SWT 11012 - Fundamentals of Programming

Structure of a simple program

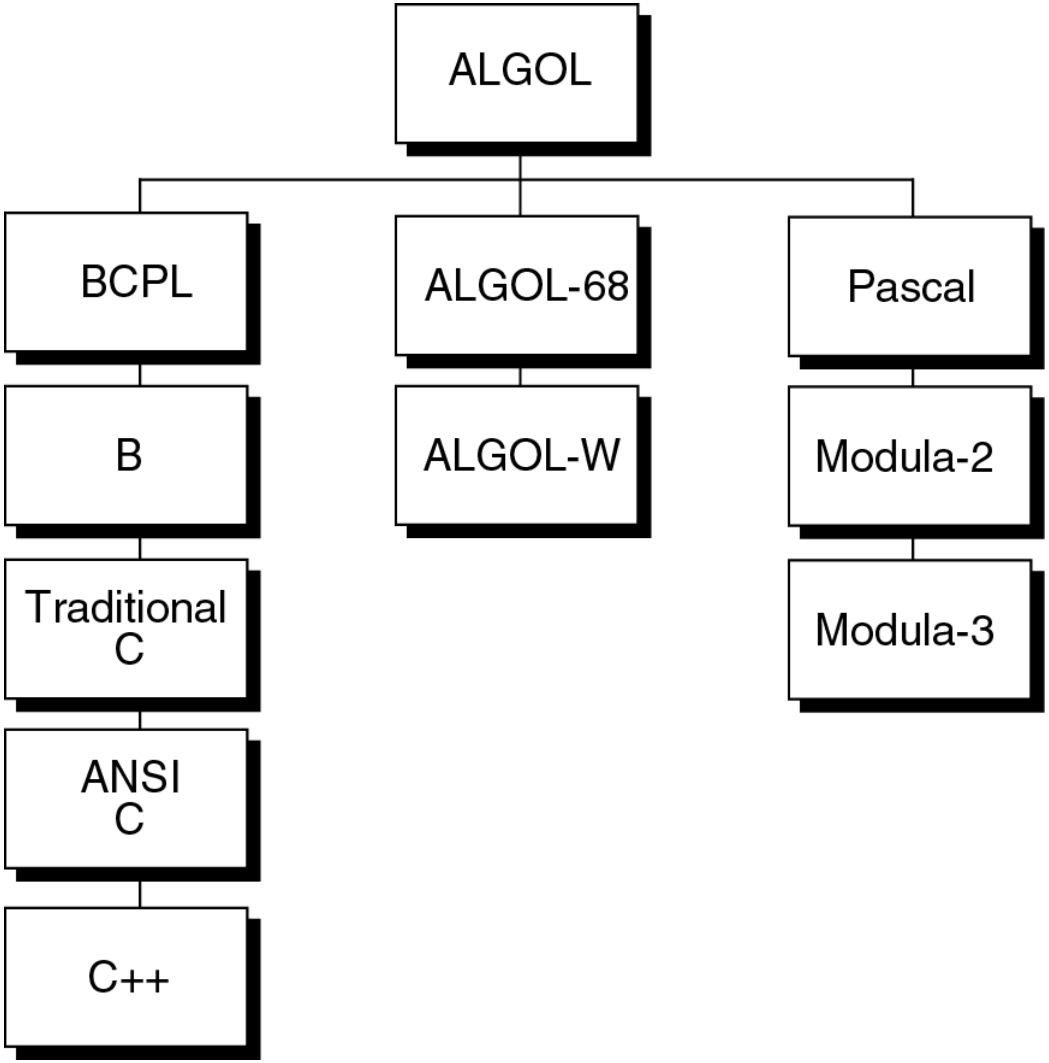
SL. Abdul Haleem
MSc in IT, PGD in IT, B.Sc .Sp. (Hons) in Computer .Sc., NDRT(UniVoTec)., MCS(SL)., MSLAAS , MCSTA(ACM)
Senior Lecturer / ICT
Department of Information & Communication Technology
Faculty of Technology
South Eastern University of Sri Lanka

2025

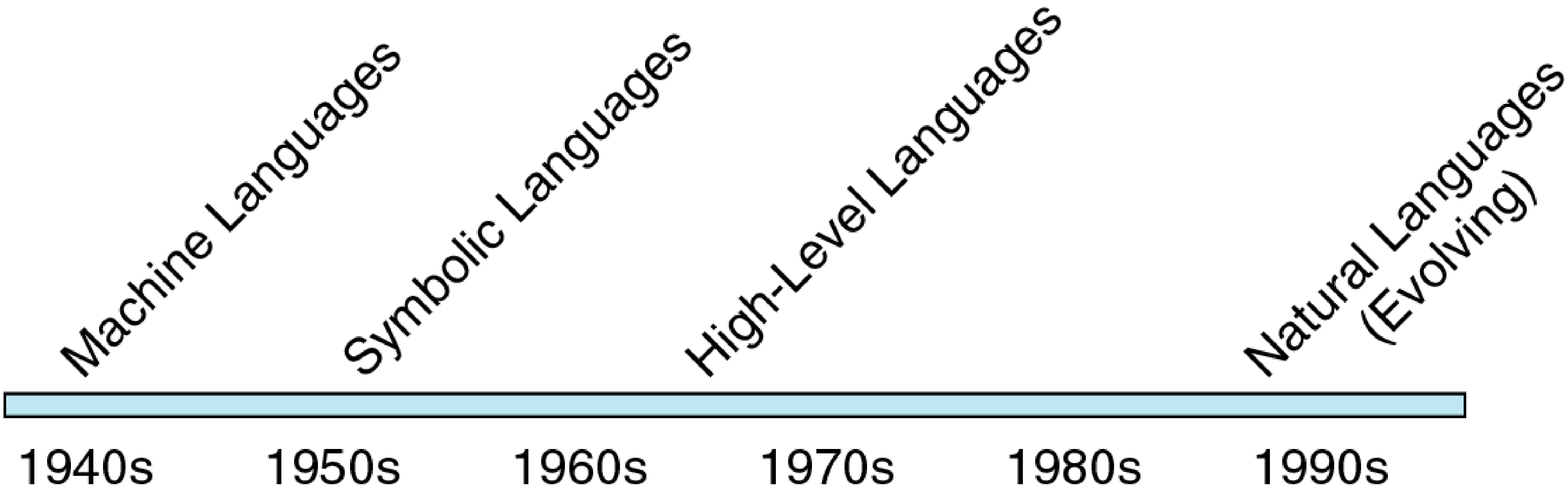
Lecture Session Two Outline

- History of Programing languages
- Program Structure using C
- Elements of Programing Language
  - Preprocessor directives
  - Function (Header and body)
  - Executable statements
- Words
  - Reserved word, Standard identifiers, user defined identifiers

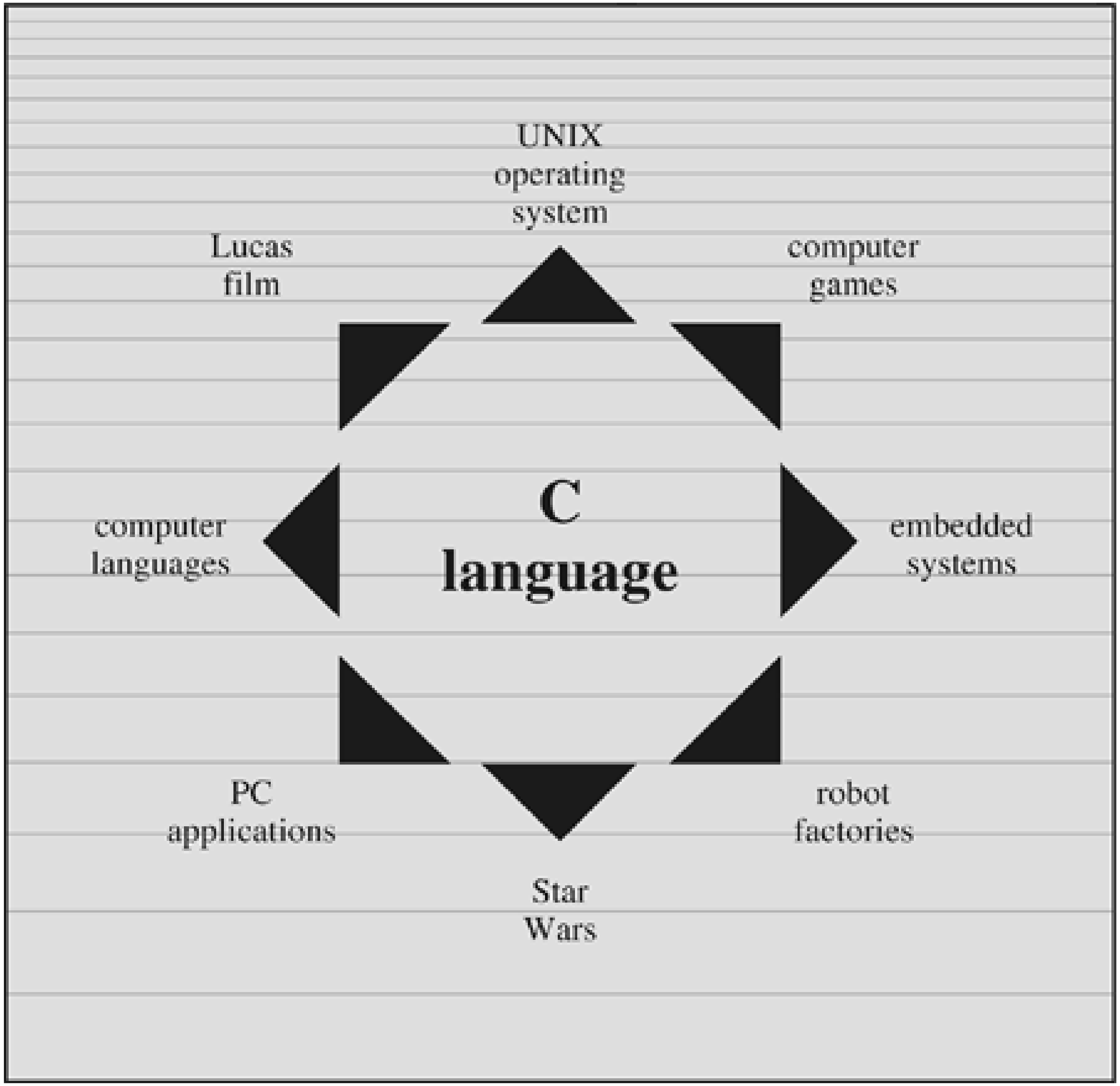
History of Programing languages



History of Computer Language



History of Computer Language



UNIX operating system

Lucas film

computer games

computer languages

C language

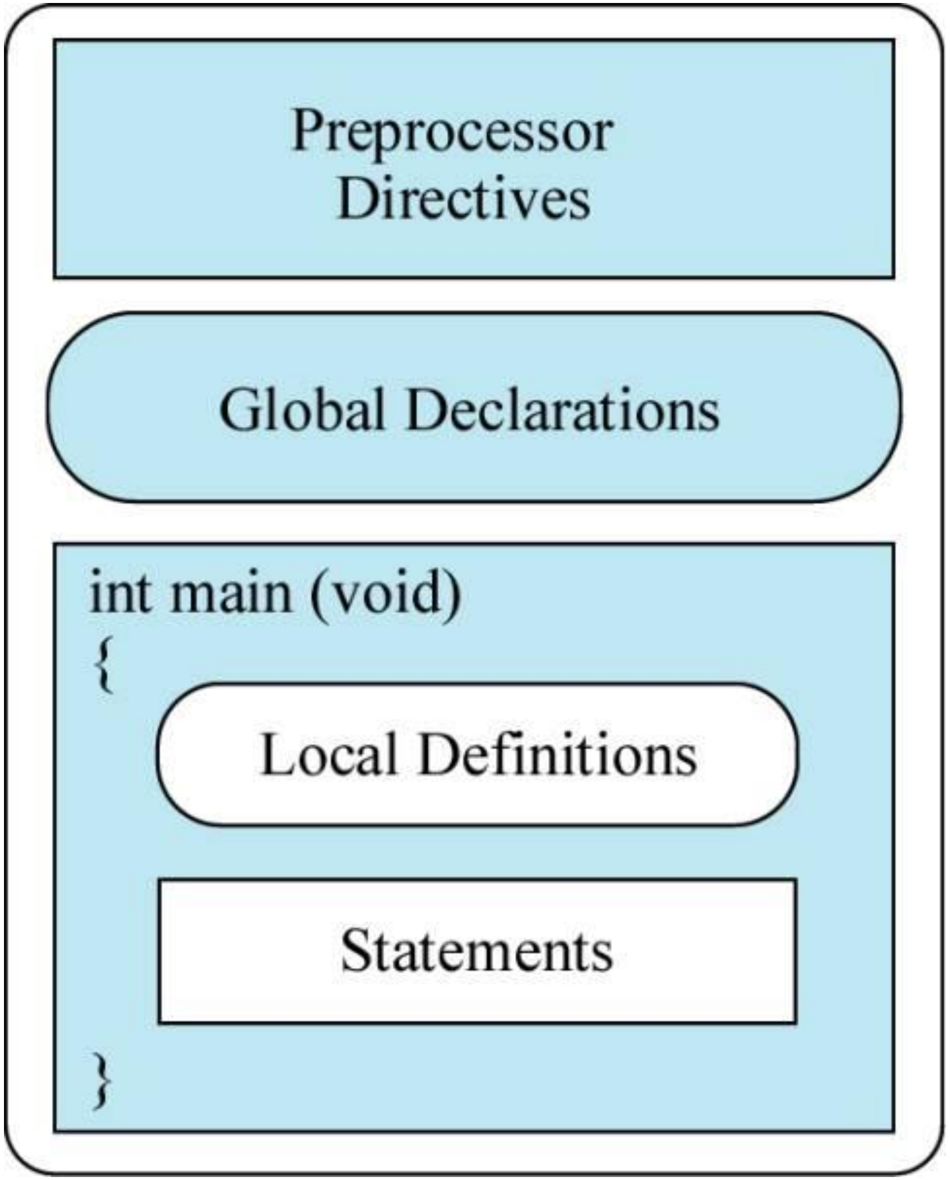
embedded systems

PC applications

robot factories

Star Wars

C Program Structure

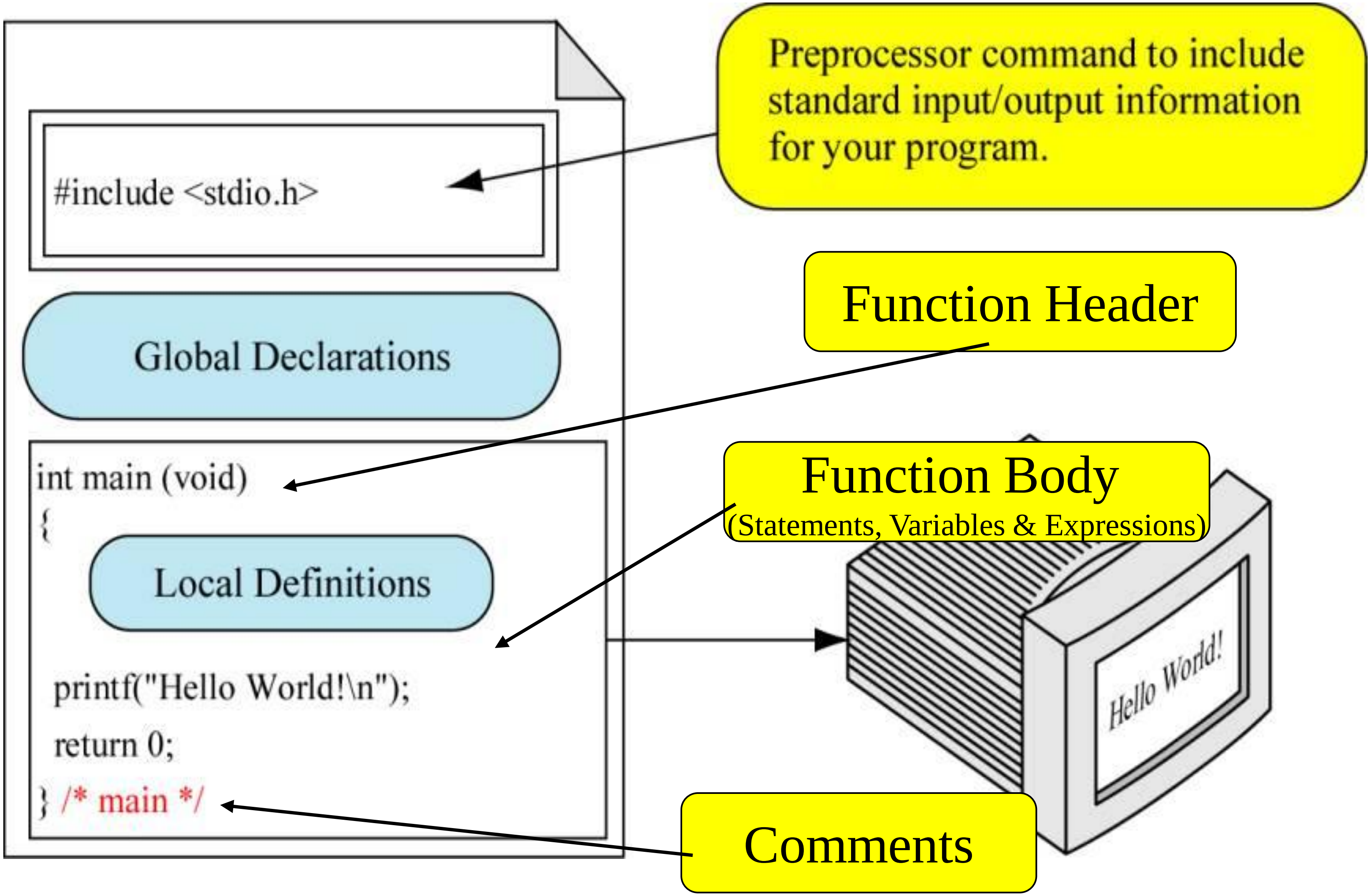


Simple C Program structure

A C program basically consists of the following parts:

- Preprocessor directives
- Functions
- Variables
- Statements & Expressions
- Comments

Simple C Program structure



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

Global Declarations

```
int main (void)
{
```

Local Definitions

```
 printf("Hello World!\n");
 return 0;
} /* main */
```

Preprocessor command to include standard input/output information for your program.

Function Header

Function Body
(Statements, Variables & Expressions)

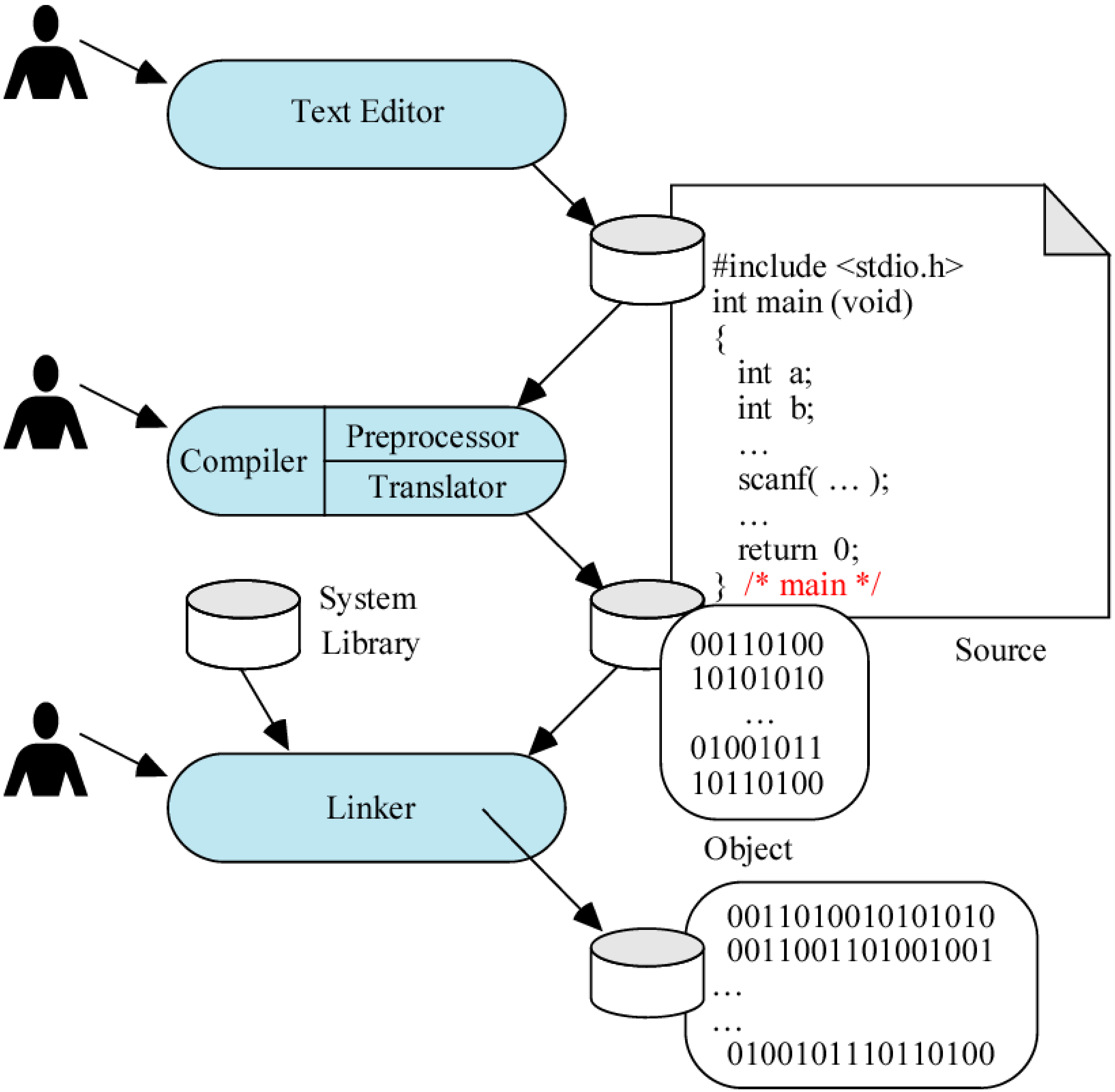
Comments

Simple C Programs

- Write a c program to print an “Welcome to C programing world”.
- **Program 01**

```
#include <stdio.h>
int main()
   {
      printf("Welcome to C programing world");
      return 0;
   /*This is printing the Welcome to C Programing world on
   the screen*/
   }
```

Programing .....

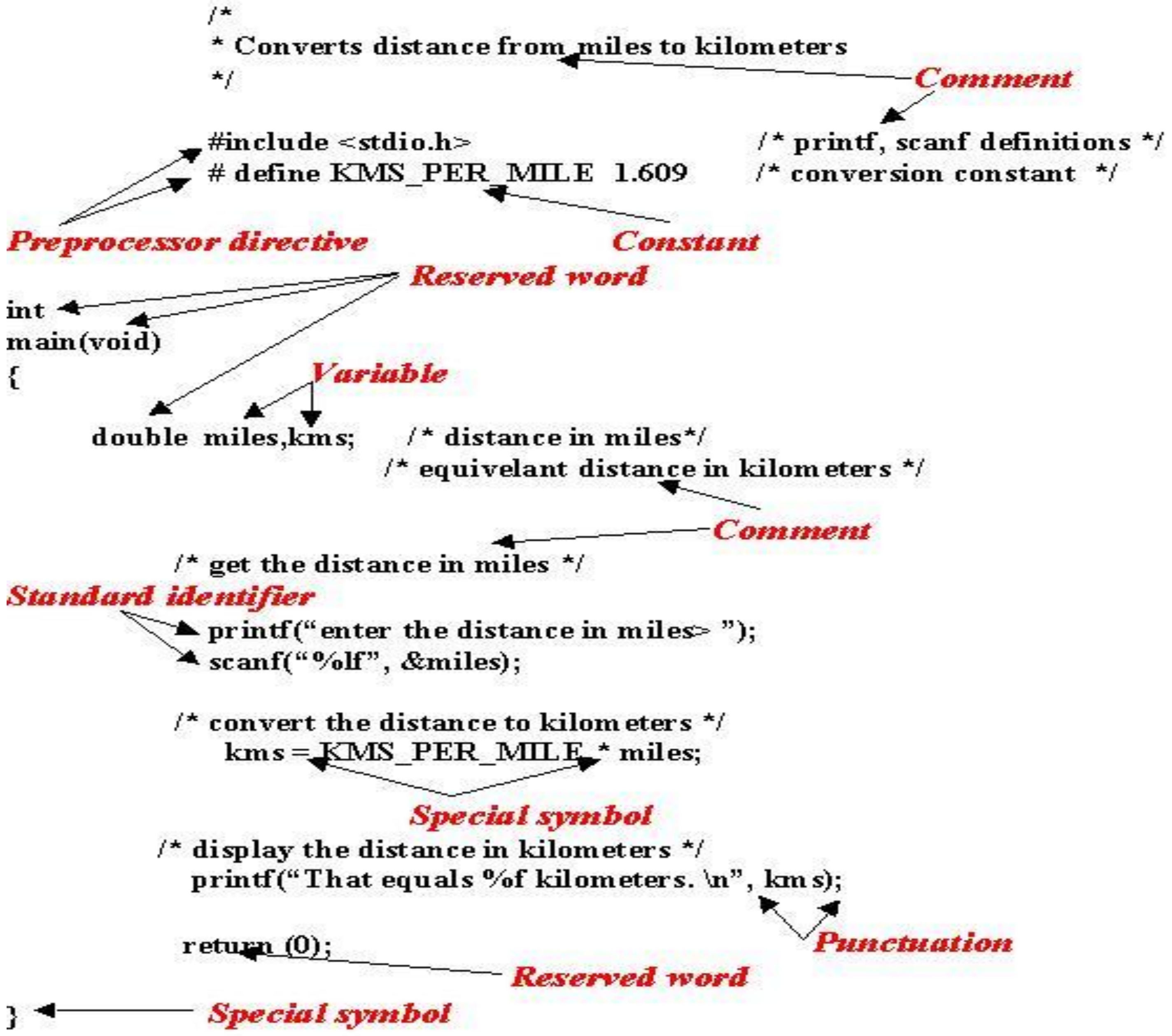


```
#include <stdio.h>
int main (void)
{
   int  a;
   int  b;
   …
   scanf( … );
   …
   return  0;
}  /* main */
```

```
00110100
10101010
   …
01001011
10110100
```

```
0011010010101010
0011001101001001
…
…
0100101110110100
```

A simple second C program



```
/*
 * Converts distance from miles to kilometers
 */

#include <stdio.h>                    /* printf, scanf definitions */
# define KMS_PER_MILE  1.609          /* conversion constant   */

int
main(void)
{
      double  miles,kms;      /* distance in miles*/
                            /* equivelant distance in kilometers */

        /* get the distance in miles */
        printf(“enter the distance in miles> ”);
        scanf(“%lf”, &miles);

        /* convert the distance to kilometers */
          kms = KMS_PER_MILE * miles;

       /* display the distance in kilometers */
         printf(“That equals %f kilometers. \n”, kms);

        return (0);
}
```

Elements of C Programing Languages :
Preprocessor Directives

- Are commands that give instructions to the C preprocessor
- Begins with a (#) as its nonblank character.
- C preprocessor is a system program that modifies a C program prior to its compilation

Example

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

- The C language cannot do I/O by itself, so we need help from the library "stdio.h" to use the screen/Keyboard !
- We can use other libraries, too , as needed.
- Another popular "library" is math.h , for advanced math functions.

Elements of C Programing Languages :

Preprocessor Directives

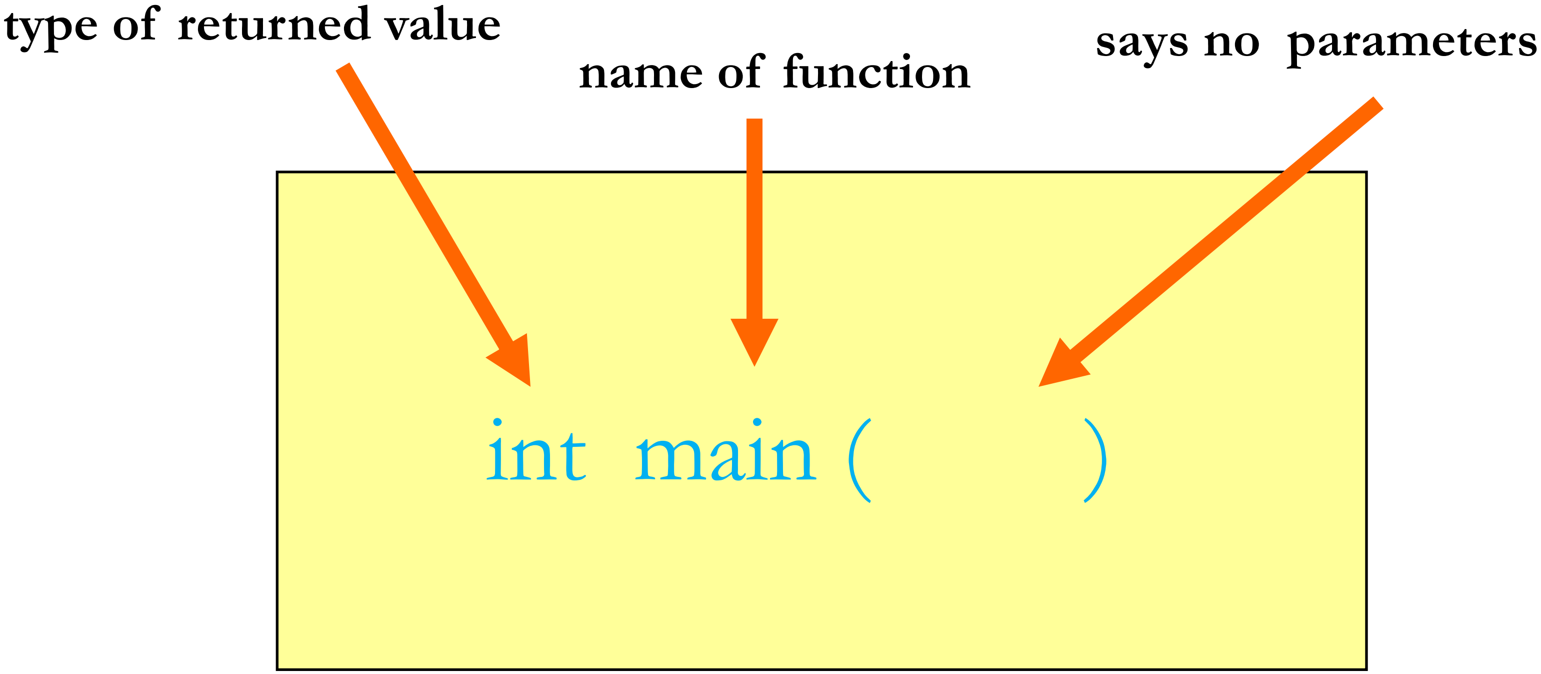
Examples

```
#define KMS_PER_MILE  1.609
# define AT  '@'
# define VOTING_AGE   18
```

- **#define** is a preprocessor directives. It is used for valid constant declarations.
- A named constant is a location in the memory that we can refer to by a name, and in which a data value that cannot be changed is stored.
- This directives instructs the processor to replace each occurrence of KMS_PER_MILE in the statements of the C program by 1.609 before compilation begins.

Elements of C Programing Languages :

The Function Header



type of returned value

name of function

says no parameters

```
int  main (          )
```

Elements of C Programing Languages :
The Function Body

- A C program is a **collection** of **one** or **more** “functions” (parts)
- There must be a function called **main( )**
- A function body has two parts **declaration** and **executable statements**.
- Execution always begins with the first statement in function **main( )**
- Any other functions in the program are **subprograms** and are not executed until they are called.

Elements of C Programing Languages :
The Executable Statements

- **printf(“ Hello, World”);**

  This statement means: Display the words **Hello, World** on the screen

- **return (0);**

  This statement **STOPs** the program and returns **Zero** to the O.S. (more about this later)

Words

- **Reserved Word**
  - A word that has special meaning in C
- **Standard Identifiers**
  - A word having special meaning but one that a programmer may redefine (but redefinition is not recommended)
- **User Defined Identifiers**
  - An **identifier** must **consist only** of letters, digits, and underscores.
  - An identifier cannot **begin** with a **digit**
  - A C reserved word cannot be used as an **identifier.**
  - An identifier defined in a C standard library should not be redefined.
  - C is a **case-sensitive** language. The names **Pressure**, **pressure**, and **PRESSURE** are viewed by the compiler as different identifier.