# Compiler Construction

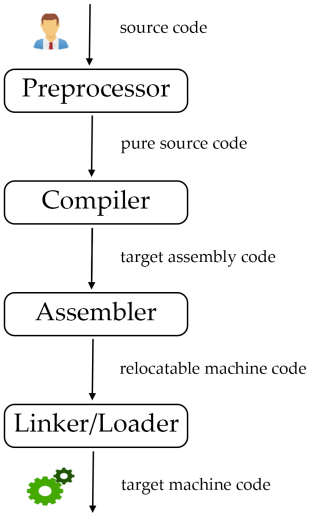
## Lecture # 03

Mr. Usman Wajid

*usman.wajid@nu.edu.pk*

National University
of Computer & Emerging Sciences

# A Language-processing System



# Structure of a Compiler

1. **Analysis**
2. **Synthesis**

# Structure of a Compiler

1. **Analysis:**

# Structure of a Compiler

1. **Analysis:**
   - division of the source code into distinctive pieces

# Structure of a Compiler

1. **Analysis:**
   - division of the source code into distinctive pieces
   - implementation of grammatical structure

# Structure of a Compiler

1. **Analysis:**
   - division of the source code into distinctive pieces
   - implementation of grammatical structure
   - an intermediate representation in the form of a syntax tree

# Structure of a Compiler

1. **Analysis:**
   - division of the source code into distinctive pieces
   - implementation of grammatical structure
   - an intermediate representation in the form of a syntax tree
   - the operations in the syntax tree are arranged in a hierarchical manner, the leaf operations will execute first and the root operation will execute last

# Structure of a Compiler

1. **Analysis:**
   - division of the source code into distinctive pieces
   - implementation of grammatical structure
   - an intermediate representation in the form of a syntax tree
   - the operations in the syntax tree are arranged in a hierarchical manner, the leaf operations will execute first and the root operation will execute last
   - also known as the font-end of a compiler

# Structure of a Compiler

2. **Synthesis:**
   - an intermediate representation (syntax tree) and symbol table is provide
   - transformation of intermediate code into machine code
   - also known as the back-end of a compiler

# Symbol table

- A symbol table is maintained carrying information regarding each phase of a compiler

# Symbol table

- A symbol table is maintained carrying information regarding each phase of a compiler
- an information is stored, retrieved or updated at each phase of a compiler

# Symbol table

- A symbol table is maintained carrying information regarding each phase of a compiler
- an information is stored, retrieved or updated at each phase of a compiler
- sample symbol table for lexical analysis phase,

```
if (i==j);\n\tz=1;\nelse;\n\tz=0;\nendif;
```

| Line | Token | Lexeme |
|---|---|---|
| 1 | BLOCK_COMMAND | if |
| 1 | OPEN_PAREN | ( |
| 1 | ID | i |
| 1 | OP_RELATION | == |
| 1 | ID | j |
| 1 | CLOSE_PAREN | ) |
| 1 | ENDLINE | ; |
| 2 | ID | z |
| 2 | ASSIGN | = |
| 2 | NUMBER | 1 |
| 2 | ENDLINE | ; |
| 3 | BLOCK_COMMAND | else |

- Etc...

# Phases of Compiler