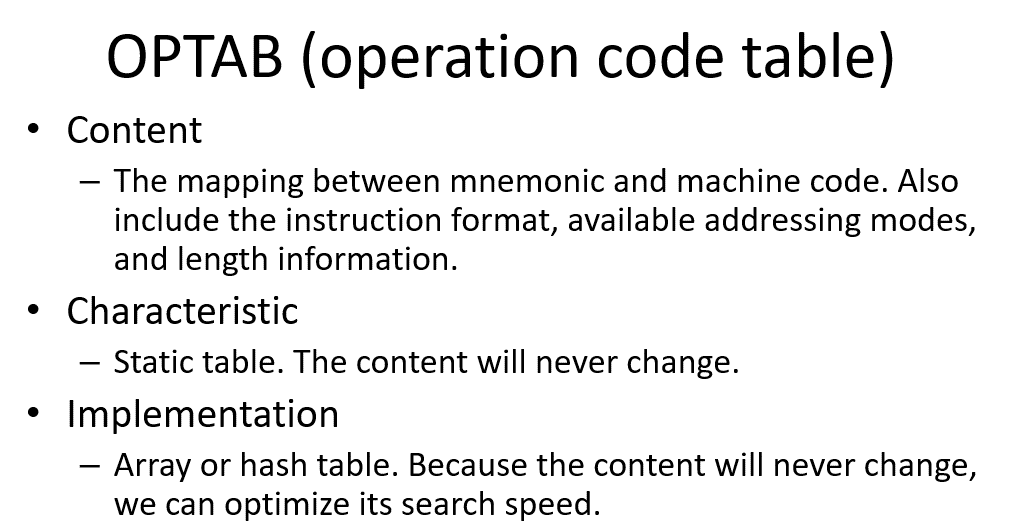
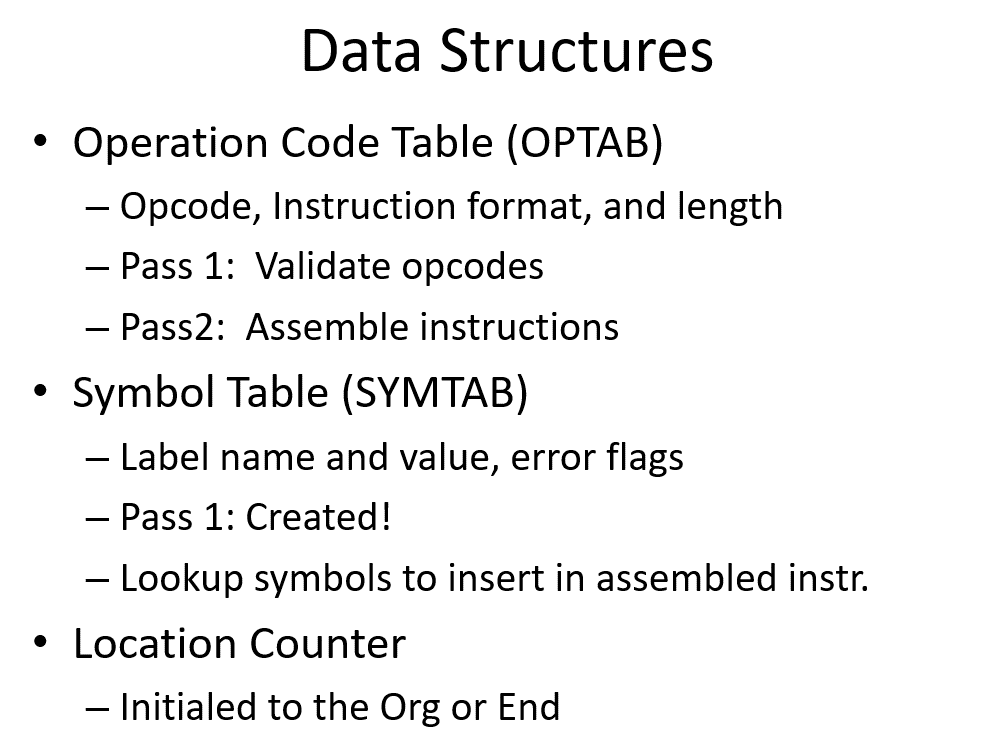
**Tutorial 5 Answers – Made By U19CS012**

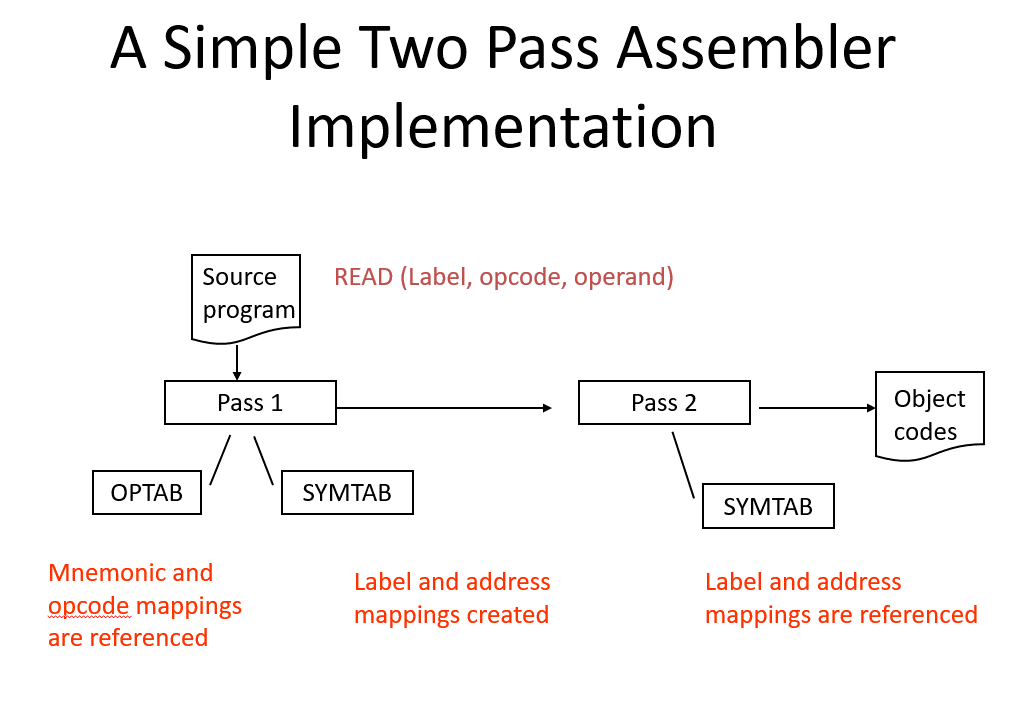
1. What are the uses of OPTAB (Mnemonic operation code) and SYMTAB (Symbol table) during assembling process? Specify the uses of each during pass 1 and pass 2 of a two pass assembler.

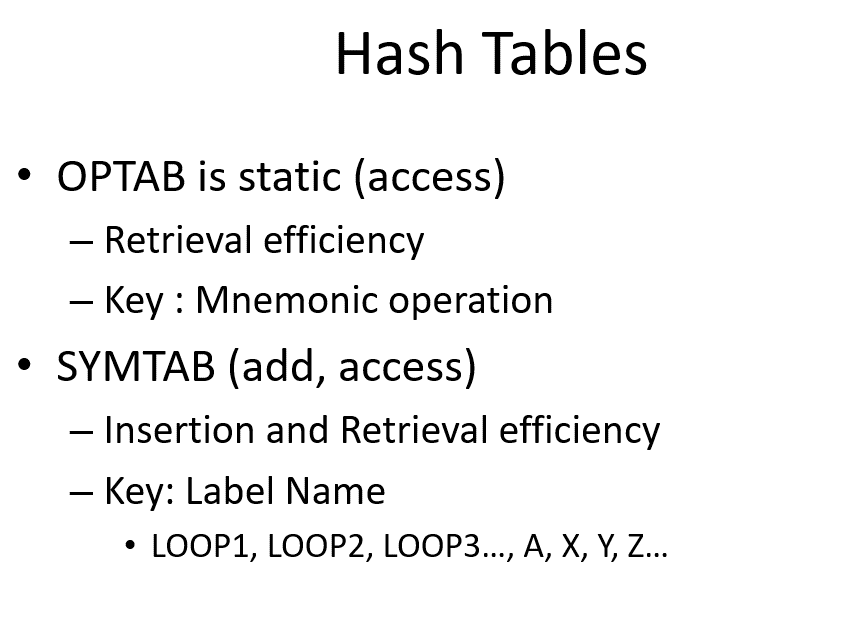
* **Defn** - OPTAB is used to look up mnemonic operation codes and translate them to their machine language equivalents.
* During **Pass 1**, OPTAB is used to look up and validate operation coded in the source program and to find the instruction length for incrementing LOCCTR.
* In **Pass 2** , it is used to translate the operation codes to machine language



* **Defn** - The Symbol table (SYMTAB) includes the name and value (address) for each label in the source program, together with flags to indicate error conditions (e.g., a symbol defined in two different places).
* During **Pass 1** of the assembler, labels are entered into SYMTAB as they are encountered in the source program, along with their assigned addresses (form LOCCTR).
* During **Pass 2**, symbols used as operands are looked up in SYMTAB to obtain the addresses to be inserted in the assembled instructions.
* **Characteristics** - Dynamic table (I.e., symbols may be inserted, deleted, or searched in the table)
* **Implementation** - Hash table can be used to speed up search
* Because variable names may be very similar (e.g., LOOP1, LOOP2), the selected hash function must perform well with such non-random keys







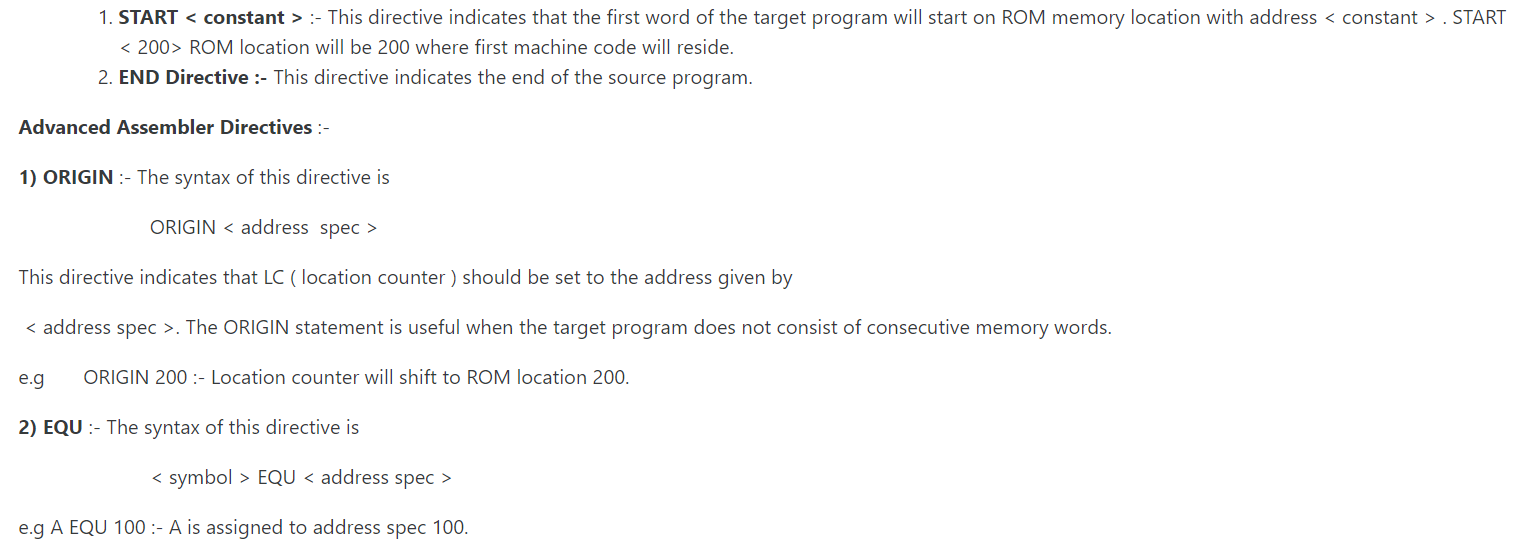
2. What are assembler directives? List any three assembler directives.

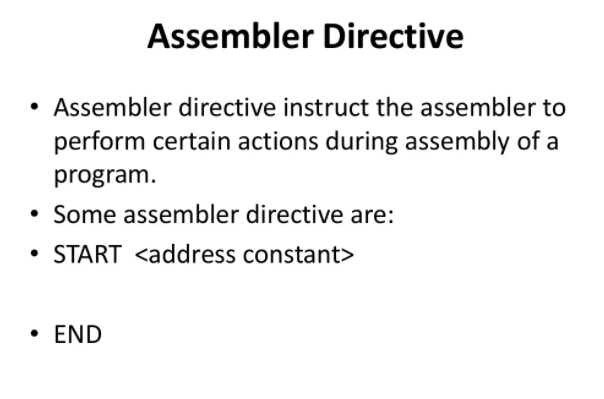
Assembler directives are instructions that direct the assembler to do something

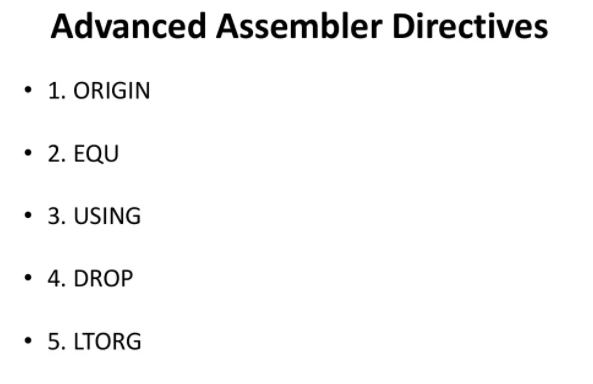
Directives do many things; some tell the assembler to set aside space for variables, others tell the assembler to include additional source files, and others establish the start address for your program.

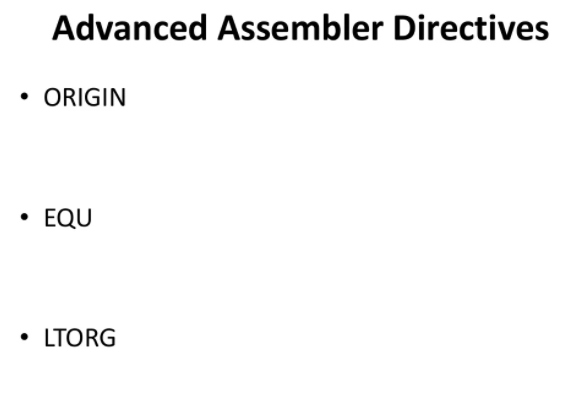
Assemble directives can’t generate machine code.

The directives available are shown below:









Link - <https://www.slideshare.net/ReshmaKapadi/system-programming-unit-1-introduction-86502311>

3. Find out addresses of variable using LC.

Step 1: First identify all variables in your program.

Step2: Replace all symbolic address with numeric address.

Step3: Replace symbolic opcodes by machine operation codes.

START 101

READ N

MOVER BREG, ONE

MOVEM BREG, TERM

AGAIN MULT BREG, TERM

MOVER CREG, TERM

ADD CREG, ONE

MOVEM CREG, TERM

COMP CREG, N

BC LE, AGAIN

MOVEM BREG, RESULT

PRINT RESULT

STOP

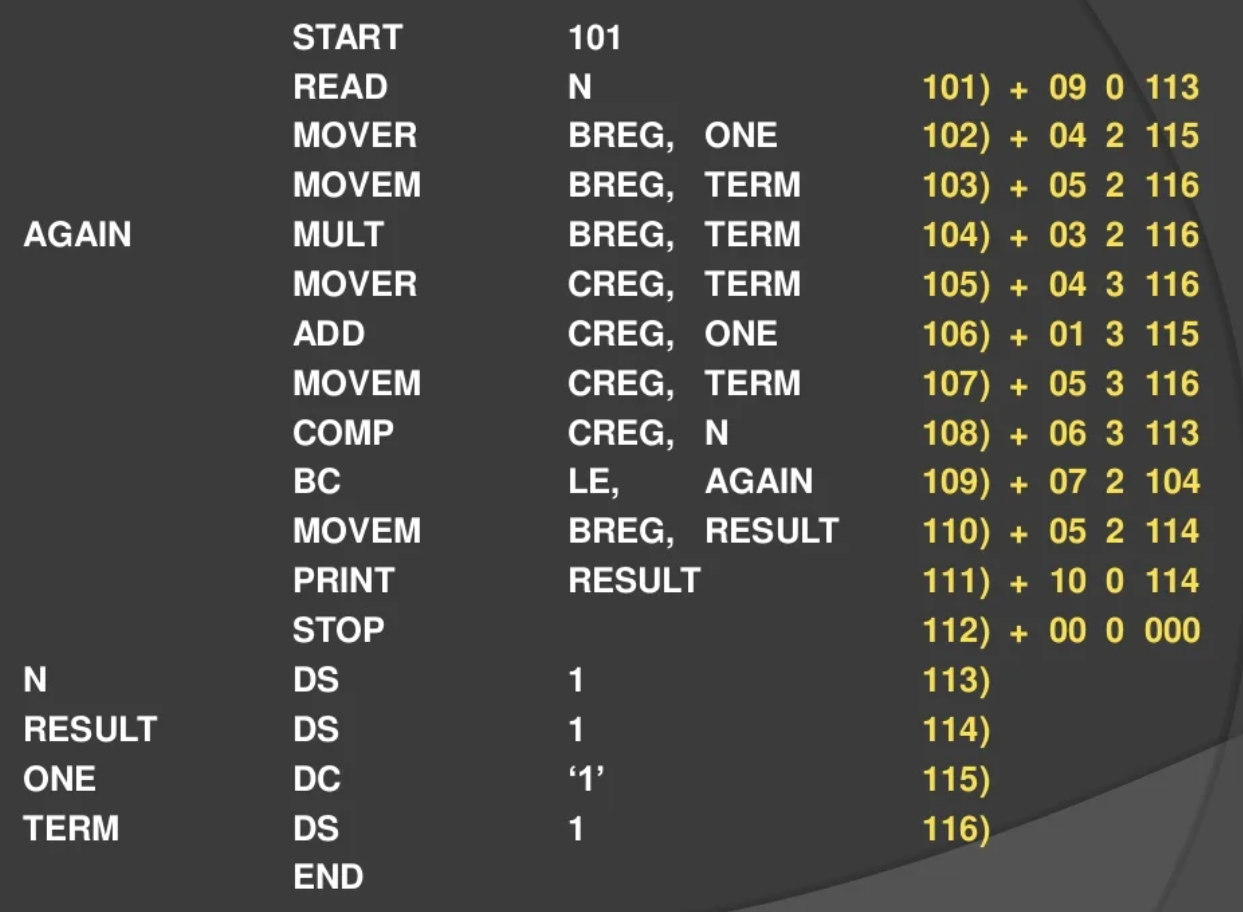
N DS 1

RESULT DS 1

ONE DC ‘1’

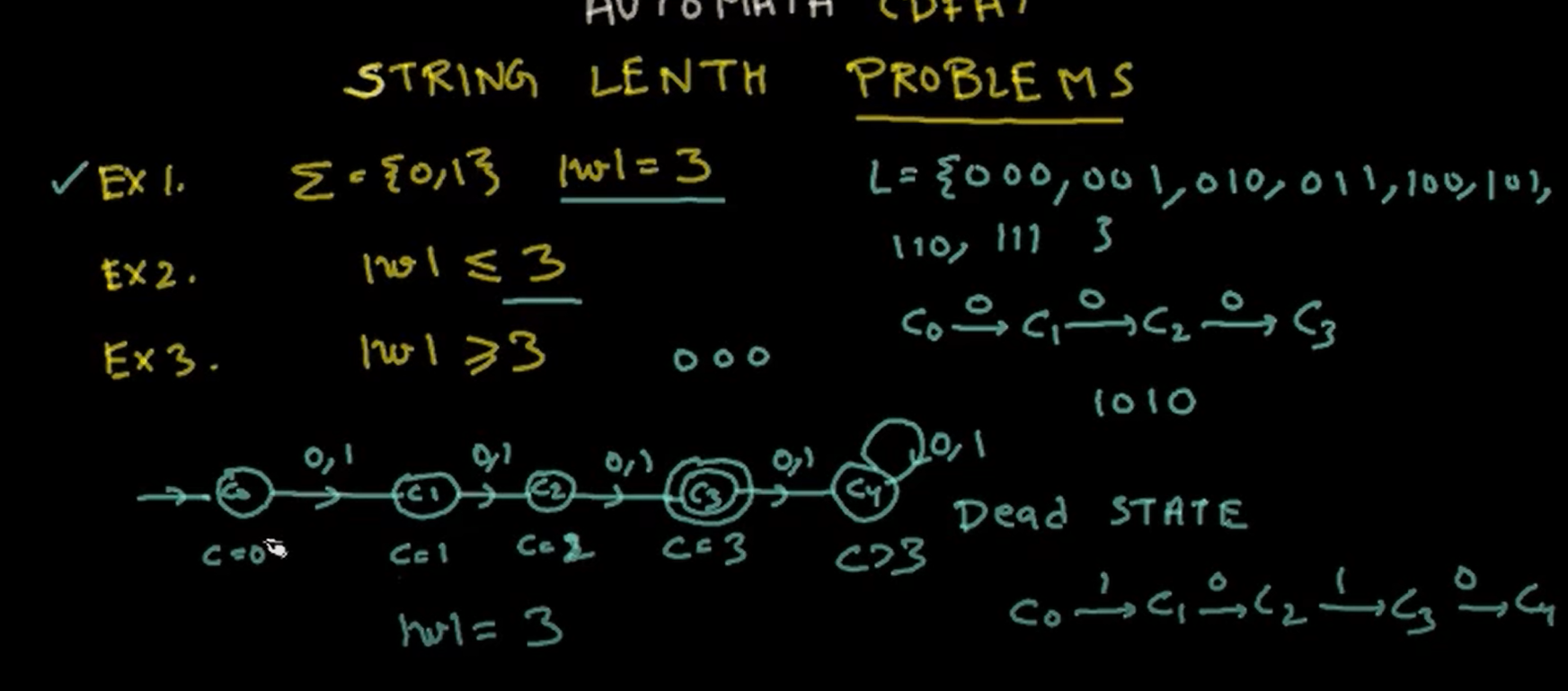
TERM DS 1

<https://www.slideshare.net/BhattBalkrishna/ch-3-assembler-in-system-programming>



<https://www.slideshare.net/ReshmaKapadi/system-programming-unit-1-introduction-86502311>

4. Design an automata for set of all strings of length 5.



5. Design an automata for identifying constants and keywords.

