(Video 1)

C → Portable [same type code run in all environment]

→ Procedural programming

[Dividing program into smaller parts]

High level vs low level

Degree of Abstraction ← Hiding more information

Less Effort More Effort

→ Mid level Language :

→ Direct access to memory through pointer

→ ~~Or~~ Bitmaniputation using bitwise operators

→ writing Assembly code within C code.

→ Popular ~~choice~~ choice for system level apps.

Kernel, Device Driver, OS,

Game, Editor

→wide variety of builtin functions, standard library, and header files.

stdio.h math.h string.h

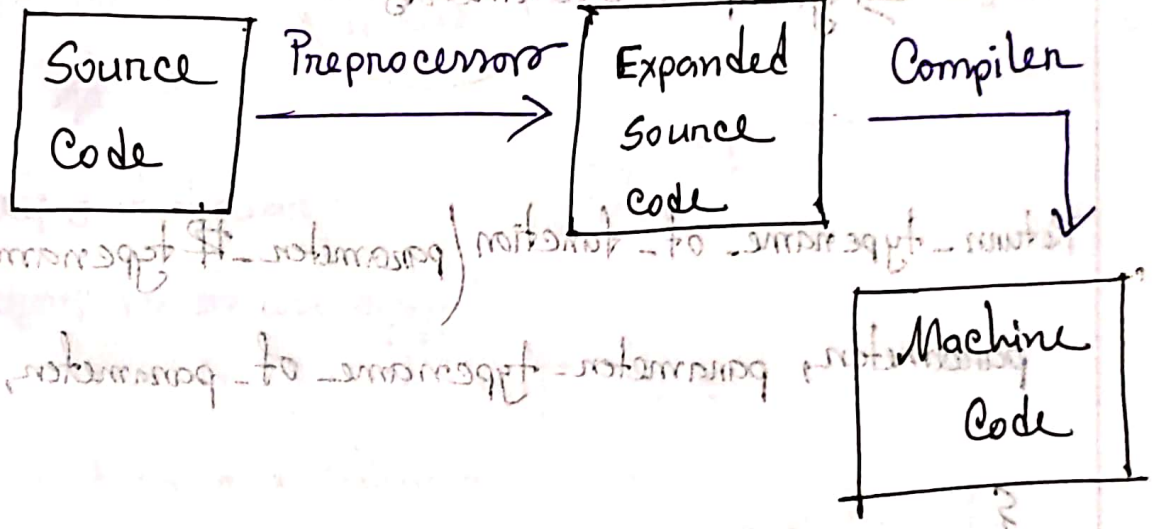
Code 1

```
#include<stdio.h>
int main()
{
    printf("Hellow World");
    return 0;
}
```

```
#include<stdio.h>
```

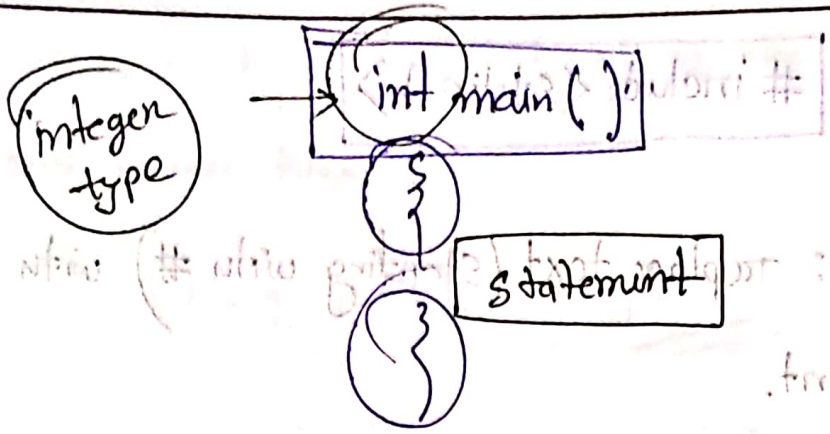
Preprocessor: replace text (starting with #) with the actual content.

↳ output of preprocessor is expanded source code



stdio → standard input output

.h → header file → contain printf, scanf etc



main function

↳ group of statements

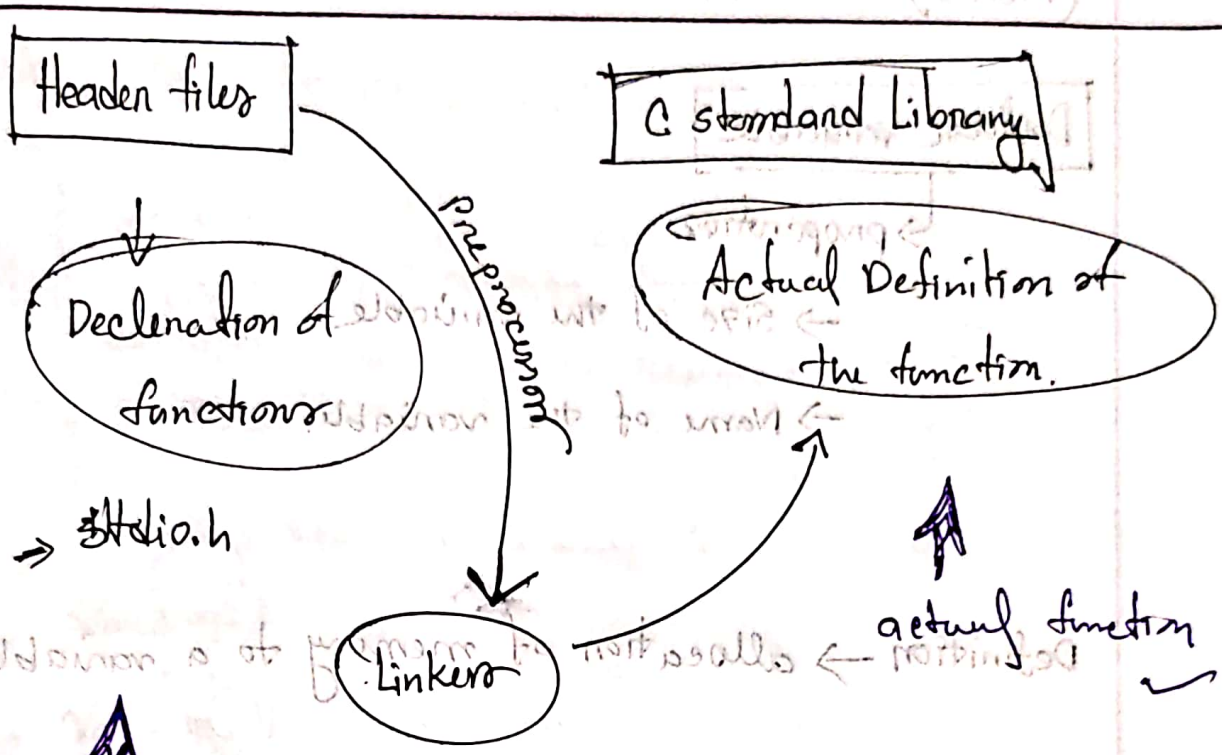
Return_typename_of_function (parameter_typename_of parameter, parameter_typename_of_parameter, --)

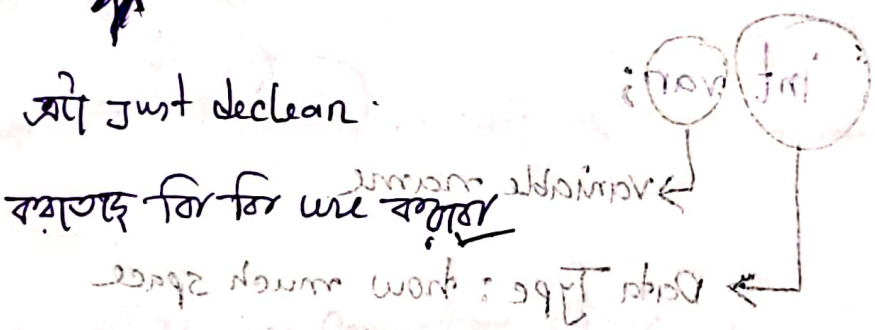
{

Set of statements;

}

Syntax of function


Header files
C standard Library
Preprocessor
Declaration of functions
Actual Definition of the function.
⇒ stdio.h
Linker
actual function


যেটা just declean
int