



**PRACTICAL JOURNAL OF PRINCIPLES OF
PROGRAMMING LANGUAGES
(CS-6002)**

BE: Third-Year

Department of Computer Science & Engineering

Name of Student : _____
Branch & Section : _____
Roll Number : _____
Year : _____

**Department of Computer Science & Engineering
AITR, INDORE**

**ACROPOLIS INSTITUTE OF TECHNOLOGY & RESEARCH,
INDORE**

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the experimental work entered in this journal as per the BE **Third** year syllabus prescribed by the RGPV was done by Mr. /Ms. _____ BE **III-year VI** semester in the **Principles of Programming Languages** Laboratory of this institute during the academic year **2019-2020**.

Signature of Head

Signature of Faculty

Name of Student:Naved Saxena		Class: CS-3
Enrollment No: 0827CS161143		Batch: B1
Date of Experiment:	Date of Submission:	Submitted on:
Remarks by faculty:		Grade:
Signature of student:		Signature of Faculty:

Name of Program-1:WRITE A CASE STUDY OF LEXICAL ANALYZER USING LEX

1. Introduction to Lexical Analyzer:

Lexical analysis is an important phase of compilation operated at beginning. The language processor that are written in the form of sentences, it takes the modified code from them. It removes the white code and source code by that it breaks the syntax into different tokens.

It generates the error when it found some invalid tokens. It works closely with the syntax analyzer. It reads streams of character from the source code, checks for legal tokens, and passes the data to the syntax analyzer whenever it demands for it. A lexer can be class implementation, having a constructor which takes a string as an input parameter (representing the source code to perform lexical analysis on). It shows a method to recognize and then it returns the next token in the input. The aim of the lexical analyzer is to read the input characters of the source program, group them into lexemes, and produce as output a sequence of tokens for each lexeme in the source program. The lexical analysis phase of a compiler performs the task of reading the source program as a file of character and dividing up into the tokens. It usually implements as a subroutine or co-routine of parser. It is front end of a compiler.

2. Why we need Lexical Analyzer:

Its main task is to read input character and produce as output a sequence of tokens that parser uses for syntax analysis.

- It separates the input source code into tokens.
- Removing out the unnecessary white spaces from the source code.
- Removing the comments.
- Keeping tracks of line numbers.
- Preprocessing of macros.

- The attribute influences the translation of tokens.

3. Literature Survey:

An exploration on lexical analysis is published by IEEE. It was published in 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). The word lexical in lexical analysis, its meaning is extracted from the word “lexeme”. A lexical analyzer is used in various applications like text editors, information retrieval system, pattern recognition programs and language compilers. In this paper to discussion is about language compilers. Lexical analyzer being the first phase of the compilation process it deals with the processing of input language. Discussion is also extended to multi-core environment. Input to the lexical analyzer is the source program itself. It scans the source program character by character and produces meaningful sequence called Tokens. Due to its scanning property this phase is also called scanner. Lexical analyzer stores the tokens in symbol table and as well as sends it to next phase as an input.

4. Advantages:

- This technique used to implement lexical analyzers can also be applied to other areas such as query languages and information retrieval system.
- It looked for patterns corresponding to imperfections in the string of line segments.
- It can utilize the best known pattern-matching algorithms and thereby create efficient lexical analyzers for people who are not experts in pattern-matching technique.
- Scanners perform pattern matching process.

5. Disadvantages:

- There are several reasons for separating the analysis phase of compiling into lexical analysis and parsing.
- Simpler design of the parser.
- Unnecessary tokens can be removed.
- The lexical analysis phase is the most time consuming phase in compilation.
- Portability of the compiler is isolated.

6. Conclusion:

It breaks the syntax into tokens. It removes the white space and source code. It takes the modified code for language processor that are written in sentences. The attribute so lexical analyzer influences the translation of tokens. It also removes the comment in the compiler design. Correlate the error messages from the compiler. To specify the lexical analyzer, the lex is used. It is also time consuming in compiling the program. It decreases the portability of the compiler. especially when we are

RESULTS



Sentence Wise Result



Matched Sources



Document View

UNIQUE	Name of Program-1: WRITE A CASE STUDY OF LEXICAL ANALYZER USING LEX.
UNIQUE	Lexical analysis is an important phase of compilation operated at beginning.
UNIQUE	The language processor that are written in the form of sentences, it takes the modi...
UNIQUE	It removes the white code and source code by that it breaks the syntax into differe...
UNIQUE	It generates the error when it found some invalid tokens.
UNIQUE	It reads streams of character from the source code, checks for legal tokens, and pas...
UNIQUE	class implementation, having a constructor which takes a string as an input parame...
UNIQUE	It shows a method to recognize and then it returns the next token in the input.
UNIQUE	The aim of the lexical analyzer is to read the input characters of the source progra...
UNIQUE	The lexical analysis phase of a compiler performs the task of reading the source pr...
UNIQUE	It usually implements as a subroutine or co-routine of parser.
UNIQUE	Its main task is to read input character and produce as output a sequence of tokens...

Enrollment No: 0827CS161143		Batch: B1
Date of Experiment:	Date of Submission:	Submitted on:
Remarks by faculty:		Grade:
Signature of student:		Signature of Faculty:

Name of Program-2: WRITE A CASE STUDY OF VIRTUAL MACHINES

1. Introduction:

A Virtual Machine is a software or a computer file that provides an application environment to any guests to meet some specific requirement. It enables morphing of a system on another system, where the first system is Virtual image that is the need of the guests whereas the latter one is the dedicated physical system also called as host. There can be multiple guests on a single host.

A software called Hypervisor provides the virtual environment, looks and controls the complete emulation of the Virtual Machine on the host. A Virtual Machine have the control over host system and provides.

2. Why to Use Virtual Machine:

- A Virtual Machine is widely used to run different Operating System on a single physical machine. It enables the user to get hands on different software environment without having different physical resources.
- A Virtual Machine is also helpful in checking of whether the program, file or software we are using is damaged or incompatible or not.
- In the recent world of research and development, Virtual Machines provide an edge to the researchers, computer scientists, to test different things and they can make mistakes even huge ones too. They make the research flexible.
- Virtual Machines enable a user to test on different stuff without having any prerequisites for technological requirements mostly on the hardware requirements.
- Virtual Machines are used to reconstruct or clone different machines for specific requirement.

3. Literature Survey:

Survey Paper Topic – A Survey on Virtual Machine Security

Author Name – Jenni Susan Reuben, Helsinki University of Technology

Summary – This paper is concerned about the security problems in virtualization technologies. Main focus is on some open security vulnerabilities that virtual machine produce in the surroundings. This paper does not provide a great solution to all the security threats but provides an idea to avoid these threats. Paper includes both the benefits and drawbacks in terms of security that virtualization provides. Resource sharing and Isolation are the main two benefits that VM provides. It also depicts the architecture of Virtual Machine and explains the terms like full and para virtualization along with application, hardware support, resource virtualization. Security vulnerabilities like VM monitoring from the host, Denial of Service etc... are explained and also the solution to these problems are provided. The overall paper has presented security flaws in VM.

4. Advantages:

- 1.Familiar interfaces.
- 2.Isolation – each OS can run individually with its own virtual resources.
- 3.High Availability – if one VM server fails then you can easily grab the data from another VM server.
- 4.Scalability – you can easily connect or disconnect resources.
- 5.Backup – you can always have backup using VMDK tool.
- 6.Reduced cost – VM supports hardware sharing thus saving cost and also you can run multiple OS on a single system.

5. Disadvantages:

- 1.Difficult Access – you cannot access the hardware directly.
- 2.More RAM utilization – since each VM occupies separate memory area for the same copies.
- 3.More disk space utilization – since all the files for each OS is installed on each VM.
- 4.Less Efficient – A VM is less efficient as compared to the actual machine when it tries to indirectly access the host Hard Drive.


6. Conclusion:


Virtual Machine which can also be said as virtualization is a very mandatory need for today's technology since it provides a whole lot of benefits which a system can easily afford without risking any of its parts. Although Virtual Machines have some drawbacks regarding security but still have huge benefits. Virtualization is a powerful solution to reduce the operational costs in today's computing but if done wrong it becomes a threat to the environment.


RESULTS

100% Completed: 100% Checked

0% Plagiarism 100% Unique

 Sentence Wise Result

 Matched Sources

 Document View

UNIQUE	Name of Program-1: WRITE A CASE STUDY ON VIRTUAL MACHINES.
UNIQUE	A Virtual Machine is a software or a computer file that provides an application envi...
UNIQUE	It enables morphing of a system on another system, where the first system is Virtu...
UNIQUE	There can be multiple guests on a single host.
UNIQUE	A software called Hypervisor provides the virtual environment, looks and controls t...
UNIQUE	A Virtual Machine have the control over host system and provides.
UNIQUE	• A Virtual Machine is widely used to run different Operating System on a single p...
UNIQUE	It enables the user to get hands on different software environment without having ...
UNIQUE	• A Virtual Machine is also helpful in checking of whether the program, file or soft...
UNIQUE	• Virtual Machines enable an user to test on different stuff without having any prer...
UNIQUE	• Virtual Machines are used to reconstruct or clone different machines for specific r...

Name of Student:Naved Saxena		Class: CS-3
Enrolment No: 0827CS161143		Batch: B1
Date of Experiment:	Date of Submission:	Submitted on:
Remarks by faculty:		Grade:
Signature of student:		Signature of Faculty:

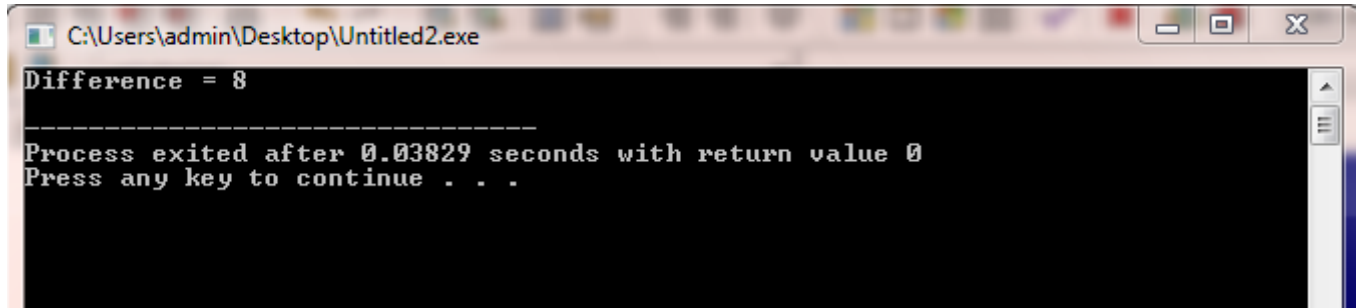
Name of Program-3: GIVEN A SQUARE MATRIX OF SIZE $N \times N$, DESIGN A PROGRAM TO CALCULATE THE ABSOLUTE DIFFERENCE BETWEEN THE SUMS OF ITS DIAGONALS.

1. Source Code:

```
#include <bits/stdc++.h>
using namespace std;
const int MAX = 100;
void DiagonalDiff(int mat[][MAX], int n)
{
    int principal = 0, secondary = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (i == j) {
                principal += mat[i][j];
                if ((i + j) == (n - 1))
                    secondary += mat[i][j];
            }
        }
    }
    cout<<"Difference = "<<(principal - secondary)<<endl;
}
int main()
{
    int a[][MAX] = { { 5, 2, 3, 4 }, { 5, 6, 7, 8 },
                     { 1, 2, 3, 4 }, { 1, 6, 7, 8 } };

    DiagonalDiff(a, 4);
    return 0;
}
```

2. Output:



A screenshot of a Windows command prompt window titled "C:\Users\admin\Desktop\Untitled2.exe". The window has a black background with white text. The output displayed is: "Difference = 8", followed by a horizontal line, "Process exited after 0.03829 seconds with return value 0", and "Press any key to continue . . .".

```
C:\Users\admin\Desktop\Untitled2.exe
Difference = 8
-----
Process exited after 0.03829 seconds with return value 0
Press any key to continue . . .
```

PRINCIPLES OF PROGRAMMING LANGUAGES I

Name of Student: Naved Saxena		Class: CS-3
Enrolment No: 0827CS161143		Batch: B1
Date of Experiment:	Date of Submission:	Submitted on:
Remarks by faculty:		Grade:
Signature of student:		Signature of Faculty:

Name of Program-4: WRITE A PROGRAM THAT READS THREE DIMENSIONAL ARRAY AND DISPLAY IT.

1. Source Code:

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"Enter a 2x2x2 multidimensional array:"<<endl;
    int a[2][2][2];
    for(int i=0;i<2;i++)
    { for(int j=0;j<2;j++)
        { for(int k=0;k<2;k++)
            { cin>>a[i][j][k]; } } }
    cout<<"The array is:"<<endl;
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<2;j++)
        { for(int k=0;k<2;k++)
            { cout<<a[i][j][k]; }
            cout<<endl;
        } } }
}
```

2. Output:

```
Enter a 2x2x2 multidimensional array:
1 2
2 3
3 4
4 5
The array is:
12
23
34
45
```

Name of Student: Naved Saxena		Class: CS-3
Enrolment No: 0827CS161143		Batch: B1
Date of Experiment:	Date of Submission:	Submitted on:
Remarks by faculty:		Grade:
Signature of student:		Signature of Faculty:

Name of Program-5: DESIGN A WEBPAGE IN PHP.

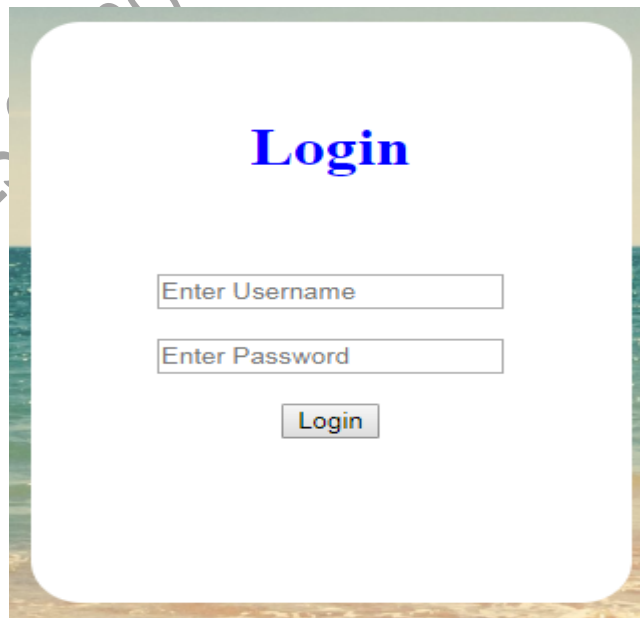
1. Source Code:

```
<html>
    <head>
        <title> Login </title>
        <style>
            body { background-image:url(bg.jpg);background-
position: center;background-repeat: no-repeat;background-size: cover;height:
100%;}
            h1 { color:Blue;}
            .login
{ background:white;width:300px;height:350px;border-radius:25px;}
        </style>
    </head>
    <body>
        <center>
<?php
$un = $_POST['un'];
$pass = $_POST['pass'];
if(isset($_POST["submit"]))
{
if($un == 'Admin' && $pass == '123')
{
echo "<br/><br/><div class='login'><br/><br/><h1><font color='Green'>Access
Granted</font><br/><br/></h1><h2> Welcome Admin </h2></div>";
exit;
}
else
```

```
{
echo "<br/><br/><div class='login'><br/><br/><h1><font
color='Red'>Unauthorized Access</font><br/><br/></div>";
exit;
}
}
?>

<br/><br/><div class="login">
    <br/><br/><h1> Login </h1><br/><br/>
    <form method="POST" action="Login.php">
        <input type="text" name="un"
placeholder="Enter Username" /><br/><br/>
        <input type="password" name="pass"
placeholder="Enter Password" /><br/><br/>
        <input type="submit" name="submit"
value="Login" />
    </form>
</div>
</center>
</body>
</html>
```

2. Output:



Login

Enter Username

Enter Password

Login

Name of Student: Naved Saxena		Class: CS-3
Enrolment No: 0827CS161143		Batch: B1
Date of Experiment:	Date of Submission:	Submitted on:
Remarks by faculty:		Grade:
Signature of student:		Signature of Faculty:

Name of Program-6: WRITE A PROGRAM IN JAVA TO IMPLEMENT EXCEPTION HANDLING.

1. Source Code:

```
import java.util.Scanner;
class Division {
    public static void main(String[] args) {
        int a, b, result;
        Scanner input = new Scanner(System.in);
        System.out.println("Input two integers");
        a = input.nextInt();
        b = input.nextInt();
        result = a / b;
        System.out.println("Result = " + result);
    }
}
```

2. Output:

```
E:\Java>javac Division.java
E:\Java>java Division
Input two integers
4 2
Result = 2

E:\Java>java Division
Input two integers
7 0
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at Division.main(Division.java:14)
E:\Java>_
```