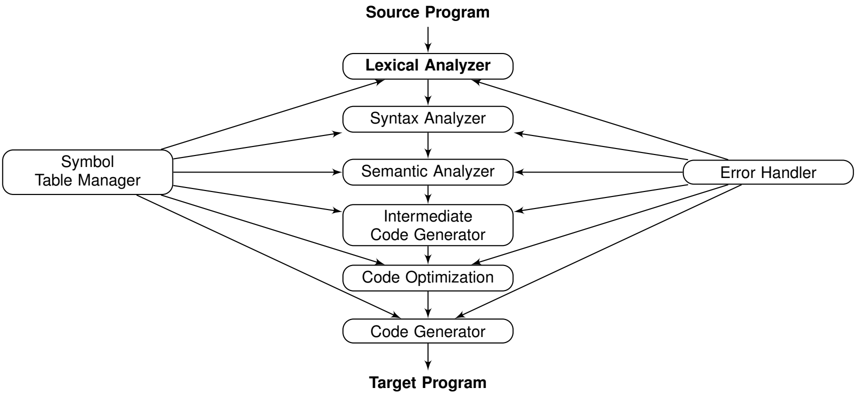
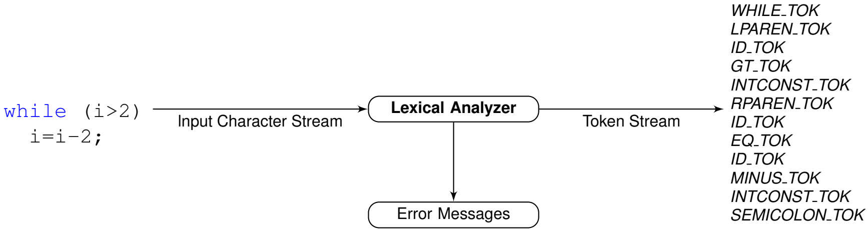
Phases Of Compilation



Lexical Analyzer

- converts the input program into a sequence of Tokens.
- can be implemented with the help of *Finite Automata*.

Lexical Analyzer



```
while (i>2)
  i=i-2;
```

Input Character Stream

Lexical Analyzer

Error Messages

Token Stream

WHILE_TOK
LPAREN_TOK
ID_TOK
GT_TOK
INTCONST_TOK
RPAREN_TOK
ID_TOK
EQ_TOK
ID_TOK
MINUS_TOK
INTCONST_TOK
SEMICOLON_TOK

Programmer‘s View

```
FILE *yyin;
char *yytext;
main(int argc, char *argv[]){
int token;
if (argc != 2){

}else{
       yyin = fopen(argv[1], "r");
       while(!feof(yyin)){
                token = yylex();
                printf("%d", token);
       }
       fclose(yyin);
     }
}
```

```
int yylex(){
    ...
    ...
}
```

Loop and switch Approach

```
/* Single caharacter lexemes */
#define LPAREN_TOK '('
#define GT_TOK '>'
#define RPAREN_TOK ')'
#define EQ_TOK '='
#define MINUS_TOK '-'
#define SEMICOLON_TOK ';'
/*....................
.......................*/
/* Reserved words */
#define WHILE_TOK 256
/*....................
.......................*/
/* Identifier, constants..*/
#define ID_TOK 350
#define INTCONST 351
/*....................
.......................*/
```