

## **Unit 2**

# **Macro Processor, Linker and Loader**

topstudymaterial.com

# Syllabus

Unit II	Macro Processor, Linker and Loader	09 Hours
<p>Macro Processor: Macro instructions, Features of macro facility, Design of two-pass, single pass and nested macro processor. Loaders: Loader schemes: Compile and go, General Loader Scheme, Absolute loaders, subroutine linkages, relocating loaders, direct linking loaders, overlay structure. Design of an absolute loader, Design of direct linking loader. Linkers: Relocation and linking concepts, Design of linker, self relocating programs, Static and dynamic link libraries, use of call back functions. Case Study: Loading phases using Java.</p>		

# Macro

- Macro allows a sequence of a source language code to be defined once and the referred many times whenever it gets called.

- A macro consist of :

**Name of macro**

**Set of parameters**

**Body of macro**

**MEND**

[Parameters in macro are optional ]

- Consider an example,

-----  
-----  
ADD AREG, X  
ADD BREG, X  
-----  
-----

ADD AREG, X  
ADD BREG, X  
-----  
-----

ADD AREG, X  
ADD BREG, X  
-----

- A MACRO allows us to attach a name to this sequence and use this name in its place.

topstudymaterial.com

- A macro consist of
  - i) **Name of the macro**
  - ii) **Set of parameters**
  - iii) **Body of macro**
  - iv) **MEND**

Parameters for an macro is optional.

# Syntax

MACRO macro\_name [parameters]

//macro body

MEND

topstudymaterial.com

# Example [without parameters]

```
MACRO MYMACRO
```

```
ADD AREG, X
```

```
ADD BREG, X
```

```
MEND
```

topstudymaterial.com



# Example [with parameters]

**MACRO MYMACRO &A**

**ADD AREG, &A**

**ADD BREG, &A**

**MEND**

Formal Parameters

topstudymaterial.com

# Original Program

# Program with macro

-----  
-----  
-----  
ADD AREG, X  
ADD BREG, X  
-----  
-----

MYMACRO  
-----  
-----

ADD AREG, X  
ADD BREG, X  
-----  
-----

MYMACRO  
-----  
-----

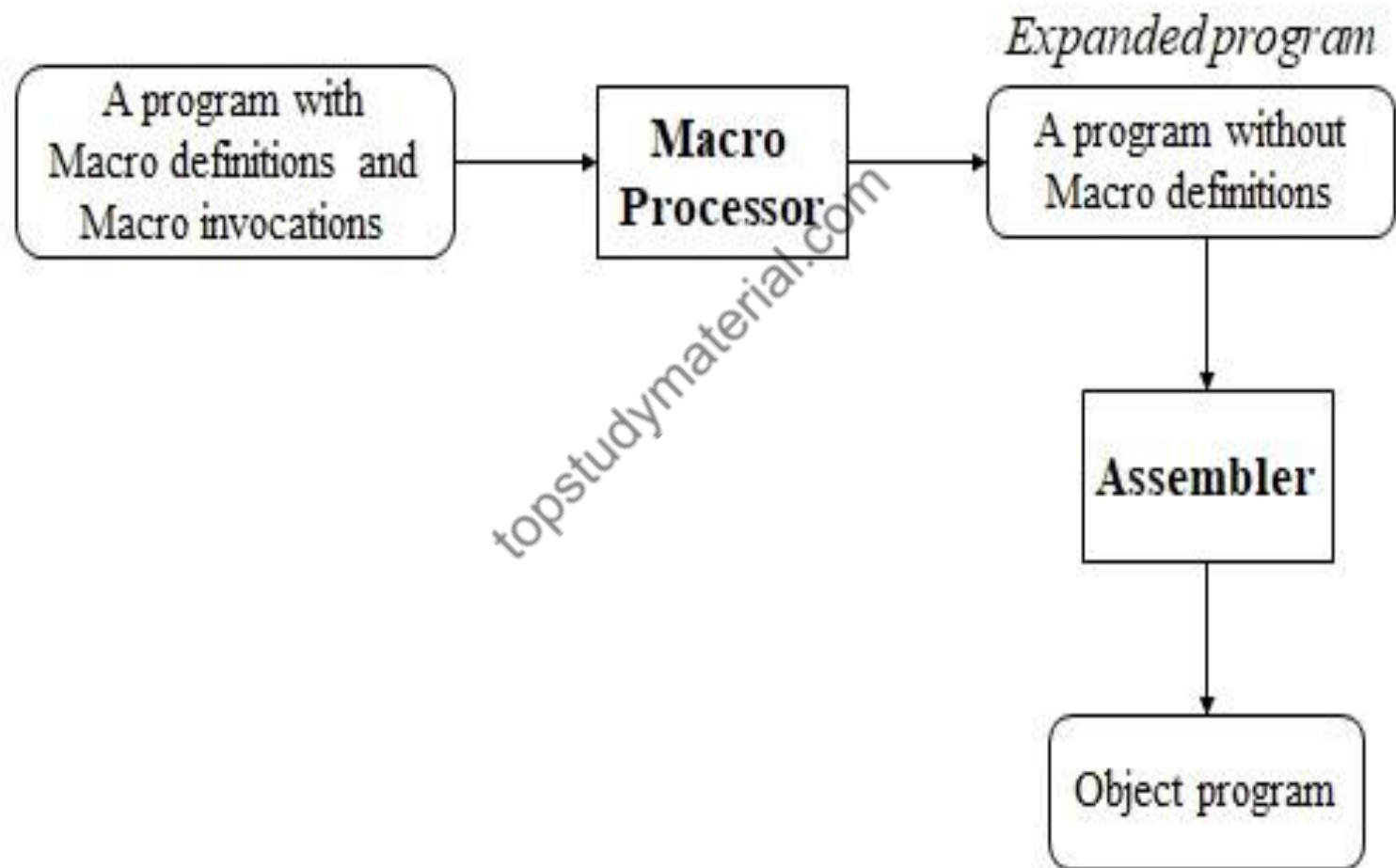
ADD AREG, X  
ADD BREG, X  
-----  
-----

MYMACRO  
-----  
-----

# Working of Macro Processor

- What will be the input for macro processor?
- What will be the output for macro processor?

topstudymaterial.com



# Working of Macro Processor

## MACRO Program

MACRO MYMACRO  
ADD AREG, X  
ADD BREG, X  
MEND

MYMACRO

MYMACRO

MYMACRO

Input

Macro  
Processor

output

## Program with macro EXAPANSION

ADD AREG, X  
ADD BREG, X

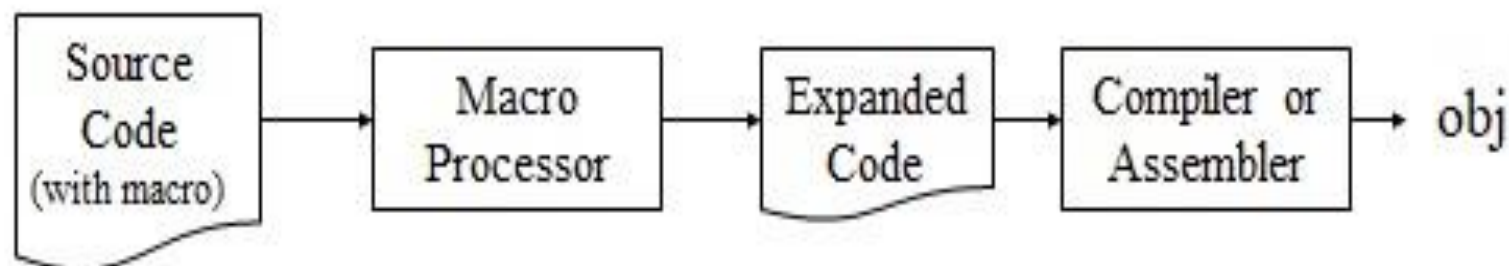
ADD AREG, X  
ADD BREG, X

ADD AREG, X  
ADD BREG, X

# Macro Processor

---

- Recognize macro definitions
- Save the macro definition
- Recognize macro calls
- Expand macro calls



# Difference between Macro and Subroutine

Parameter	Macro	Subroutine/Function
Code space requirement	More	less
Execution Speed	More	Less. Due to overhead
Processing required by the assembler	More	Less.
Flexibility and generality	Macro can not handle labels	A subroutine can handle every type of feature.
Scope	Macro is always local to the program that defines it.	Subroutine may or may not be local

# Macro

- Defining a Macro
- Calling a Macro
- Macro Expansion

topstudymaterial.com



# Types of Parameter

- Positional Parameter
- Keyword Parameter

topstudymaterial.com

For Macro Processor Pass 1 and Pass 2,  
Loader and Linker Please Download the  
handwritten notes which are given notes  
section.....

Visit the website

[www.topstudymaterial.com](http://www.topstudymaterial.com)

You can also watch video Lectures on  
loader and linker for better understanding.