VCOMP :-
(Version Comparator/Difference Tool)

A Thesis
Submitted in partial fulfillment of the
requirements for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS

BY

TATHAGAT KUMAR

(MCA/10023/2020)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BIRLA INSTITUTE OF TECHNOLOGY
MESRA-835215, RANCHI
2022

DECLARATION

I certify that

a. The work contained in the thesis is original and has been done by myself under the general supervision of my supervisor(s).

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in writing the thesis.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

Signature of the Student

Name : Tathagat Kumar
Roll No : MCA/10023/2020

APPROVAL OF THE GUIDE(S)

Recommended that the thesis entitled **"VCOMP :- Version Comparator"** prepared By Mr. Tathagat Kumar under my/our supervision and guidance be accepted as fulfilling this part of the requirements for the degree of Master of Computer Applications. To the best of my/our knowledge, the contents of this Project did not form a basis for the award of any previous degree to anyone else.

Date:

Jyoti Mehta
Software Engineering Manager
Amdocs Development Centre India
Pune.

B.K.Chanda
Assistant Professor,
Department of CSE,
BIT-Mesra, Ranchi

THESIS APPROVAL CERTIFICATE

This is to certify that the work embodied in this thesis entitled ""**VCOMP :- Version Comparator**" is carried out by Mr. Tathagat Kumar (Roll No. MCA/10023/2020) is approved for the degree of Master of Computer Applications of Birla Institute of Technology, Mesra, Ranchi.

Date:

Place:

Internal Examiner(s)

Name & Signature

External Examiner(s)
Name & Signature

Head
Dept of Computer Science and Engineering
Birla Institute of technology, Mesra

ACKNOWLEDGEMENT

We take this opportunity to express our profound gratitude and deep regards to our mentor **Jyoti Mehta**, Software Engineering Manager, Amdocs Development Centre, Pune, India, for their exemplary guidance, monitoring and constant encouragement throughout the course of this project work. The blessing, help and guidance given by them time to time shall carry us a long way in the journey of life on which we are about to embark.

We are obliged to all the faculty members of MCA, Computer Science Department, Birla Institute of Technology Mesra, for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our assignment. I would like to take this opportunity to acknowledge and express my gratitude to all the people directly or indirectly involved with me in making this project a success.

Tathagat Kumar
Roll no. – MCA/10023/2020

ABSTRACT

‘ VCOMP : Version Comparator ’ is a tool or resource that generates a Report showing the changes in two different versions of same product, which will help the developers or user to know exactly what are the changes made in new file that will result in faster debugging of Code, Error detection and its rectification.

User will have to provide two library files using CMD, which will be Input to the tool and will generate a Text File (.txt/.rtf) or a HTML File(.htm /.html) containing detailed report of all the changes.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter-1

COMPANY PROFILE

Amdocs, founded in 1982, is a leading software and services provider to communications and media companies of all sizes, accelerating the industry's dynamic and continuous digital transformation. With a rich set of innovative solutions, long-term business relationships with 350 communications and media providers, and technology and distribution ties to 600 content creators.

Amdocs delivers business improvements to drive growth. Amdocs and its 25,000 employees serve customers in over 85 countries.

Amdocs' market offerings address five business imperatives:

- Consumer experience and monetization
- Media and digital services
- Enterprise and connected society
- Service-driven networks
- Services and agile operations

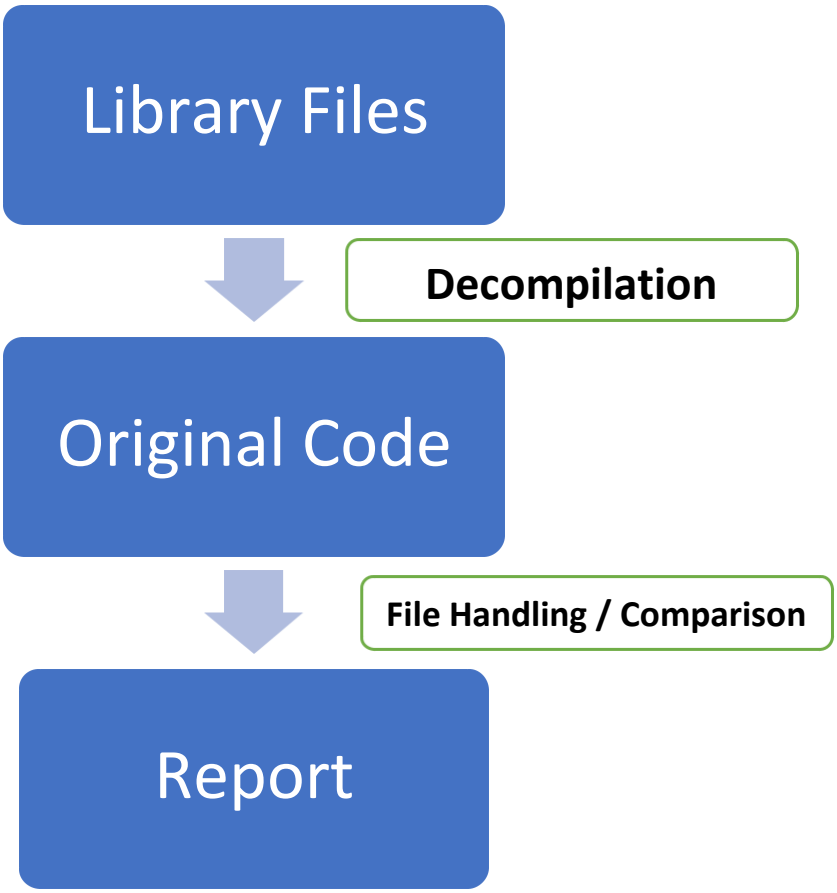
Global presence:

- Workforce of around 25,000 serving customers across six continents
- Support and development Centers located worldwide, including Brazil, Canada, Cyprus, India, Ireland, Israel, Mexico, the Philippines, the United Kingdom and the USA.

Market Position:

Amdocs ONE provides service providers with the broadest set of products and services to grow revenue and build loyalty. Uniquely designed to meet the challenge of our customers' hybrid environment, Amdocs ONE is available on an open, modular architecture, built on cloud-native microservice technologies for high-velocity time to market. Deployed using DevOps in small iterations to control costs, Amdocs ONE brings optimal scope and drives agility.

PROJECT OVERVIEW



TECHNICAL TERMINOLOGY :-

Library

The files that tell the compiler how to call some functionality (without knowing how the functionality actually works) are called header files. They contain the function prototypes. They also contain Data types and constants used with the libraries.

Whereas, Library is the place where the actual functionality is implemented i.e. they contain function body.

Library files are non-human-readable. Since they are in the form of machine code. Library files in our program are included in last stage by special software called as *linker*.

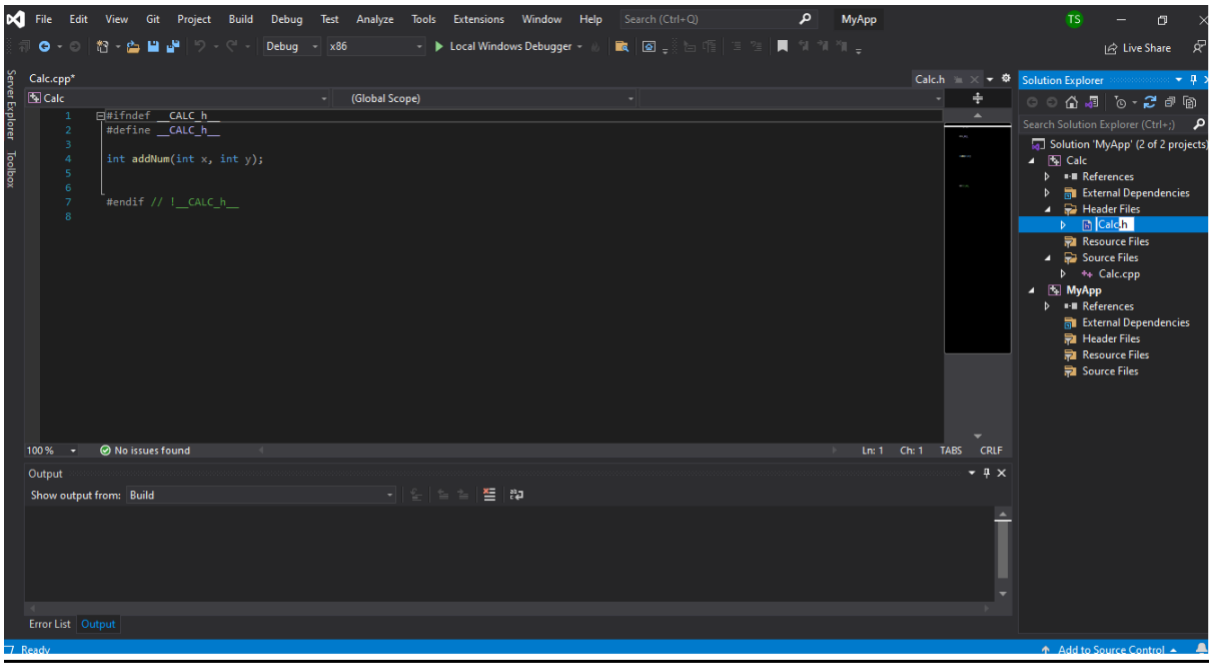
Libraries have mainly two categories :

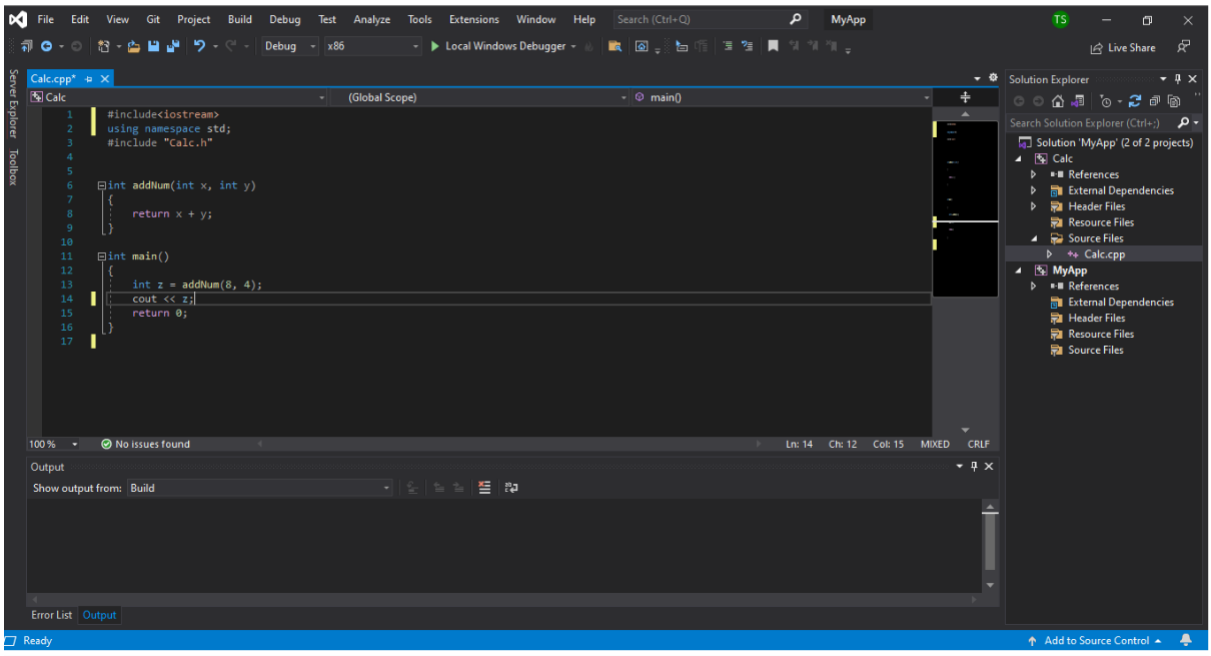
1. Static
2. Shared or Dynamic

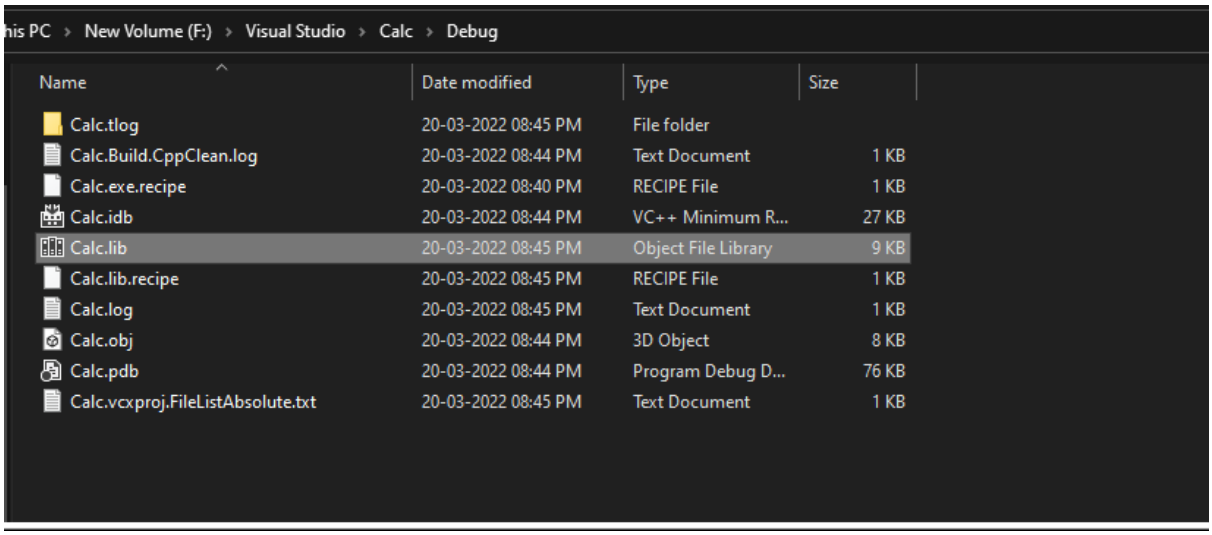
Static: Static libraries contains object code linked with an end user application and then they become the part of the executable. These libraries are specifically used at ***compile time*** which means the library should be present in correct location when user wants to compile his/her C or C++ program. In windows they end with .lib extension and with .a for MacOS.

Shared or Dynamic: These libraries are only required at ***run-time*** i.e, user can compile his/her code without using these libraries. For example, when we open our game folders we can find many .dll (dynamic link libraries) files. As these libraries can be shared by multiple programs, they are also called as shared libraries. These files end with .dll or .lib extensions. In windows they end with .dll extension.

1.Creation of Header File and .lib file:-

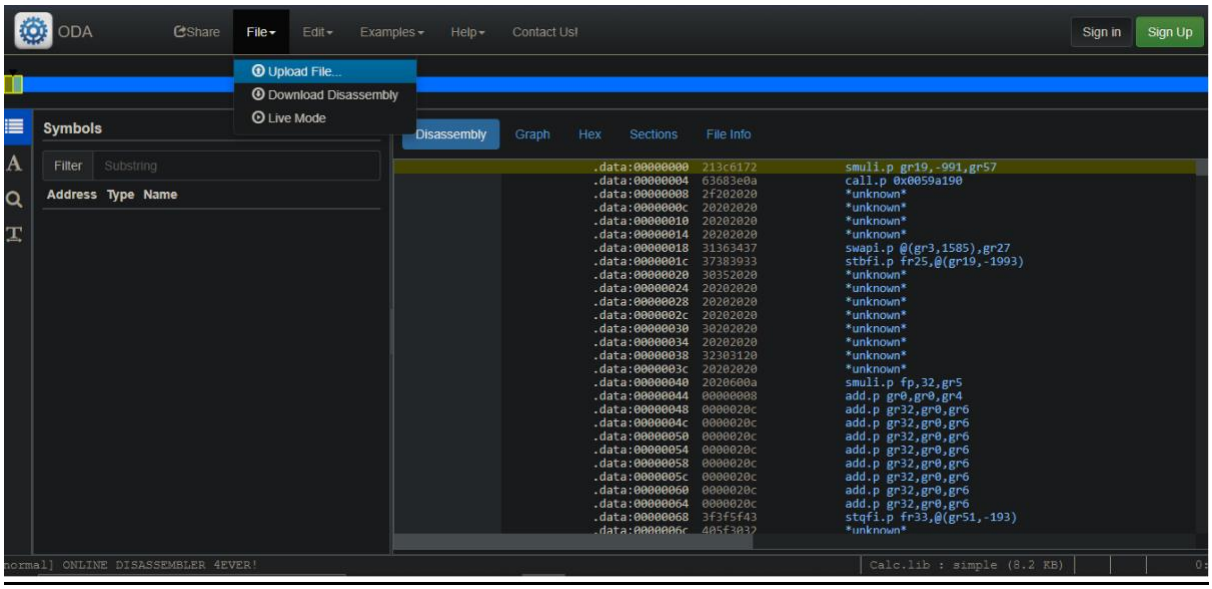


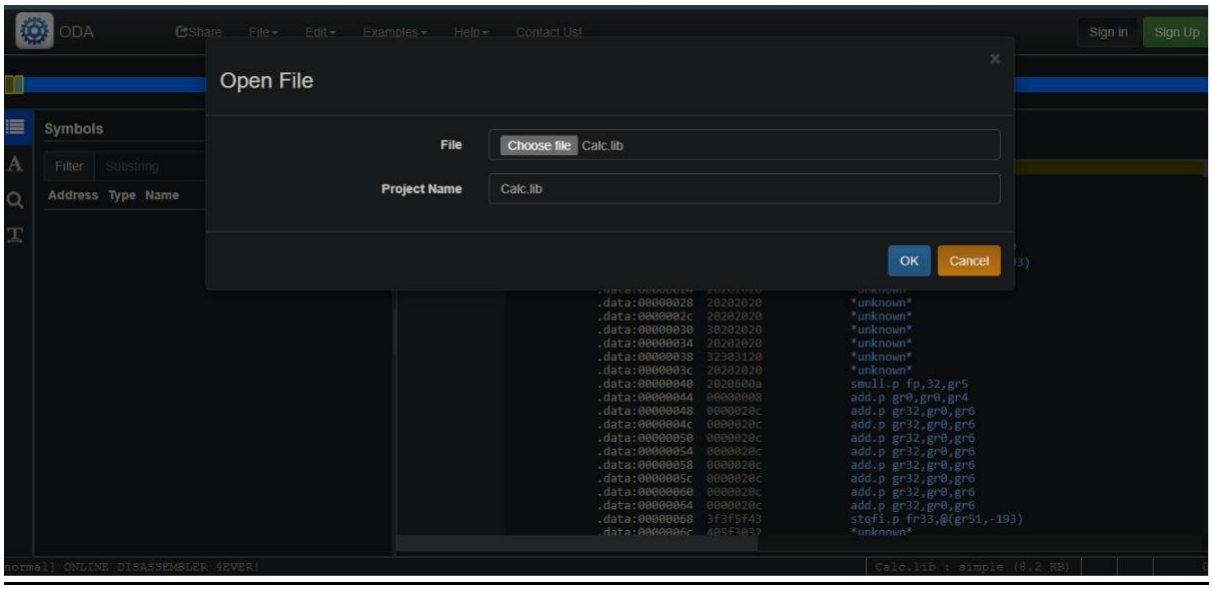




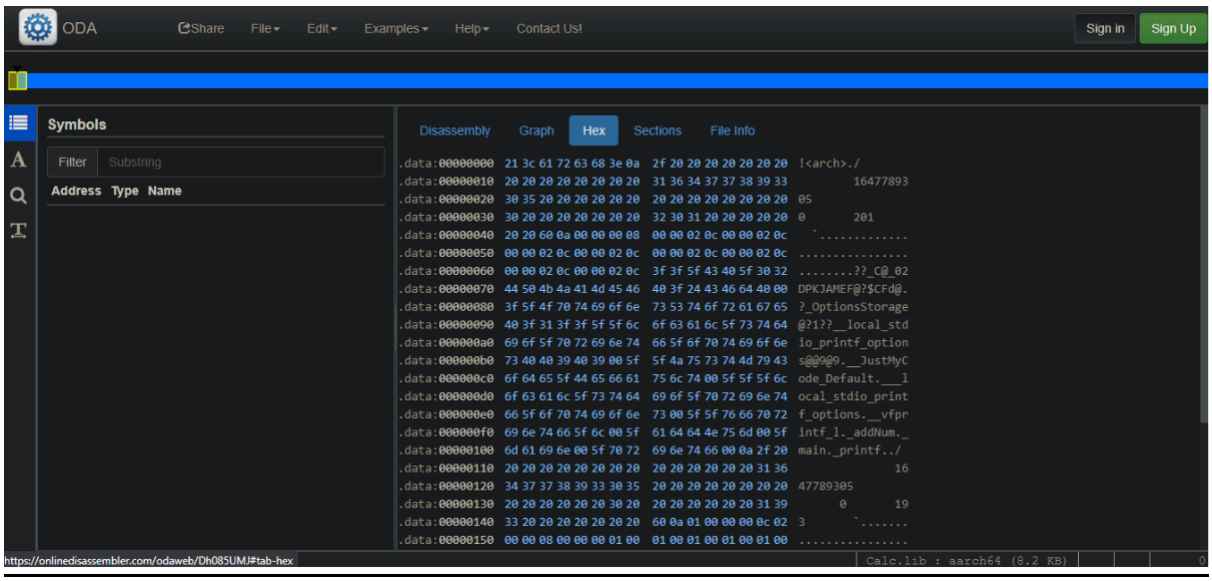
2.Using Online Decompiler (https://onlinedisassembler.com/):-

Uploading .lib files-



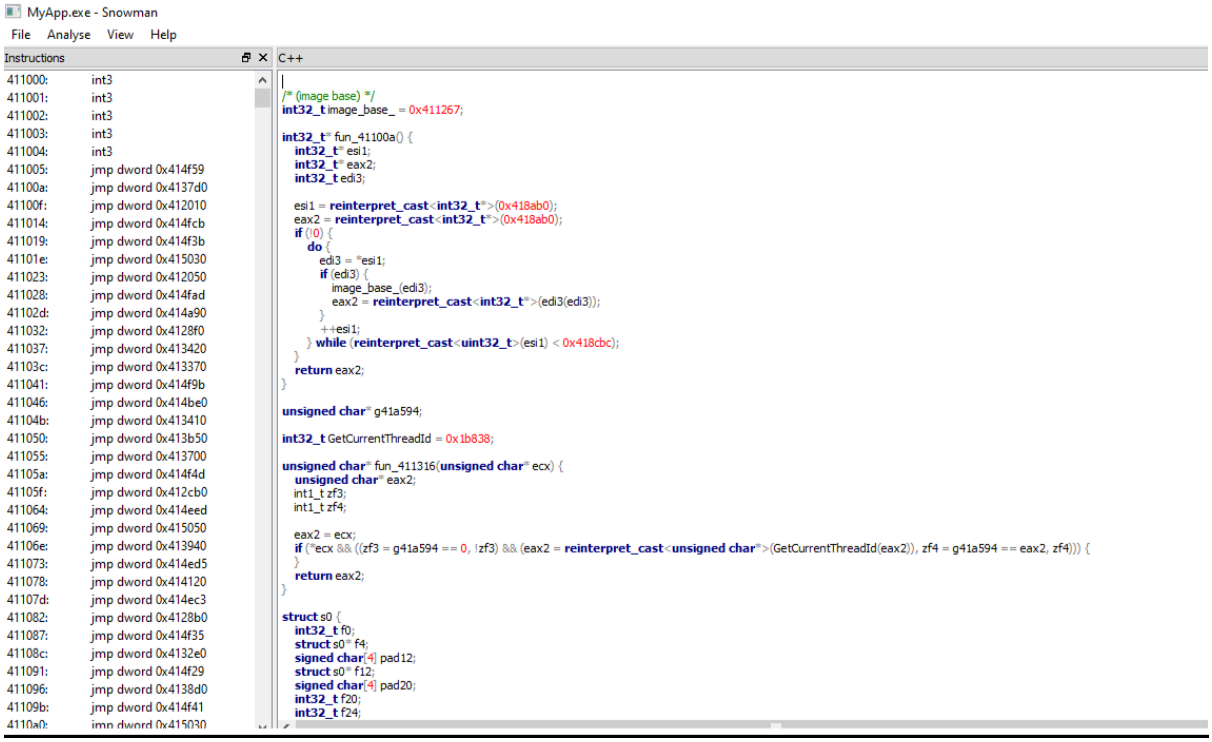

ODA
Share
File
Edit
Examples
Help
Contact Us!
Sign in
Sign Up
Open File
File
Choose file
Calc.lib
Project Name
Calc.lib
OK
Cancel
Symbols
Filter
Substring
Address
Type
Name

3.Output:

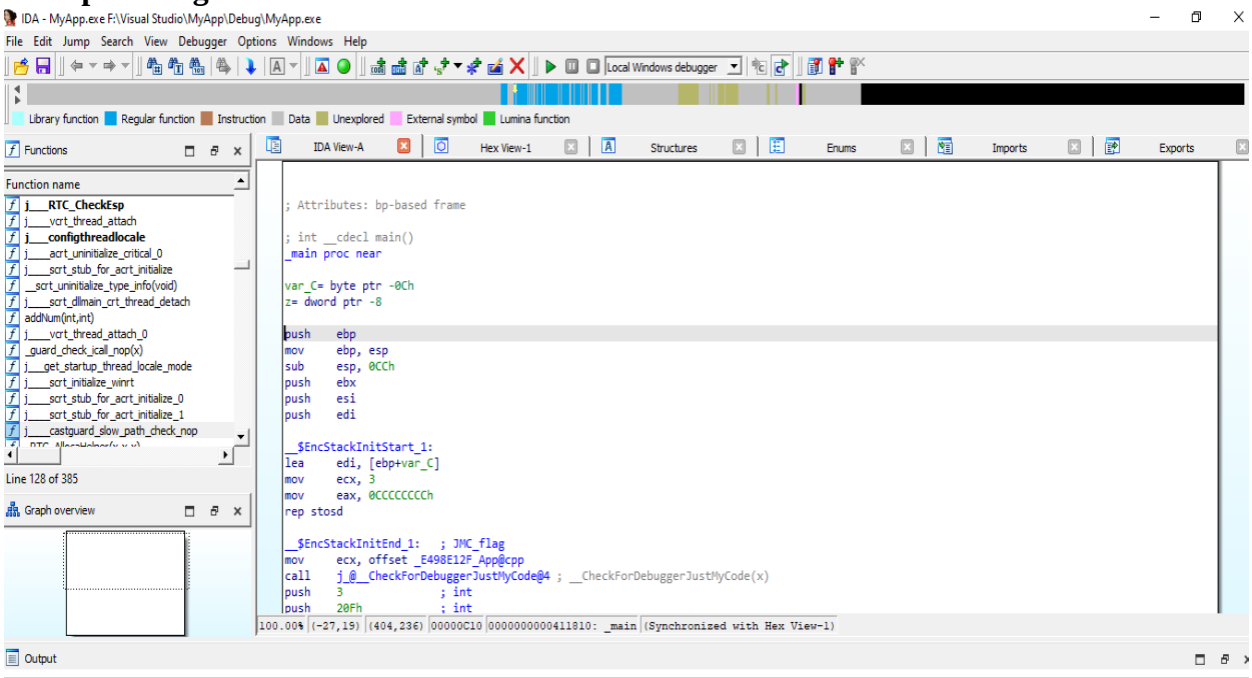

ODA
Share
File
Edit
Examples
Help
Contact Us!
Sign in
Sign Up
Symbols
Filter
Substring
Address
Type
Name
Disassembly
Graph
Hex
Sections
File Info

4.Output using another application (Snowman)

https://derevenets.com/



5.Output using IDA Freeware:



Chapter 2- Requirement Analysis
EXISTING SYSTEMS AND SOLUTIONS

Source Code Comparison using Online Comparator:-
(https://www.diffnow.com/compare-clips)

Calc1.cpp

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

class Math
{
    int add(int a, int b, int c,int d)
    {
        return a + b + c+ d;
    }

};

class Mathematics
{
    int add(int a, int b,int c)
    {
        return a + b + c;
    }

};
```

Calc2.cpp

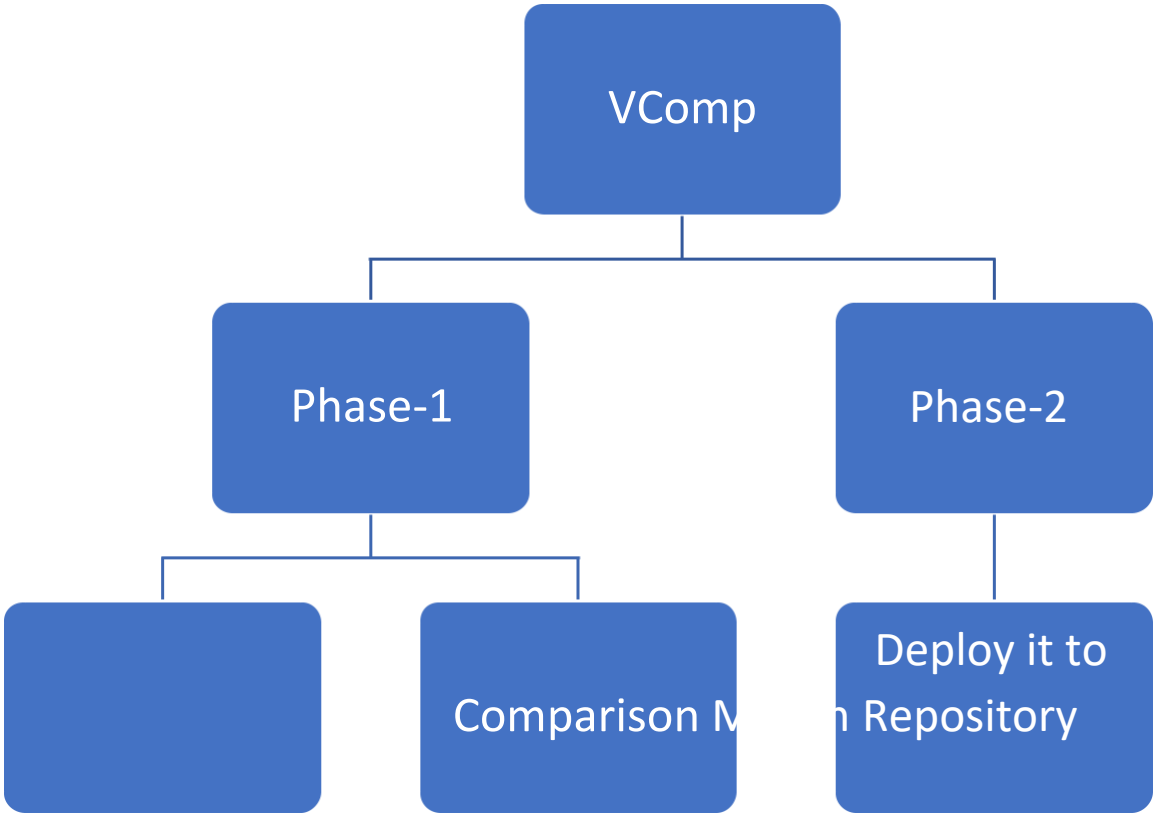
```
#include<iostream>
using namespace std;

class Math
{
    int add(int a, int b, int c)
    {
        return a + b + c;
    }

};

class Mathematics
{
    int add(int a, int b)
    {
        return a + b;
    }

};

```

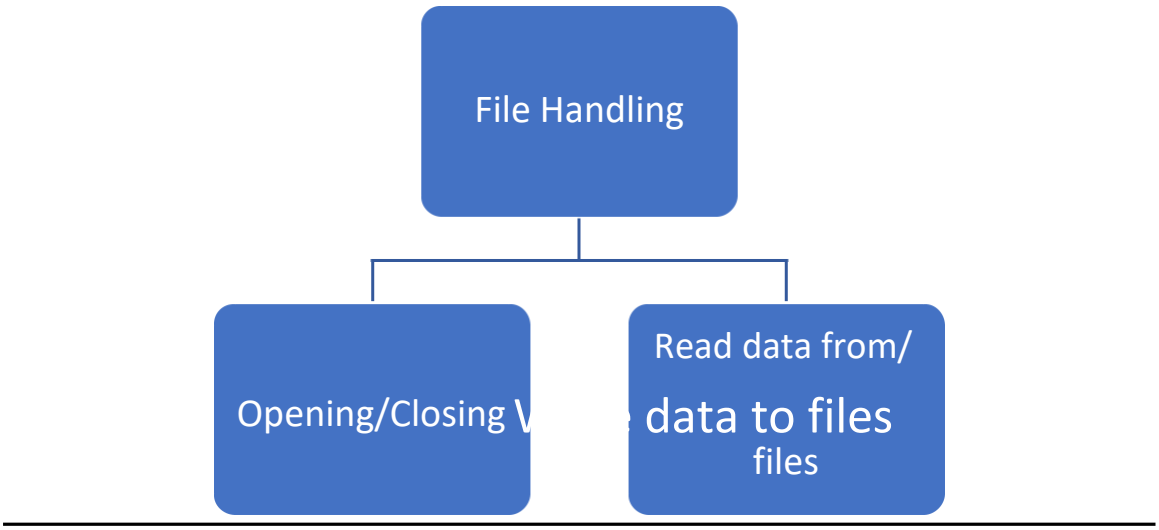
Limitation:-

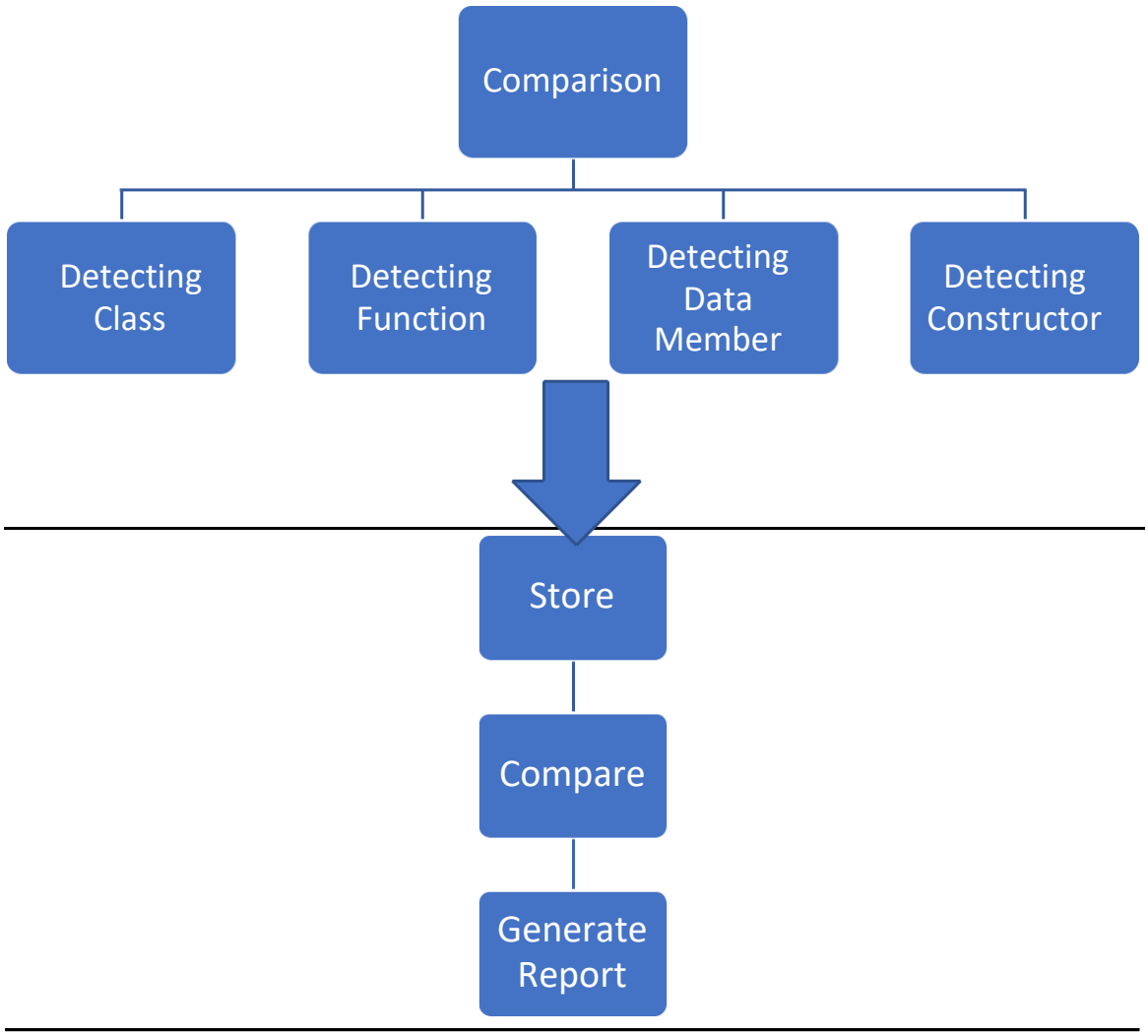
- Performing Text-based comparison only
- Cannot display specific changes
- Not able to generate any kind of report

CHAPTER 3 – METHODOLOGY ADOPTED

WORK BREAKDOWN STRUCTURE:-




File Handling
Opening/Closing
Read data from/
data to files
files



Tools and Technologies Used:-

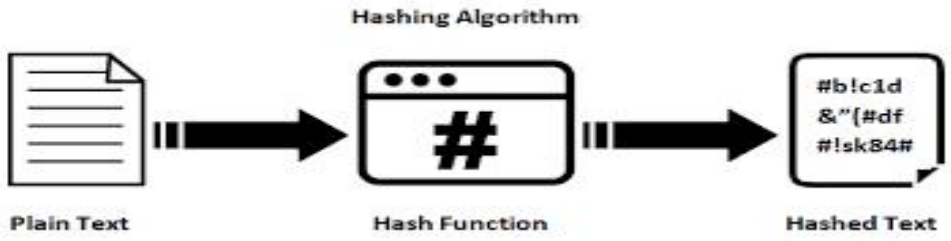
- **C++**

PROPOSED SOLUTION:-

For Comparison, We can use Hashing Concept.

DIGEST OR HASH FUNCTION

A digest or hash function is a process which transforms any random dataset in a fixed length character series, regardless of the size of input data. The output is called hash value or code, digest, image or hash. Often, the term “hash” is used both in reference to the hash function as to the hash value, which is the output of playing this function on a particular message. The data that are to be run through the hash function are called the message or preimage. The set formed by all possible messages or preimages is the message domain or message space.



Terminology:-

- Hash table:

  A data structure where the data is stored based upon its hashed key which is obtained using a hashing function.

- Hash function:

  A function which for a given data, outputs a value mapped to a fixed range. A hash table leverages the hash function to efficiently map data

such that it can be retrieved and updated quickly. Simply put, assume S = {s1, s2, s3, ..., sn} to be a set of objects that we wish to store into a map of size N, so we use a hash function H, such that for all s belonging to S; H(s) -> x, where x is guaranteed to lie in the range [1,N]

- Perfect Hash function:
  A hash function that maps each item into a unique slot (no collisions).

PROPERTIES IN A HASH FUNCTION

- Any given input may produce a fixed size numerical output
- This output is deterministic, that is; the same input message or dataset always yields the same output
- A minimum variation in the original message (one bit) must yield a completely different hash (diffusion).
- The hash algorithm must cover the entire hash space uniformly, which means that any output of a hash function has, in principle, the same probability of occurrence as any other. Therefore, all values in the hash space may be an output of the hash function.

DESCRIPTION OF A HASH FUNCTION

In general, hash functions work as follows:

- The input message is divided into blocks.
- Then the hash for the first block, a value with a fixed size, is calculated for the first block.
- Then, the hash for the second block is obtained and added to the previous output.
- This process is repeated until all blocks are calculated.

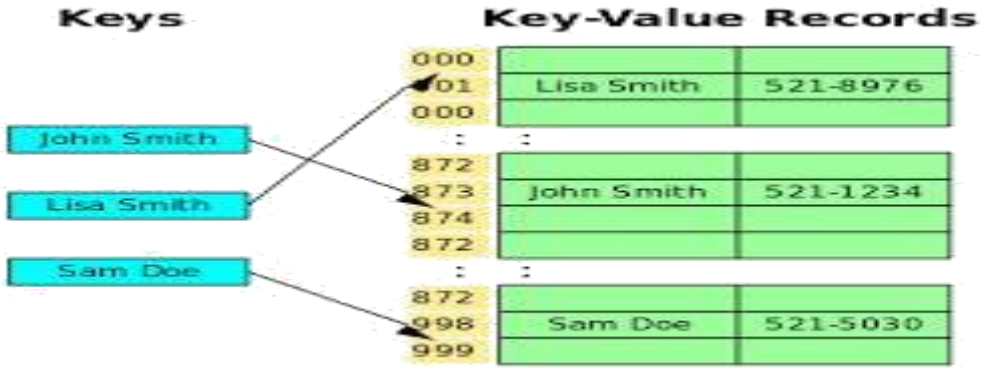
Why Hashing?

HASH AS AN UNIQUE IDENTIFIER:-

Hashing provides *constant time search, insert and delete operations* on average. This is why hashing is one of the most used data structure, example problems are, distinct elements, counting frequencies of items, finding duplicates, etc.

There are many other applications of hashing, including modern day cryptography hash functions.

Various programming languages have hash table based Data Structures. The basic idea is to create a key-value pair where key is supposed to be a unique value, whereas value can be same for different keys. This implementation is seen in unordered_set & unordered_map in C++, HashSet & HashMap in java, dict in python etc.

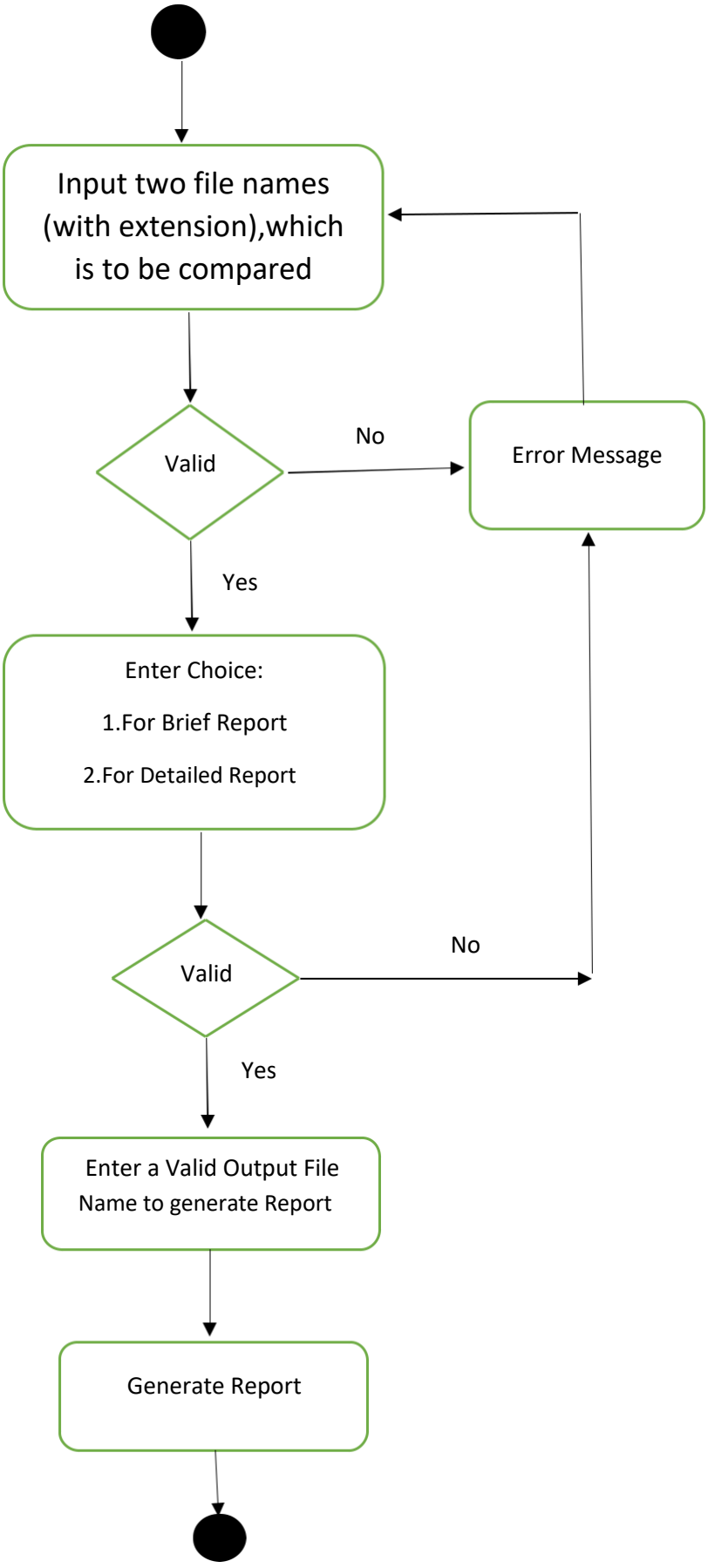


- unordered_map is used as a container that stores elements formed by the combination of key-value and a mapped value. The key value is used to uniquely identify the element and the mapped value is the content associated with the key. Both key and value can be of any type predefined or user-defined.

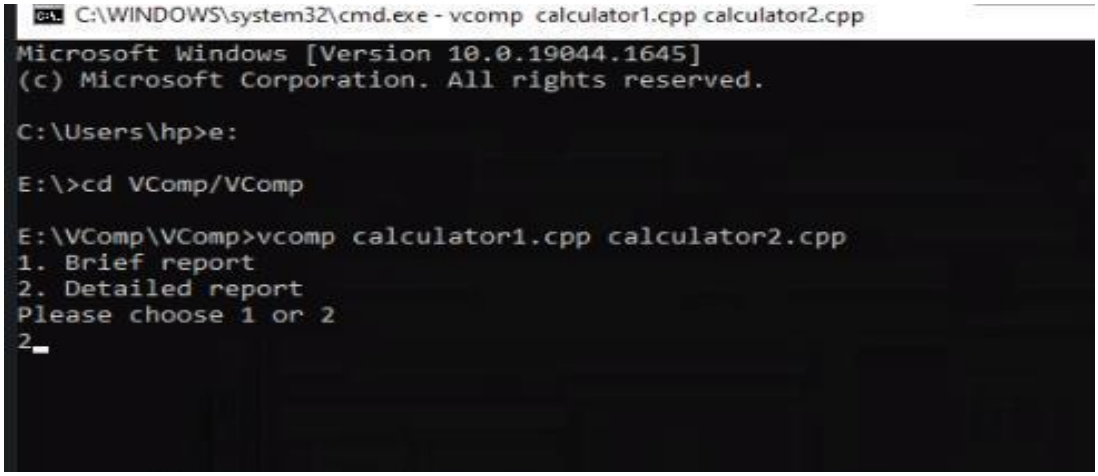
- Methods on unordered_map

A lot of functions are available which work on unordered_map. Most useful of them are – operator =, operator [], empty and size for capacity, begin and end for the iterator, find and count for lookup, insert and erase for modification.The STL library also provides functions to see internally used bucket count, bucket size, and also used hash function and various hash policies but they are less useful in real applications.We can iterate over all elements of unordered_map using Iterator.

Chapter 4 - DESIGN SPECIFICATIONS



System Screenshots:-



Ouput:- Report in Text Format-

File Edit Format View Help

File Comparison Report

Classes found in 1st file only

Classes found in 2nd file only

advance_calculator

Changes in Data-Members:

Variable	Class	Data-type in 1st file	Data-type in 2nd file
abc	calculator	int	char
x	calculator	long int	double
y	calculator	int	double
z	calculator	double	int
ch	calculator	Deleted	char
g	calculator	Deleted	int
h	calculator	Deleted	int
l	calculator	Deleted	int
p	calculator	int	Deleted
q	calculator	int	Deleted

Changes in Functions:

Function	Class	1st file	2nd file
add	calculator	Found	Found
main	NA	Found	Found
mul	calculator	Found	Found
divide	calculator	Not Found	Found

Constructors:

Name	1st file	2nd file
calculator	Found	Found

Ouput:- Report in HTML Format-

Changes in Classes

Classes Name	1st file	2nd file
advance_calculator	Not Found	Found
Car	Not Found	Found
calculator	Found	Found

Changes in Data members

Variable	Class	Data type in 1st file	Data type in 2nd file
abc	calculator	int	char
x	calculator	long int	int
z	calculator	double	int
excuses	calculator	Deleted	double
p	calculator	int	Deleted
parts	Car	Deleted	int
q	calculator	int	Deleted
r	calculator	int	Deleted
x	none	int	Deleted
y	calculator	int	Deleted

Changes in Functions

Function	Class	1st file	2nd file	Changes (if any)
add	calculator	Found	Found	No changes
main	none	Found	Found	No changes
divide	calculator	Not Found	Found	--
log	calculator	Found	Not Found	--
set_parts	Car	Not Found	Found	--
subtract	calculator	Not Found	Found	--

Constructors

Constructor Name	1st file	2nd file
Car	Not Found	Found
calculator	Found	Found

Chapter 5 - CONCLUSION AND FUTURE SCOPE

This Project provides a platform for developers or users to generate a brief or detailed report about changes made in two files in text or html format.

The project was successfully made and implemented in the company, still there is lot of improvement and features which we can further expand.

Some of them are :-

- ✓ Deployment in Maven Repository
- ✓ Make it Unix Compatible
- ✓ Adding some more features like comparing multiples files at a time
- ✓ Handling multiple reports

REFERENCES

- https://www.tutorialspoint.com/
- https://stackoverflow.com/
- https://www.geeksforgeeks.org/
- https://www.giac.org/paper/gsec/3294/study-hash-functions-cryptography/105433
- https://www.educative.io/