj,' and in UNIX, the extension is 'o'. If the name of the source file is **'hello.c',** then the name of the object file would be 'hello.obj'.

### **Linker**

Mainly, all the programs written in C use library functions. These library functions are pre-compiled, and the object code of these library files is stored with '.lib' (or '.a') extension. The main working of the linker is to combine the object code of library files with the object code of our program. Sometimes the situation arises when our program refers to the functions defined in other files; then linker plays a very important role in this. It links the object code of these files to our program. Therefore, we conclude that the job of the linker is to link the object code of our program with the object code of the library files and other files. The output of the linker is the executable file. The name of the executable file is the same as the source file but differs only in their extensions. In DOS, the extension of the executable file is '.exe', and in UNIX, the executable file can be named as 'a.out'. For example, if we are using printf() function in a program, then the linker adds its associated code in an output file.

# **Data Types in C**

A data type specifies the type of data that a variable can store such as integer, floating, character, etc.



There are the following data types in C language.

|  |  |
| --- | --- |
| **Types** | **Data Types** |
| Basic Data Type | int, char, float, double |
| Derived Data Type | array, pointer, structure, union |
| Enumeration Data Type | enum |
| Void Data Type | void |

# **Keywords in C**

A keyword is a **reserved word**. You cannot use it as a variable name, constant name, etc. There are only 32 reserved words (keywords) in the C language.

A list of 32 keywords in the c language is given below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Auto | break | case | char | const | continue | default | do |
| double | else | enum | extern | float | for | goto | if |
| Int | long | register | return | short | signed | sizeof | static |
| struct | switch | typedef | union | unsigned | void | volatile | while |

### Applications of C programming language

* + C language is used for creating computer applications. It is used in writing Embedded software, Firmware for various electronics, industrial and communications products which use micro- controllers. It is also used in developing verification software, test code, simulators etc. for various applications and hardware products. There are so many applications of C , here are some of them:

###### Operating Systems: -

What is better than writing your own operating system? And yes, with the help of the C programming language, you can write your own operating system. Windows Kernel, Linux Kernel and Apple’s OS X kernel are mostly written in C.

###### GUI: -

It stands for Graphical User Interface. The C programming language also helps in developing popular adobe software’s like Photoshop, Premier Pro, Illustrator etc.

###### Embedded Systems: -

In daily life, we use different embedded systems like coffee machines, microwaves, climate control systems etc. These all are mostly programmed in C.

###### Database: -

The C programming language helps in developing the popular database management system, MySQL.

###### Ease of Computation: -

C provides faster computation in programs. The implementation of algorithms and data structures is swift in C. With the help of C, you can perform high degree calculations such as MATLAB, Mathematica etc.

###### Gaming: -

C programming is relatively faster than Java or Python. It has been used in various gaming applications and graphics. C programming language also helps in creating many popular childhood games like Tic-Tac-Toe, The Snake game etc.

###### Development of New languages: -

Due to the fast execution and simplicity, many languages like Java, C++, Python, PHP, PERL, JavaScript, etc were influenced by the development of C. In Python, C is used for building standard libraries. The syntax and control structures of PERL, PHP and C++ are based upon the C programming language.

###### Google: -

In the Google open-source community, the projects are being handled by C/C++. And C/C++ also helped in developing google file system and chromium browser.

**OBJECTIVE OF THE PROJECT**

The Bank Management System is an application for maintaining a person's account in a bank. The purpose of this bank management system project is to organize and manage the bank tasks digitally. In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System.

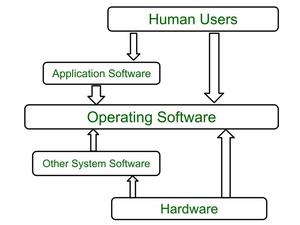
* Every successful banker has to perform managerial responsibilities along with technical banking activities. Although a bank is a financial institution like other businesses, its main objective is to maximize its wealth by earning profit. Bank business is different from other types of businesses.
* Other businesses can transfer/shift their goods or services from factory to office, or other far places even can export to foreign countries. But banks can transfer funds or extend services only after fulfilling the demand for a loan of the area where the office or branch of that bank is situated.
* High-quality service can be offered by efficient management, and efficient management can be ensured by efficient organization management. So, professional management is impossible without crystallizing the authority & responsibility of all the personnel employed in a bank.

**HARDWARE REQUIREMENT**

**WINDOW 7 AND LATER OPERATING SYSTEM:**

Operating System lies in the category of system software. It basically manages all the resources of the computer. An operating system acts as an interface between the software and different parts of the computer or the computer hardware. The operating system is designed in such a way that it can manage the overall resources and operations of the computer. It is a fully integrated set of specialized programs that handle all the operations of the computer. It controls and monitors the execution of all other programs that reside in the computer, which also includes application programs and other system software of the computer. Examples of the operating system are Windows, Linux, Mac OS, etc.

Diagram of the operating system:



**Objectives:**

Let us now see some of the objectives of the operating system, which are:

1. Convenient to use: One of the objectives is to make the computer system more convenient to use in an efficient manner.
2. User Friendly: To make the computer system more interactive with a more convenient interface for the users.
3. To provide easy access to users for using resources by acting as an intermediary between the hardware and its users.
4. For managing the resources of a computer.
5. Controls and Monitoring: By keeping the track of who is using which resource, granting resource requests, and mediating conflicting requests from different programs and users.
6. Providing efficient and fair sharing of resources between the users and programs.

**Characteristics:**

Let us now discuss some of the important characteristic features of operating systems:

* **Device Management:**The operating system keeps track of all the devices. So, it is also called the Input / Output controller that decides which process gets the device, when, and for how much time.
* **File Management:**It allocates and de-allocates the resources and also decides who gets the resource.
* **Job Accounting:**It keeps the track of time and resources used by various jobs or users.
* **Error-detecting Aids:**It contains methods that include the production of dumps, traces, error messages, and other debugging and error-detecting methods.

**Types of the operating system:**

* **Batch Operating System:**This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirements and groups them into batches.
* **Time-sharing operating System:**This type of operating system allows many users to share computer resources.
* **Distributed operating System:**This type of operating system manages a group of different computers and makes appear to be a single computer.
* **Network operating system:** This type of operating system running on a server and provides the capability to manage data, users, groups, security, applications, and other networking functions.
* **Real-time operating system:**This type of operating system serves real time system and the time interval required to process and respond to inputs is very small.

**SOFTWARE REQUIREMENT**

###### Wing personal – version 21H2

Release: - 2-4-2021

###### Language: - C programming language or 64-bit

###### **C Programming software**

Before you start writing in C, you will need some C programming software. Let's see exactly what you need!

## What C programming software I need?

Before you can write a program, you need an **editor** and a **compiler**.

### Code editor

The editor is a program where you write the source code of your program. You want to use a smart editor that helps with coding. Such editors will use different colours for keywords, values and variables. It is also desired that it will help with code suggestions.

Note**:** The editor is not a compiler!

## Compiler

A compiler is a [software](https://www.toppr.com/guides/computer-aptitude-and-knowledge/basics-of-computers/hardware-and-software/) that converts the source code to the object code. In other words, we can say that it converts the [high-level language](https://www.toppr.com/guides/computer-aptitude-and-knowledge/basics-of-computers/computer-languages/) to machine/binary language. Moreover, it is necessary to perform this step to make the program executable. This is because the computer understands only binary language.

Some compilers convert the high-level language to an assembly language as an intermediate step. Whereas some others convert it directly to machine code. This process of converting the source code into machine code is called **compilation.**Let us learn more about it in detail.

### Analysis of a Source Program

We can analyze a source code in three main steps. Moreover, these steps are further divided into different phases. The three steps are:

#### 1. Linear Analysis

Here, it reads the character of the code from left to right. The characters having a collective meaning are formed. We call these groups tokens.

#### 2. Hierarchical Analysis

According to collective meaning, we divide the tokens hierarchically in a nested manner.

#### 3. Semantic Analysis

In this step, we check if the components of the source code are appropriate in meaning.

### Phases/Structure of Compiler

The compilation process takes place in several phases. Moreover, for each step, the output of one step acts as the input for the next step. The phases/structure of the compilation process are is follows:

#### 1. Lexical Analyzer

* It takes the high-level language source code as the input.
* It scans the characters of source code from left to right. Hence, the name scanner also.
* It groups the characters into lexemes. Lexemes are a group of characters which has some meaning.
* Each lexeme corresponds to form a token.
* It removes white spaces and comments.
* It checks and removes the lexical errors.

#### 2. Syntax Analyzer

* ‘Parser’ is the other name for the syntax analyzer.
* The output of the lexical analyzer is its input.
* It checks for syntax errors in the source code.
* It does this by constructing a parse tree of all the tokens.
* For the syntax to be correct, the parse tree should be according to the rules of source code grammar.
* The grammar for such codes is context-free grammar.

#### 3. Semantic Analyzer

* It verifies the parse tree of the syntax analyzer.
* It checks the validity of the code in terms of programming language. Like, compatibility of data types, declaration, and initialization of variables, etc.
* It also produces a verified parse tree. Furthermore, we also call this tree an annotated parse tree.
* It also performs flow checking, type checking, etc.

#### 4. Intermediate Code Generator (ICG)

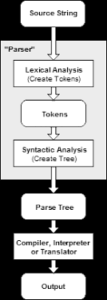
* It generates an intermediate code.
* This code is neither in high-level language nor in machine language. It is in an intermediate form.
* It is converted to machine language but, the last two phases are platform dependent.
* The intermediate code is the same for all the compilers. Further, we generate the machine code according to the platform.
* An example of an intermediate code is three address code.

#### 5. Code Optimizer

* It optimizes the intermediate code.
* Its function is to convert the code so that it executes faster using fewer resources (CPU, memory).
* It removes any useless lines of code and rearranges the code.
* The meaning of the source code remains the same.

#### 6. Target Code Generator

* Finally, it converts the optimized intermediate code into the machine code.
* This is the final stage of the compilation.
* The machine code which is produced is relocatable.



### Types of Compiler

#### 1. Cross Compilers

They produce an executable machine code for a platform but, this platform is not the one on which the compiler is running.

#### 2. Bootstrap Compilers

These compilers are written in a programming language that they have to compile.

#### 3. Source to source/transcompiler

These compilers convert the source code of one programming language to the source code of another programming language.

#### 4. Decompiler

Basically, it is not a compiler. It is just the reverse of the compiler. It converts the machine code into high-level language.

## Features of a Compiler

The features are as follows:

* Compilation speed.
* The correctness of machine code.
* The meaning of code should not change.
* Speed of machine code.
* Good error detection.
* Checking the code correctly according to grammar.

## Uses/Application of Compilers

* Helps to make the code independent of the platform.
* Makes the code free of syntax and semantic errors.
* Generate executable files of code.
* Translates the code from one language to another.

## Difference Between Compiler and Interpreter

A compiler checks the whole program at once. It displays all the errors at a place once the whole program is checked. On the other hand, an interpreter checks the program line by line. If an error is detected the execution stops.

### IDE

There are programs that contain in one place both a good editor to write your code and the compiler to translate that code to machine language. These more complex programs are called Integrated Development Environment (IDE). Most of them also provide a convenient way to debug your programs.

If you are just starting with programming – start with an IDE. Right now, you need to focus on the language and programming principles. Later you can transition to another environment, if you need to.

### **How does IDE make working so easy?**

IDEs are a migration form of primitive text editors, which uses full functionality technologies that quickly and efficiently edit the code. An IDE typically contains a graphical user interface (GUI) to access the code editor, a compiler or interpreter and a debugger. An IDE starts with a model that translates into a suitable code.

An integrated development environment (IDE) makes working easy by providing facilities such as a source code editor, build automation tools and a debugger to [programmers for software development](https://www.educba.com/what-is-software-development/). IDE makes it easy to see a visual representation of the files and makes it more understandable for the user.

Depending on the kind of program, IDE can build and deploy the whole process easier. Using a good IDE makes compiling and debugging easier. You can compile and run the program by clicking the Run button and debug the program by clicking the debug button.

IDE’s have some common features as listed below:

* Text editor: It provides a text editor to write and manage source code.
* Debugger: It uses debugging tools to identify the errors in the source code.
* Compiler
* Code Completion
* Programming language support
* Integration and use of plug-ins

**PROJECT CODE**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<windows.h>**

**#include<conio.h>**

**int i,j;**

**int main\_exit;**

**void menu();**

**struct date{**

**int month,day,year;**

**};**

**struct {**

**char name[60];**

**int acc\_no,age;**

**char address[60];**

**char citizenship[15];**

**double phone;**

**char acc\_type[10];**

**float amt;**

**struct date dob;**

**struct date deposit;**

**struct date withdraw;**

**}add,upd,check,rem,transaction;**

**float interest(float t,float amount,int rate)**

**{**

**float SI;**

**SI=(rate\*t\*amount)/100.0;**

**return (SI);**

**}**

**void fordelay(int j)**

**{ int i,k;**

**for(i=0;i<j;i++)**

**k=i;**

**}**

**void new\_acc()**

**{**

**int choice;**

**FILE \*ptr;**

**ptr=fopen("record.dat","a+");**

**account\_no:**

**system("cls");**

**printf("\t\t\t\xB2\xB2\xB2\ ADD RECORD \xB2\xB2\xB2\xB2");**

**printf("\n\n\nEnter today's date(mm/dd/yyyy):");**

**scanf("%d/%d/%d",&add.deposit.month,&add.deposit.day,&add.deposit.year);**

**printf("\nEnter the account number:");**

**scanf("%d",&check.acc\_no);**

**while(fscanf(ptr,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",&add.acc\_no,add.name,&add.dob.month,&add.dob.day,&add.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc\_type,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)**

**{**

**if (check.acc\_no==add.acc\_no)**

**{printf("Account no. already in use!");**

**fordelay(1000000000);**

**goto account\_no;**

**}**

**}**

**add.acc\_no=check.acc\_no;**

**printf("\nEnter the name:");**

**scanf("%s",add.name);**

**printf("\nEnter the date of birth(mm/dd/yyyy):");**

**scanf("%d/%d/%d",&add.dob.month,&add.dob.day,&add.dob.year);**

**printf("\nEnter the age:");**

**scanf("%d",&add.age);**

**printf("\nEnter the address:");**

**scanf("%s",add.address);**

**printf("\nEnter the citizenship number:");**

**scanf("%s",add.citizenship);**

**printf("\nEnter the phone number: ");**

**scanf("%lf",&add.phone);**

**printf("\nEnter the amount to deposit:Rs.");**

**scanf("%f",&add.amt);**

**printf("\nType of account:\n\t#Saving\n\t#Current\n\t#Fixed1(for 1 year)\n\t#Fixed2(for 2 years)\n\t#Fixed3(for 3 years)\n\n\tEnter your choice:");**

**scanf("%s",add.acc\_type);**

**fprintf(ptr,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",add.acc\_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,add.acc\_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);**

**fclose(ptr);**

**printf("\nAccount created successfully!");**

**add\_invalid:**

**printf("\n\n\n\t\tEnter 1 to go to the main menu and 0 to exit:");**

**scanf("%d",&main\_exit);**

**system("cls");**

**if (main\_exit==1)**

**menu();**

**else if(main\_exit==0)**

**void close();**

**else**

**{**

**printf("\nInvalid!\a");**

**goto add\_invalid;**

**}**

**}**

**void view\_list()**

**{**

**FILE \*view;**

**view=fopen("record.dat","r");**

**int test=0;**

**system("cls");**

**printf("\nACC. NO.\tNAME\t\t\tADDRESS\t\t\tPHONE\n");**

**while(fscanf(view,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d",&add.acc\_no,add.name,&add.dob.month,&add.dob.day,&add.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc\_type,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)**

**{**

**printf("\n%6d\t %10s\t\t\t%10s\t\t%.0lf",add.acc\_no,add.name,add.address,add.phone);**

**test++;**

**}**

**fclose(view);**

**if (test==0)**

**{ system("cls");**

**printf("\nNO RECORDS!!\n");}**

**view\_list\_invalid:**

**printf("\n\nEnter 1 to go to the main menu and 0 to exit:");**

**scanf("%d",&main\_exit);**

**system("cls");**

**if (main\_exit==1)**

**menu();**

**else if(main\_exit==0)**

**void close();**

**else**

**{**

**printf("\nInvalid!\a");**

**goto view\_list\_invalid;**

**}**

**}**

**void edit(void)**

**{**

**int choice,test=0;**

**FILE \*old,\*newrec;**

**old=fopen("record.dat","r");**

**newrec=fopen("new.dat","w");**

**printf("\nEnter the account no. of the customer whose info you want to change:");**

**scanf("%d",&upd.acc\_no);**

**while(fscanf(old,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d",&add.acc\_no,add.name,&add.dob.month,&add.dob.day,&add.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc\_type,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)**

**{**

**if (add.acc\_no==upd.acc\_no)**

**{ test=1;**

**printf("\nWhich information do you want to change?\n1.Address\n2.Phone\n\nEnter your choice(1 for address and 2 for phone):");**

**scanf("%d",&choice);**

**system("cls");**

**if(choice==1)**

**{printf("Enter the new address:");**

**scanf("%s",upd.address);**

**fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",add.acc\_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,upd.address,add.citizenship,add.phone,add.acc\_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);**

**system("cls");**

**printf("Changes saved!");**

**}**

**else if(choice==2)**

**{**

**printf("Enter the new phone number:");**

**scanf("%lf",&upd.phone);**

**fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",add.acc\_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,upd.phone,add.acc\_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);**

**system("cls");**

**printf("Changes saved!");**

**}**

**}**

**else**

**fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",add.acc\_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,add.acc\_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);**

**}**

**fclose(old);**

**fclose(newrec);**

**remove("record.dat");**

**rename("new.dat","record.dat");**

**if(test!=1)**

**{ system("cls");**

**printf("\nRecord not found!!\a\a\a");**

**edit\_invalid:**

**printf("\nEnter 0 to try again,1 to return to main menu and 2 to exit:");**

**scanf("%d",&main\_exit);**

**system("cls");**

**if (main\_exit==1)**

**menu();**

**else if (main\_exit==2)**

**void close();**

**else if(main\_exit==0)**

**edit();**

**else**

**{printf("\nInvalid!\a");**

**goto edit\_invalid;}**

**}**

**else**

**{printf("\n\n\nEnter 1 to go to the main menu and 0 to exit:");**

**scanf("%d",&main\_exit);**

**system("cls");**

**if (main\_exit==1)**

**menu();**

**else**

**void close();**

**}**

**}**

**void transact(void)**

**{ int choice,test=0;**

**FILE \*old,\*newrec;**

**old=fopen("record.dat","r");**

**newrec=fopen("new.dat","w");**

**printf("Enter the account no. of the customer:");**

**scanf("%d",&transaction.acc\_no);**

**while (fscanf(old,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d",&add.acc\_no,add.name,&add.dob.month,&add.dob.day,&add.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc\_type,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)**

**{**

**if(add.acc\_no==transaction.acc\_no)**

**{ test=1;**

**if(strcmpi(add.acc\_type,"fixed1")==0||strcmpi(add.acc\_type,"fixed2")==0||strcmpi(add.acc\_type,"fixed3")==0)**

**{**

**printf("\a\a\a\n\nYOU CANNOT DEPOSIT OR WITHDRAW CASH IN FIXED ACCOUNTS!!!!!");**

**fordelay(1000000000);**

**system("cls");**

**menu();**

**}**

**printf("\n\nDo you want to\n1.Deposit\n2.Withdraw?\n\nEnter your choice(1 for deposit and 2 for withdraw):");**

**scanf("%d",&choice);**

**if (choice==1)**

**{**

**printf("Enter the amount you want to deposit:Rs.");**

**scanf("%f",&transaction.amt);**

**add.amt+=transaction.amt;**

**fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",add.acc\_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,add.acc\_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);**

**printf("\n\nDeposited successfully!");**

**}**

**else**

**{**

**printf("Enter the amount you want to withdraw:Rs. ");**

**scanf("%f",&transaction.amt);**

**if(transaction.amt>add.amt)**

**{**

**printf("insufficient ammount");**

**getch();**

**return menu();**

**}**

**add.amt-=transaction.amt;**

**fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",add.acc\_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,add.acc\_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);**

**printf("\n\nWithdrawn successfully!");**

**}**

**}**

**else**

**{**

**fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",add.acc\_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,add.acc\_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);**

**}**

**}**

**fclose(old);**

**fclose(newrec);**

**remove("record.dat");**

**rename("new.dat","record.dat");**

**if(test!=1)**

**{**

**printf("\n\nRecord not found!!");**

**transact\_invalid:**

**printf("\n\n\nEnter 0 to try again,1 to return to main menu and 2 to exit:");**

**scanf("%d",&main\_exit);**

**system("cls");**

**if (main\_exit==0)**

**transact();**

**else if (main\_exit==1)**

**menu();**

**else if (main\_exit==2)**

**void close();**

**else**

**{**

**printf("\nInvalid!");**

**goto transact\_invalid;**

**}**

**}**

**else**

**{**

**printf("\nEnter 1 to go to the main menu and 0 to exit:");**

**scanf("%d",&main\_exit);**

**system("cls");**

**if (main\_exit==1)**

**menu();**

**else**

**void close();**

**}**

**}**

**void erase(void)**

**{**

**FILE \*old,\*newrec;**

**int test=0;**

**old=fopen("record.dat","r");**

**newrec=fopen("new.dat","w");**

**printf("Enter the account no. of the customer you want to delete:");**

**scanf("%d",&rem.acc\_no);**

**while (fscanf(old,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d",&add.acc\_no,add.name,&add.dob.month,&add.dob.day,&add.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc\_type,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)**

**{**

**if(add.acc\_no!=rem.acc\_no)**

**fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d\n",add.acc\_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,add.acc\_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);**

**else**

**{test++;**

**printf("\nRecord deleted successfully!\n");**

**}**

**}**

**fclose(old);**

**fclose(newrec);**

**remove("record.dat");**

**rename("new.dat","record.dat");**

**if(test==0)**

**{**

**printf("\nRecord not found!!\a\a\a");**

**erase\_invalid:**

**printf("\nEnter 0 to try again,1 to return to main menu and 2 to exit:");**

**scanf("%d",&main\_exit);**

**if (main\_exit==1)**

**menu();**

**else if (main\_exit==2)**

**void close();**

**else if(main\_exit==0)**

**erase();**

**else**

**{printf("\nInvalid!\a");**

**goto erase\_invalid;}**

**}**

**else**

**{printf("\nEnter 1 to go to the main menu and 0 to exit:");**

**scanf("%d",&main\_exit);**

**system("cls");**

**if (main\_exit==1)**

**menu();**

**else**

**void close();**

**}**

**}**

**void see(void)**

**{**

**FILE \*ptr;**

**int test=0,rate;**

**int choice;**

**float time;**

**float intrst;**

**ptr=fopen("record.dat","r");**

**printf("Do you want to check by\n1.Account no\n2.Name\nEnter your choice:");**

**scanf("%d",&choice);**

**if (choice==1)**

**{ printf("Enter the account number:");**

**scanf("%d",&check.acc\_no);**

**while (fscanf(ptr,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d",&add.acc\_no,add.name,&add.dob.month,&add.dob.day,&add.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc\_type,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)**

**{**

**if(add.acc\_no==check.acc\_no)**

**{ system("cls");**

**test=1;**

**printf("\nAccount NO.:%d\nName:%s \nDOB:%d/%d/%d \nAge:%d \nAddress:%s \nCitizenship No:%s \nPhone number:%.0lf \nType Of Account:%s \nAmount deposited:Rs. %.2f \nDate Of Deposit:%d/%d/%d\n\n",add.acc\_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,**

**add.acc\_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);**

**if(strcmpi(add.acc\_type,"fixed1")==0)**

**{**

**time=1.0;**

**rate=9;**

**intrst=interest(time,add.amt,rate);**

**printf("\n\nYou will get Rs.%.2f as interest on %d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+1);**

**}**

**else if(strcmpi(add.acc\_type,"fixed2")==0)**

**{**

**time=2.0;**

**rate=11;**

**intrst=interest(time,add.amt,rate);**

**printf("\n\nYou will get Rs.%.2f as interest on %d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+2);**

**}**

**else if(strcmpi(add.acc\_type,"fixed3")==0)**

**{**

**time=3.0;**

**rate=13;**

**intrst=interest(time,add.amt,rate);**

**printf("\n\nYou will get Rs.%.2f as interest on %d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+3);**

**}**

**else if(strcmpi(add.acc\_type,"saving")==0)**

**{**

**time=(1.0/12.0);**

**rate=8;**

**intrst=interest(time,add.amt,rate);**

**printf("\n\nYou will get Rs..%.2f as interest on %d of every month",intrst,add.deposit.day);**

**}**

**else if(strcmpi(add.acc\_type,"current")==0)**

**{**

**printf("\n\nYou will get no interest\a\a");**

**}**

**}**

**}**

**}**

**else if (choice==2)**

**{ printf("Enter the name:");**

**scanf("%s",&check.name);**

**while (fscanf(ptr,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d",&add.acc\_no,add.name,&add.dob.month,&add.dob.day,&add.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc\_type,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)**

**{**

**if(strcmpi(add.name,check.name)==0)**

**{ system("cls");**

**test=1;**

**printf("\nAccount No.:%d\nName:%s \nDOB:%d/%d/%d \nAge:%d \nAddress:%s \nCitizenship No:%s \nPhone number:%.0lf \nType Of Account:%s \nAmount deposited:Rs.%.2f \nDate Of Deposit:%d/%d/%d\n\n",add.acc\_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,**

**add.acc\_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);**

**if(strcmpi(add.acc\_type,"fixed1")==0)**

**{**

**time=1.0;**

**rate=9;**

**intrst=interest(time,add.amt,rate);**

**printf("\n\nYou will get Rs.%.2f as interest on %d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+1);**

**}**

**else if(strcmpi(add.acc\_type,"fixed2")==0)**

**{**

**time=2.0;**

**rate=11;**

**intrst=interest(time,add.amt,rate);**

**printf("\n\nYou will get Rs.%.2f as interest on %d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+2);**

**}**

**else if(strcmpi(add.acc\_type,"fixed3")==0)**

**{**

**time=3.0;**

**rate=13;**

**intrst=interest(time,add.amt,rate);**

**printf("\n\nYou will get Rs.%.2f as interest on %d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+3);**

**}**

**else if(strcmpi(add.acc\_type,"saving")==0)**

**{**

**time=(1.0/12.0);**

**rate=8;**

**intrst=interest(time,add.amt,rate);**

**printf("\n\nYou will get Rs.%.2f as interest on %d of every month",intrst,add.deposit.day);**

**}**

**else if(strcmpi(add.acc\_type,"current")==0)**

**{**

**printf("\n\nYou will get no interest\a\a");**

**}**

**}**

**}**

**}**

**fclose(ptr);**

**if(test!=1)**

**{ system("cls");**

**printf("\nRecord not found!!\a\a\a");**

**see\_invalid:**

**printf("\nEnter 0 to try again,1 to return to main menu and 2 to exit:");**

**scanf("%d",&main\_exit);**

**system("cls");**

**if (main\_exit==1)**

**menu();**

**else if (main\_exit==2)**

**void close();**

**else if(main\_exit==0)**

**see();**

**else**

**{**

**system("cls");**

**printf("\nInvalid!\a");**

**goto see\_invalid;}**

**}**

**else**

**{printf("\nEnter 1 to go to the main menu and 0 to exit:");**

**scanf("%d",&main\_exit);}**

**if (main\_exit==1)**

**{**

**system("cls");**

**menu();**

**}**

**else**

**{**

**system("cls");**

**void close();**

**}**

**}**

**void close(void)**

**{**

**printf("\n\n\n\nThis C Mini Project is developed by divya dubey!");**

**}**

**void menu(void)**

**{ int choice;**

**system("cls");**

**system("color 8");**

**printf("\n\n\t\t\tCUSTOMER ACCOUNT BANKING MANAGEMENT SYSTEM");**

**printf("\n\n\n\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2 WELCOME TO THE MAIN MENU \xB2\xB2\xB2\xB2\xB2\xB2\xB2");**

**printf("\n\n\t\t1.Create new account\n\t\t2.Update information of existing account\n\t\t3.For transactions\n\t\t4.Check the details of existing account\n\t\t5.Removing existing account\n\t\t6.View customer's list\n\t\t7.Exit\n\n\n\n\n\t\t Enter your choice:");**

**scanf("%d",&choice);**

**system("cls");**

**switch(choice)**

**{**

**case 1:new\_acc();**

**break;**

**case 2:edit();**

**break;**

**case 3:transact();**

**break;**

**case 4:see();**

**break;**

**case 5:erase();**

**break;**

**case 6:view\_list();**

**break;**

**case 7:close();**

**break;**

**}**

**}**

**int main()**

**{**

**char pass[20],password[20]="1234";**

**int i=0;**

**printf("\n\n\t\tEnter the password to login:");**

**scanf("%s",pass);**

**/\*do**

**{**

**//if (pass[i]!=13&&pass[i]!=8)**

**{**

**printf("\*");**

**pass[i]=getch();**

**i++;**

**}**

**}while (pass[i]!=13);**

**pass[10]='\0';\*/**

**if (strcmp(pass,password)==0)**

**{printf("\n\nPassword Match!\nLOADING");**

**for(i=0;i<=6;i++)**

**{**

**fordelay(100000000);**

**printf(".");**

**}**

**system("cls");**

**menu();**

**}**

**else**

**{ printf("\n\nWrong password!!\a\a\a");**

**login\_try:**

**printf("\nEnter 1 to try again and 0 to exit:");**

**scanf("%d",&main\_exit);**

**if (main\_exit==1)**

**{**

**system("cls");**

**main();**

**}**

**else if (main\_exit==0)**

**{**

**system("cls");**

**close();}**

**else**

**{printf("\nInvalid!");**

**fordelay(1000000000);**

**system("cls");**

**goto login\_try;}**

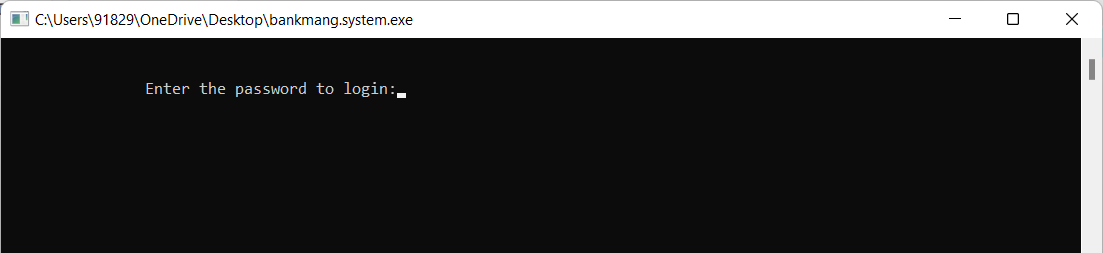
**}**

**return 0;**

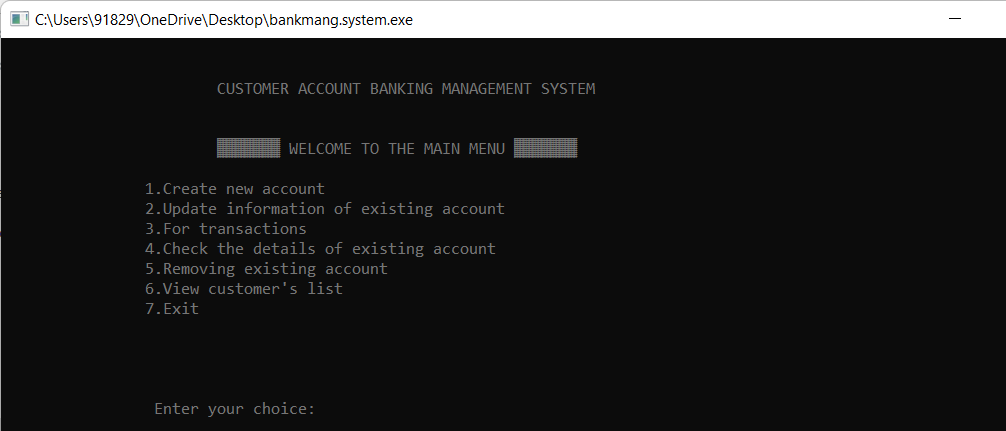
**}**

**OUTPUT SCREENSHOTS**

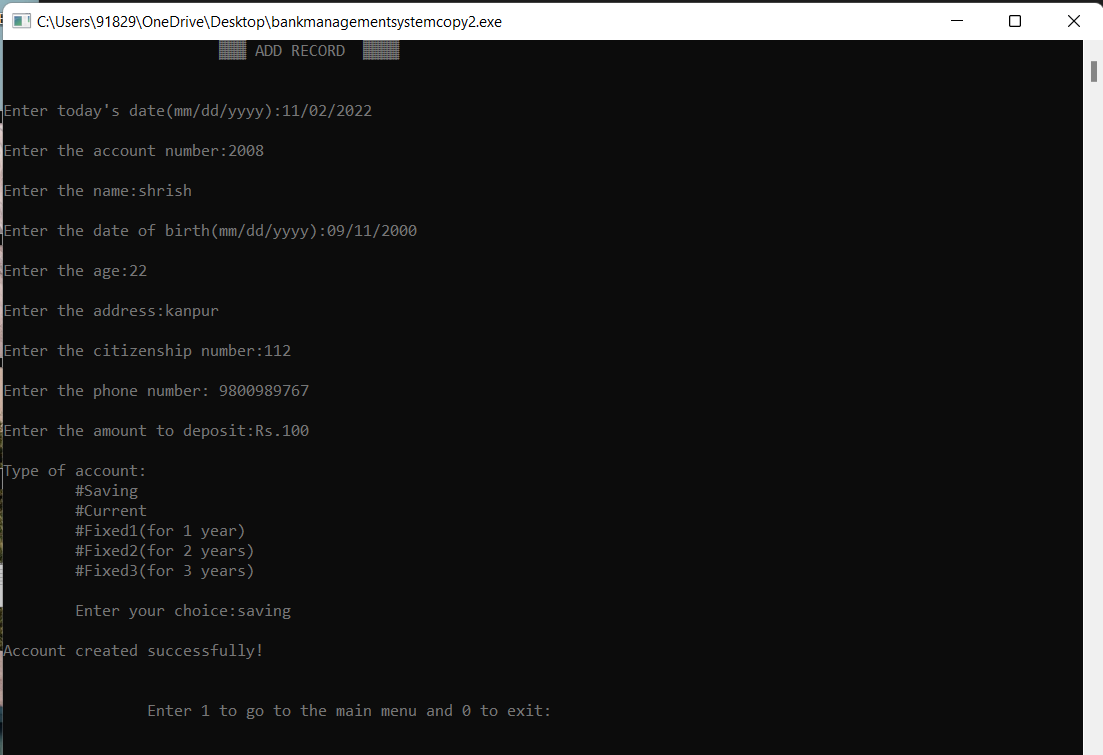
Snapshot 1:



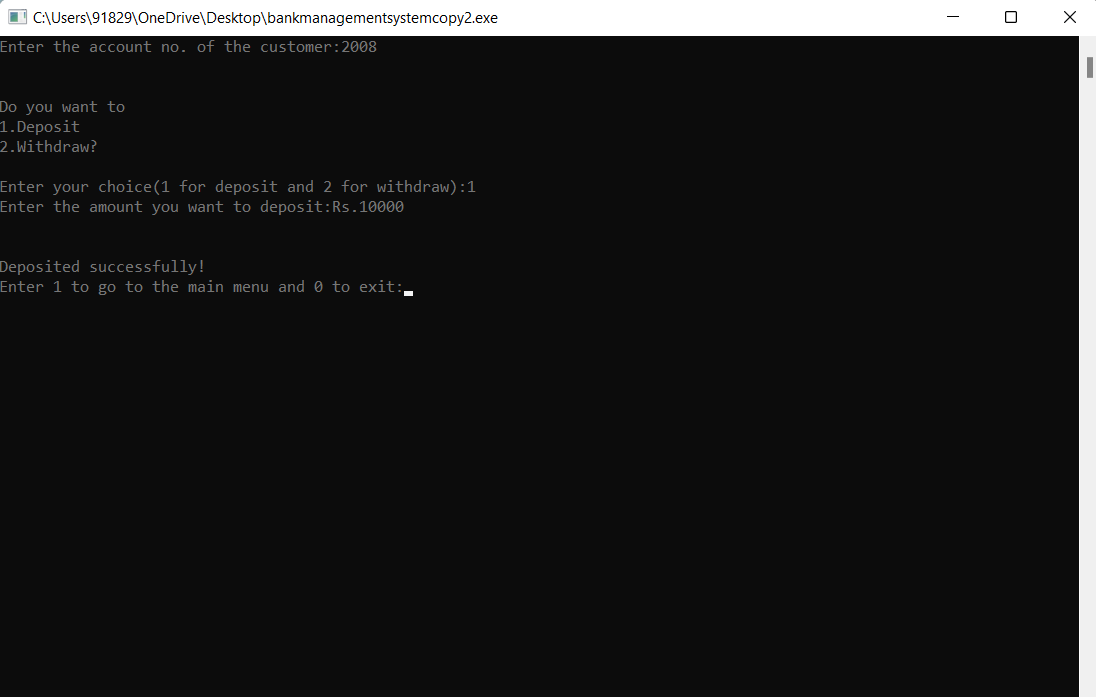
Snapshot 2:



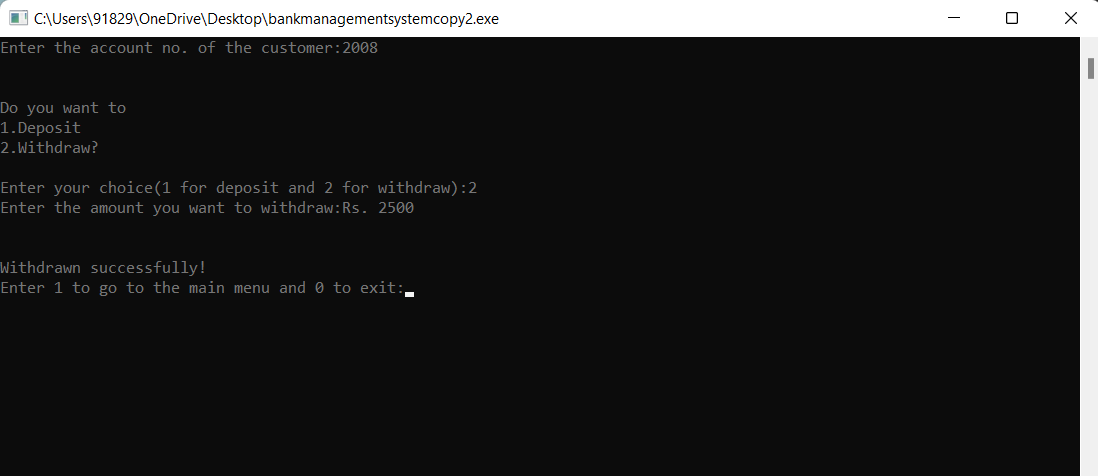
**Snapshot 3:**



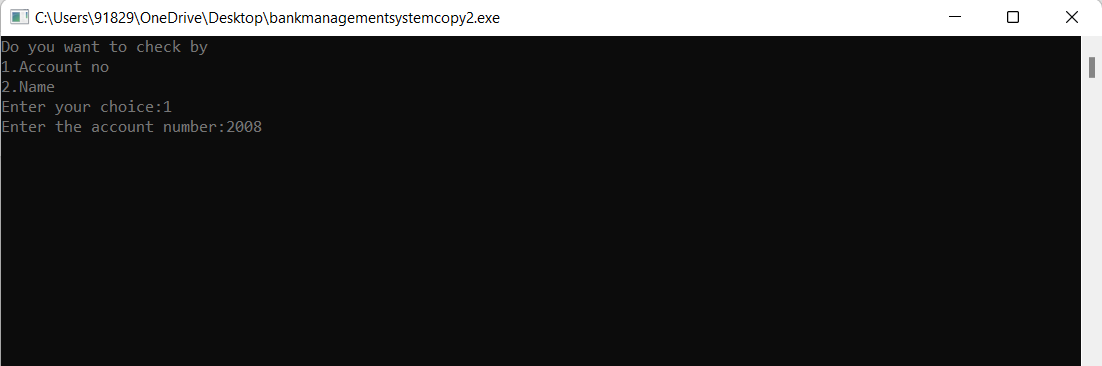
**Snapshot 4:**



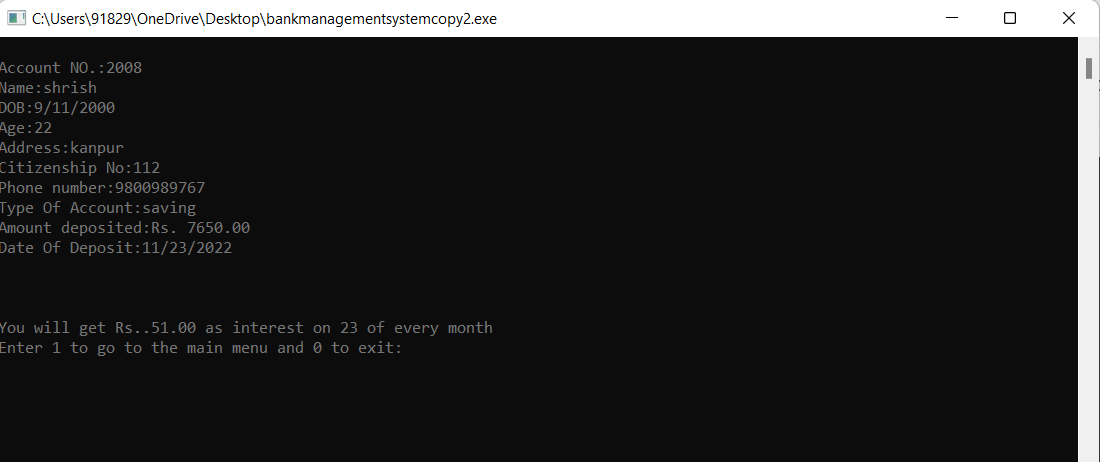
**Snapshot 5:;**



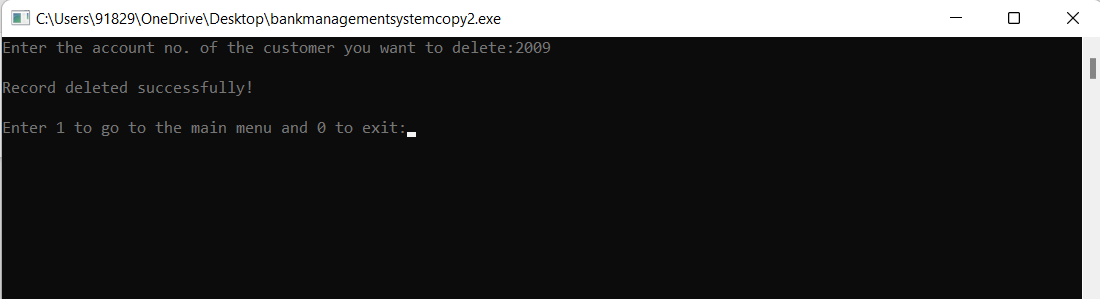
**Snapshot 6:**



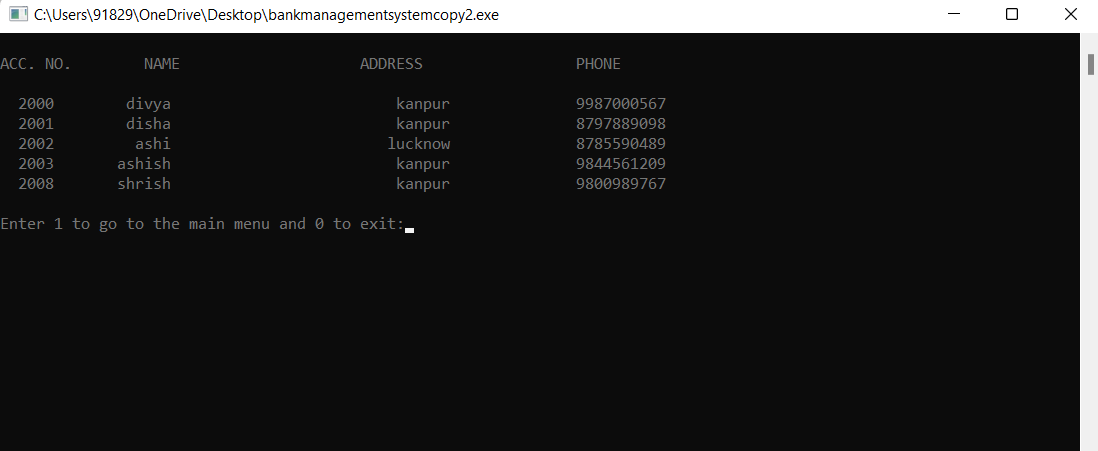
**Snapshot 7:**



**Snapshot 8:**



**Snapshot 9:**



# CONCLUSION

This system is named as bank management system. This system is made to create account , update the information of existing accounts, transaction process( deposit and withdraw),check the details of existing accounts and removing the existing account.

# REFERENCES

## Bibliography:

Book: let us c

[www.google.com](http://www.google.com/) [www.youtube.com](http://www.youtube.com/)