



Aim Introduction : objective scope and outcome of the course

Objective The objective of this course is to explore the principles of compiler. To introduce the major concept areas of language translation and compiler design

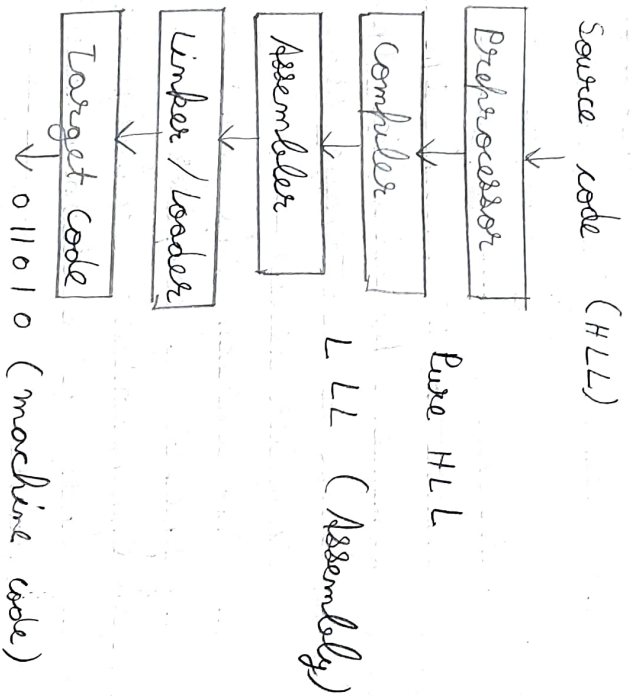
→ To enrich the knowledge in various phases of compiler and its use, code optimization techniques, machine code generation and use of symbol tables.

→ To extend the knowledge of parser by parsing LL parser theory, compiler theory, error handler, YACC etc

Scope

→ Compiler design is a basic element of programming language research many languages researchers write compilers for the language they design

* Language processing system



HLL → High level language

LLL → Low level language



Estd. Yr. 2000

ARYA College of Engineering & I.T.

Kukas, Jaipur

Experiment No.:

Date:

Language processing system The program that are written in high-level languages (eg C++, java etc) are packed into a sequence of component to generate desired machine code are known as language processing system and has following components

(1) Preprocessor It takes source code as input and produces modified source code as output

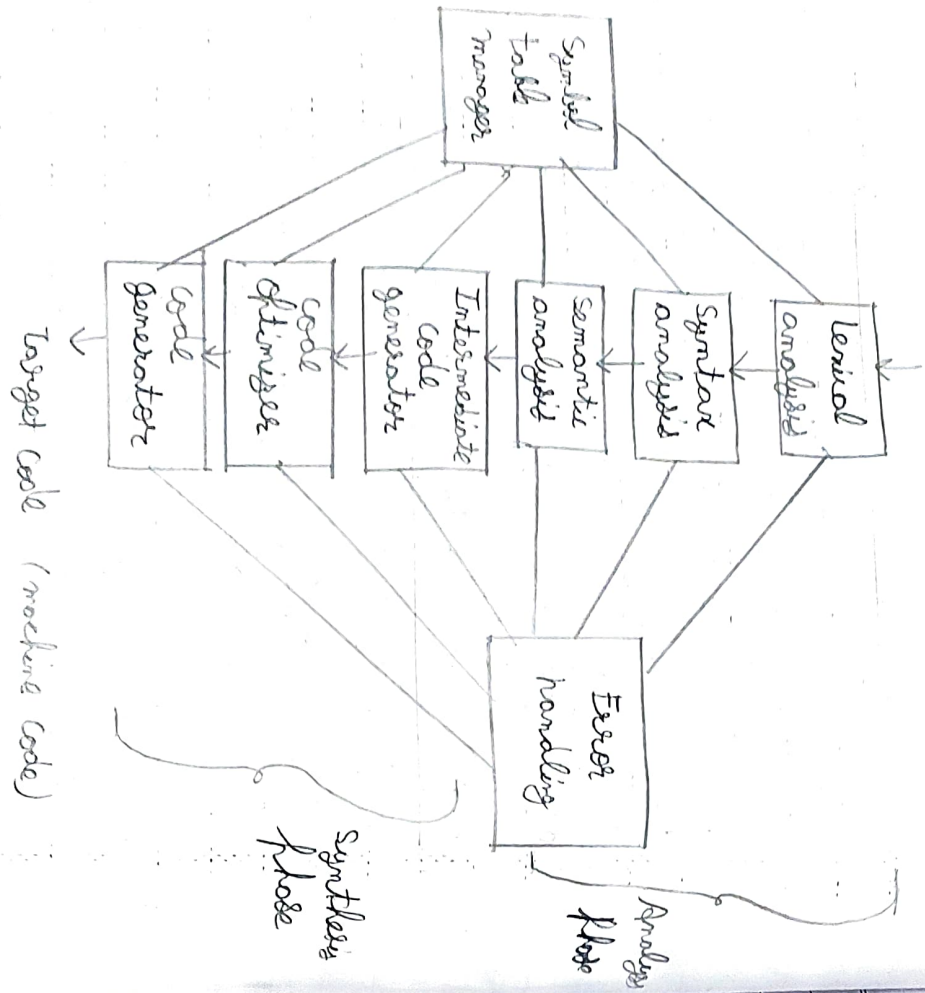
(2) compiler Takes modified code as input and produces assembly code

(3) Assembler takes assembly language code and produces machine code as output

(4) linker / loader takes collection of object and link and load them in memory

(5) Target code Low level and machine specific code that machine can easily understand

Source Code (HLL)



Phases of compiler



Experiment No.:

Date:

Different phases of compiler design

(i) Analysis phase This include

lexical analysis, syntax analysis, Semantic analysis and they construct sequence of token according to language's grammar

(ii)

Synthesis phase This include intermediate code generator, code optimizer, code generator and generates validate parse tree to a machine code

OUTPUT 2

opening file in write mode
Enter the data for the file.
This is the class of compiler design
opening file in read mode
This is the class of compiler design

Q2

```
C program to open file in write and
read mode
#include <stdio.h>
#include <conio.h>
void main() {
    FILE *fp;
    char str[100];
    char ch;
    clrscr();
    fp = fopen("C:\\TC\\test.txt", "w");
    printf("Opening the file in write mode\n");
    if (fp == NULL) {
        printf("Error opening file\n");
    }
    printf("Enter the data for file\n");
    gets(str);
    printf(fp, "%s", str);
    fclose(fp);
    printf("Opening file in read mode\n");
    fp = fopen("C:\\TC\\test.txt", "r");
    if (fp == NULL) {
        printf("Error opening the file\n");
    }
}
```



Estd. Yr. 2000

ARYA College of Engineering & I.T.

ARYA 1ST OLD CAMPUS

www.aryacollege.in

Kukas, Jaipur

Experiment No. :

Date :



Experiment No. :

Date :

```
while ((ch = getc (fp)) != EOF) {  
    putchar (ch);  
}  
  
fclose (fp);  
getch (); }
```

Q 3 Write a C program to identify whether given string is keyword or not

Ans

```
#include <stdio.h>  
#include <conio.h>  
#include <string.h>  
void main () {  
    char keywords [32][10] = {  
        "auto", "break", "case", "char", "const",  
        "continue", "default", "do", "int", "long",  
        "register", "return", "short", "signed",  
        "sizeof", "static", "struct", "switch",  
        "typedef", "union", "unsigned", "void",  
        "volatile", "while", "for", "goto", "if",  
        "else", "enum", "extern", "main" };
```


OUT PUT 3

Enter a string: main
Given string main is keyword

```
char str[100];  
int i, flag = 0;  
char c;  
printf("Enter a string");  
gets(str);  
for (i = 0; i < 32; i++) {  
    if (strcmp(keywords[i], str) == 0) {  
        flag = 1;  
        break; } }  
if (flag == 1) {  
    printf("Given string %s is  
    keyword", str);  
}  
else {  
    printf("Given string is not a  
    keyword");  
}  
getch();  
}
```



Estd. Yr. 2000

ARYA College of Engineering & I.T.

ARYA 1st OLD CAMPUS

www.aryacollege.in

Kukas, Jaipur

Experiment No.:

Date:

OUTPUT

Enter the input for the file
hi int long main
total no of keywords in file are: 2



Experiment No.:

Date:

Q1

Count total no of keywords in a file [Taking file from the user]

Ans

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main() {
    FILE *fp;
    int count = 0;
    char ch[100];
    char str[200];

    char keywords[32][10] = { "auto",
    "break", "case", "char", "const", "continue",
    "default", "do", "int", "long", "register",
    "return", "short", "signed", "sizeof",
    "static", "struct", "switch", "typedef",
    "union", "unsigned", "void", "volatile",
    "while", "for", "goto", "if", "else",
    "enum", "extern", "main" };

    close();
    fp = fopen("C:\\\\TC\\\\test.txt", "w");
    if (fp == NULL) { printf("Error"); }
```



Experiment No. :

Date :

```
printf("Enter the data input for the file \n");
fgets(str, sizeof(str), stdin);
fprintf(fp, "%s", str);
fclose(fp);
fp = fopen("C:\\TC\\test.txt", "r");
while((fscanf(fp, "%s", ch)) != EOF) {
    for(i = 0; i < 32; i++) {
        if(strcmp(keyword[i], ch) == 0) {
            count++;
        }
    }
    fclose(fp);
    printf("total no of keyword in\nfile are %d", count);
    getch();
}
```