CPE101 Programming Languages I

Week 7
Introduction to C Programming Language and C Compiler

Assoc. Prof. Dr. Caner ÖZCAN

History of the C Programming Language

- C programming language was developed in 1972 at Bell Labs by Dennis Ritchie (Dennis Ritchie has also worked on the UNIX operating system and B programming language with Ken Thompson before).
- Although the development history of 1972, it has been the widespread later "C Programming Language" book which is Brian W. Kernighan and Dennis M. Ritchie published in 1978.

History of the C Programming Language

- ►It is used in all kinds of transactions such as operating systems, embedded systems, software and drivers which is required speed.

- ►After developed several different versions on 70s and 80s, it has been defined by the ANSI standard C language (ANSI C) in 1989.

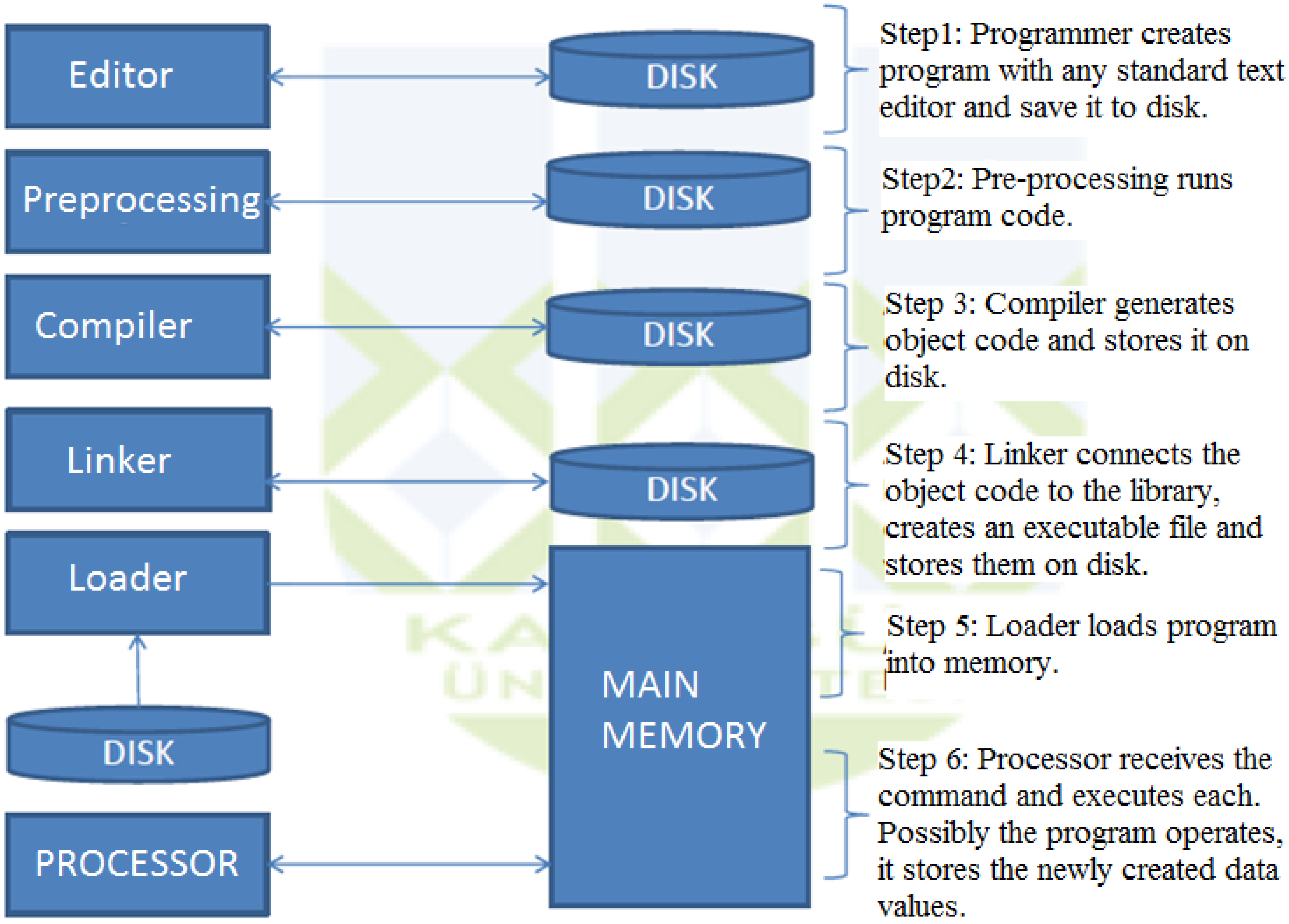
History of the C Programming Language

- A complete and independent from the machine C define was accepted as a standard in 1989.
- This standard was updated in 1999.
- C99, is an updated version of the C language standard.
- Some of the popular C Compiler does not support C99.

C Standard Library

- C programs consist of modules called functions.
- A programmer can define their own functions.
  - Advantage: Programmer knows exactly how to operate the function.
  - Disadvantages: a waste of time
- Programmers often use the C library functions.
  - These functions are used to build the program.
- Do not try to rediscover America.
  - If you have already created functions, it is best to use existing instead of typing again.
  - Library functions can be used very carefully and in a manner consistent, in accordance with written requirements.

A Typical C Program Development Environment



Step1: Programmer creates program with any standard text editor and save it to disk.

Step2: Pre-processing runs program code.

Step 3: Compiler generates object code and stores it on disk.

Step 4: Linker connects the object code to the library, creates an executable file and stores them on disk.

Step 5: Loader loads program into memory.

Step 6: Processor receives the command and executes each. Possibly the program operates, it stores the newly created data values.

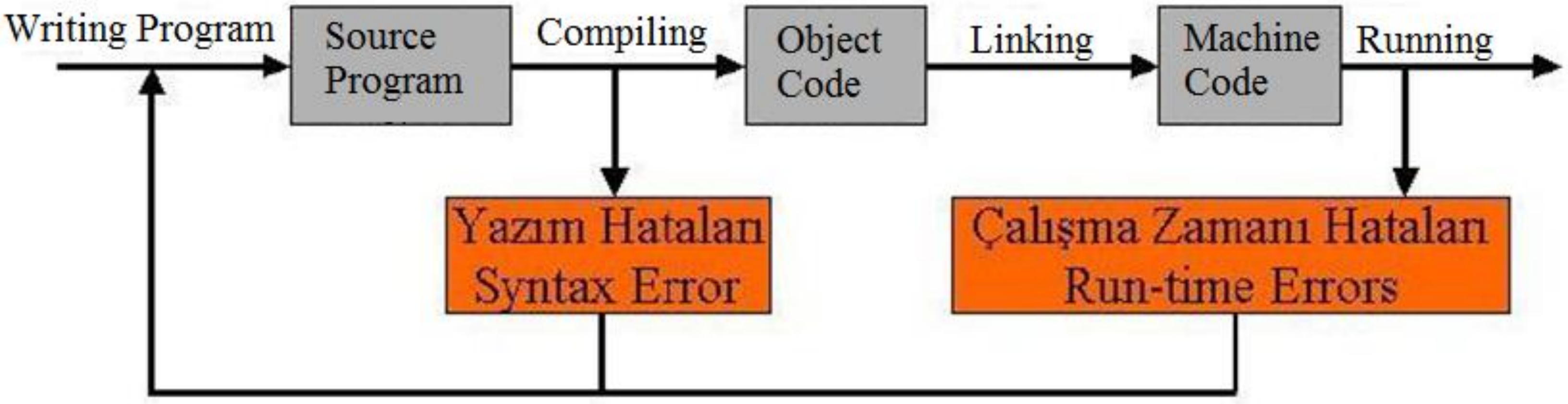
A Typical C Program Development Environment

Programs may not always work the first time.

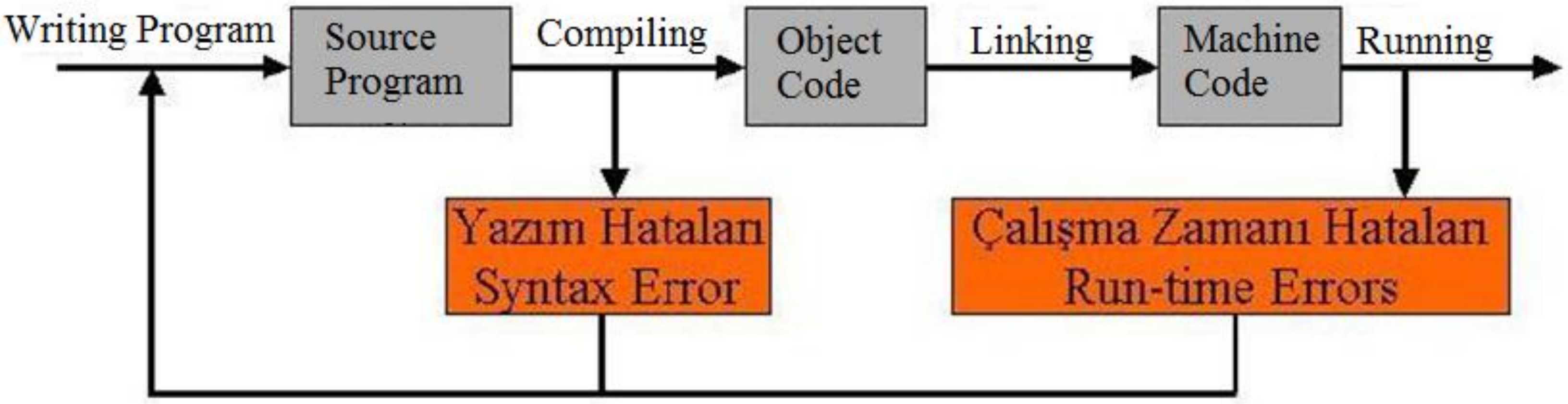
- For example, it can run the program tries to perform division by zero.
  - Errors occur during the operation of the program, such as division by zero, so these errors called as run-time errors.
- Most C program makes data input / output operation.
  - standard input stream(stdin)
  - standard outputstream(stdout)
  - standard error stream (stderr)

Introduction to C Programming Language

Provision of the life cycle: Given flowchart is a loop. If in need of some changes, the steps are re-examined. This chart is not included software's maintenance cycle.



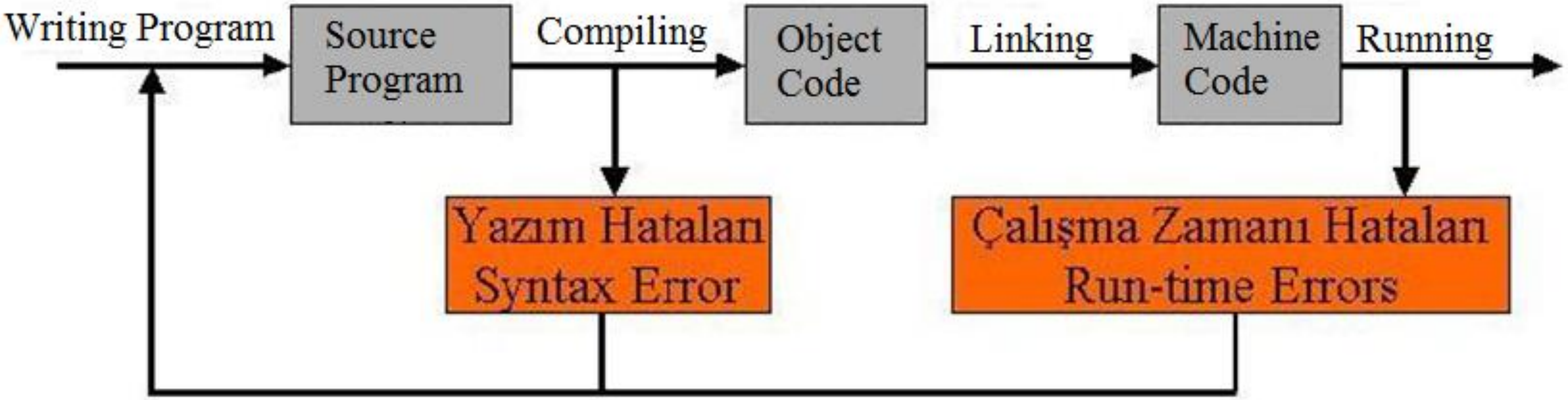
Introduction to C Programming Language



Error detection and debugging: Stretches indicated in red-orange color indicates the error condition.

Syntax Error: Occurs when the programming language rules are violated. Correction is extremely simple. Some compilers will report errors and how to resolve it.

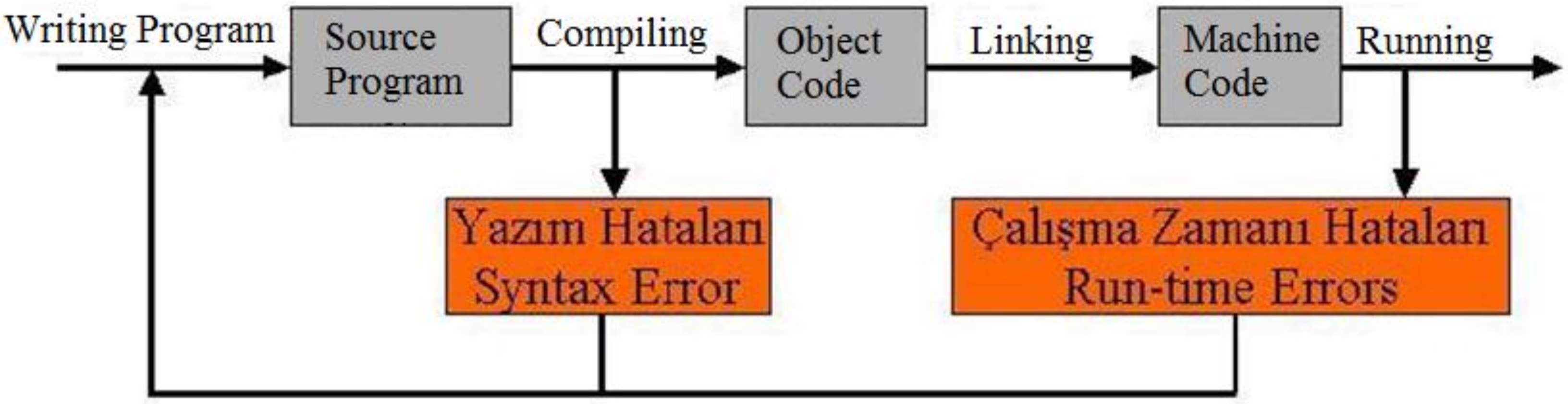
Introduction to C Programming Language



Run-time Error: Errors are encountered during the execution of the program. The program is interrupted by the operating system. Error message is given by the language of the working operating system.

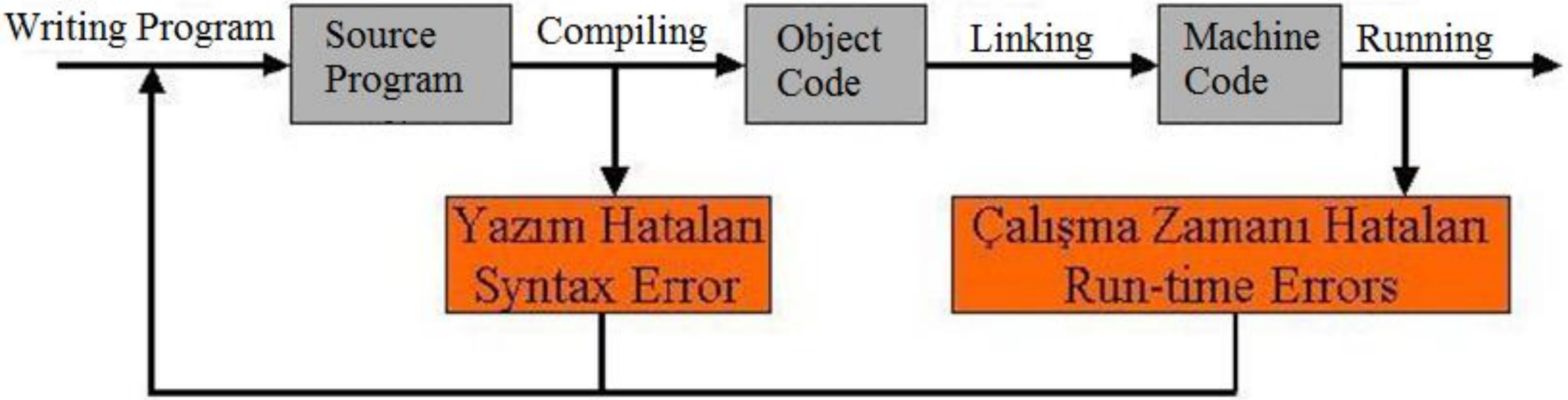
For example, try to open non-existing file, try to reach a non-existing hardware, attempt to allocate memory from a non-existing memory source.

Introduction to C Programming Language



Logical Error: Most dangerous bugs you may encounter. Some things are caused by improper consideration in programming logic. The error occurs in the test phase. Should return to the design or analysis stage to solve the problem. Sometimes, it is very difficult to find the source of the error.

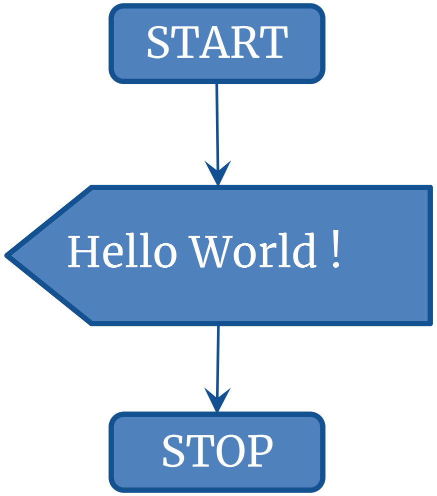
Introduction to C Programming Language



Bug: is the name given to a logical error. A program carried out for the first time, fully expecting it to be extremely accurate incorrect.

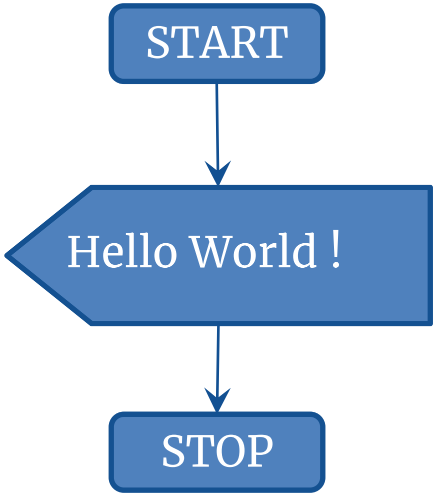
Debug: is the name of the operation to find bugs and to resolve logical errors. Running the code step by step under control. It provides an analysis of the program.

First Program in C (Hello World)

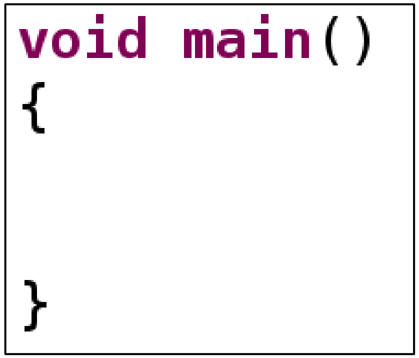


First Program in C (Hello World)

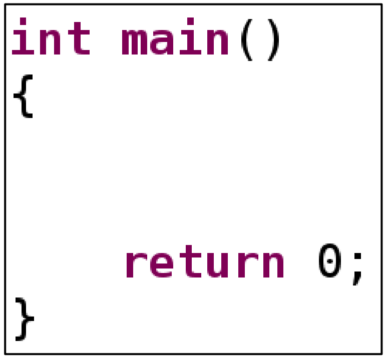
```c
#include <stdio.h>

void main()
{
    printf("Merhaba Dunya!");
}
```



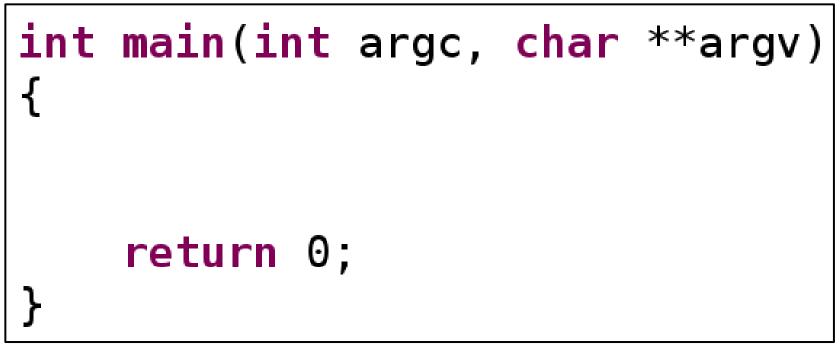
Main Functions Writing Types in C Language



```c
void main()
{


}
```



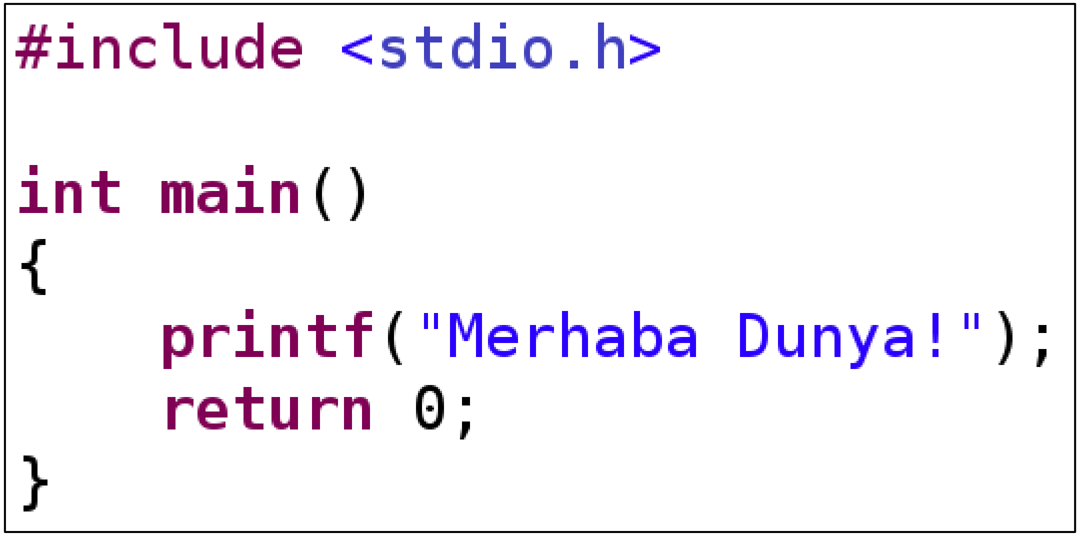
```c
int main()
{


    return 0;
}
```



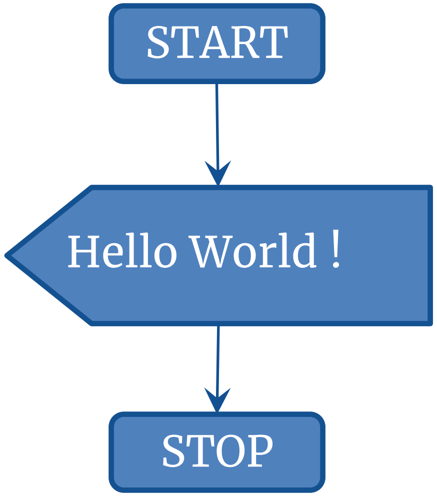
```c
int main(int argc, char **argv)
{


    return 0;
}
```

- You can use any of the above forms of writing your program.

First Program in C (Hello World)



```
#include <stdio.h>

int main()
{
    printf("Merhaba Dunya!");
    return 0;
}
```



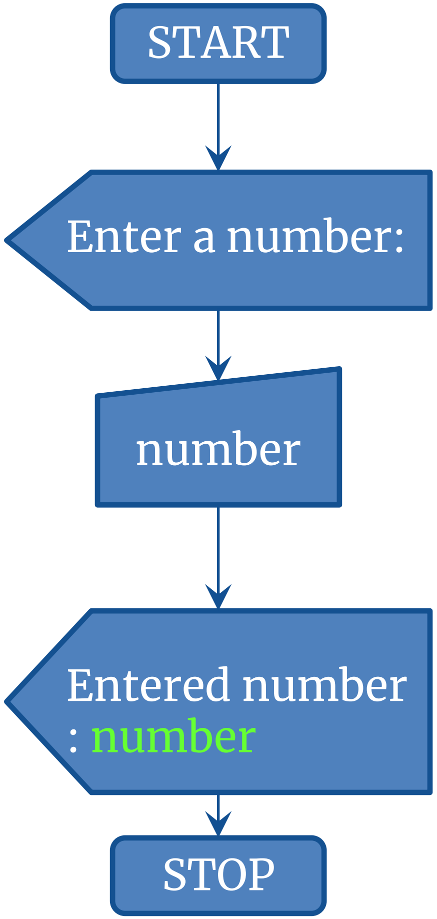
START

Hello World !

STOP

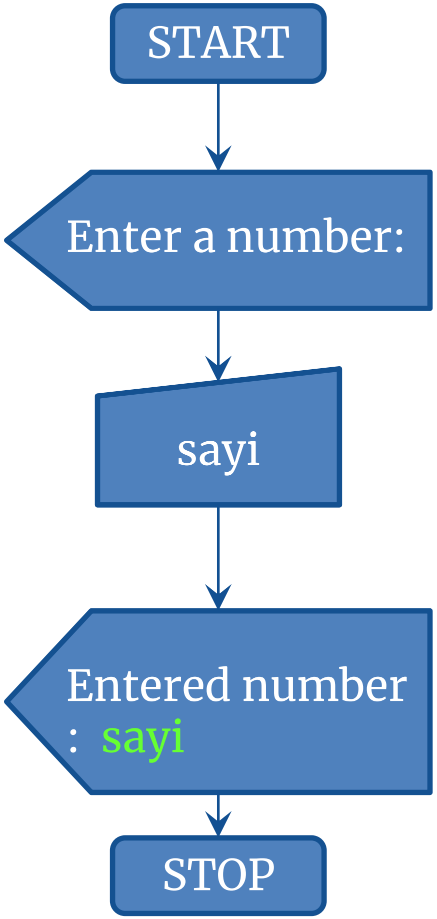
(Previous example, differently written form)

Example: Reading and Writing Number

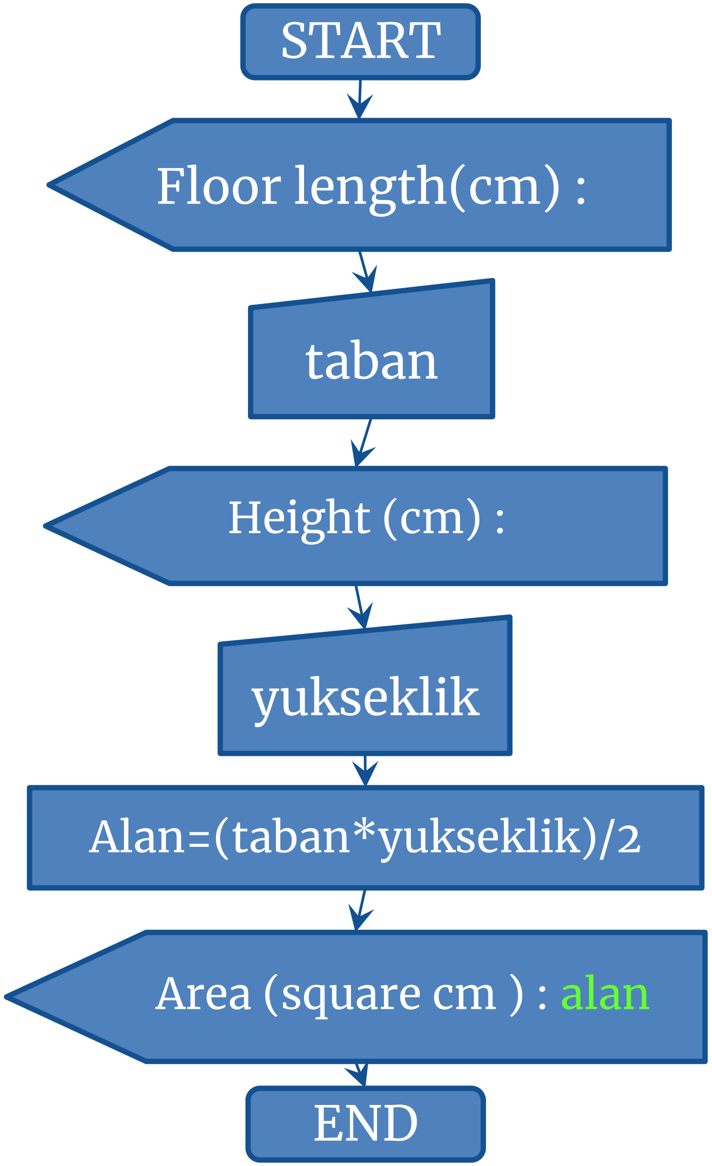


Example: Reading and Writing Number

```c
#include <stdio.h>

void main()
{
    int sayi;

    printf("Bir sayi giriniz :");

    scanf("%d",&sayi);

    printf("Girilen sayi : %d\n",sayi);
}
```

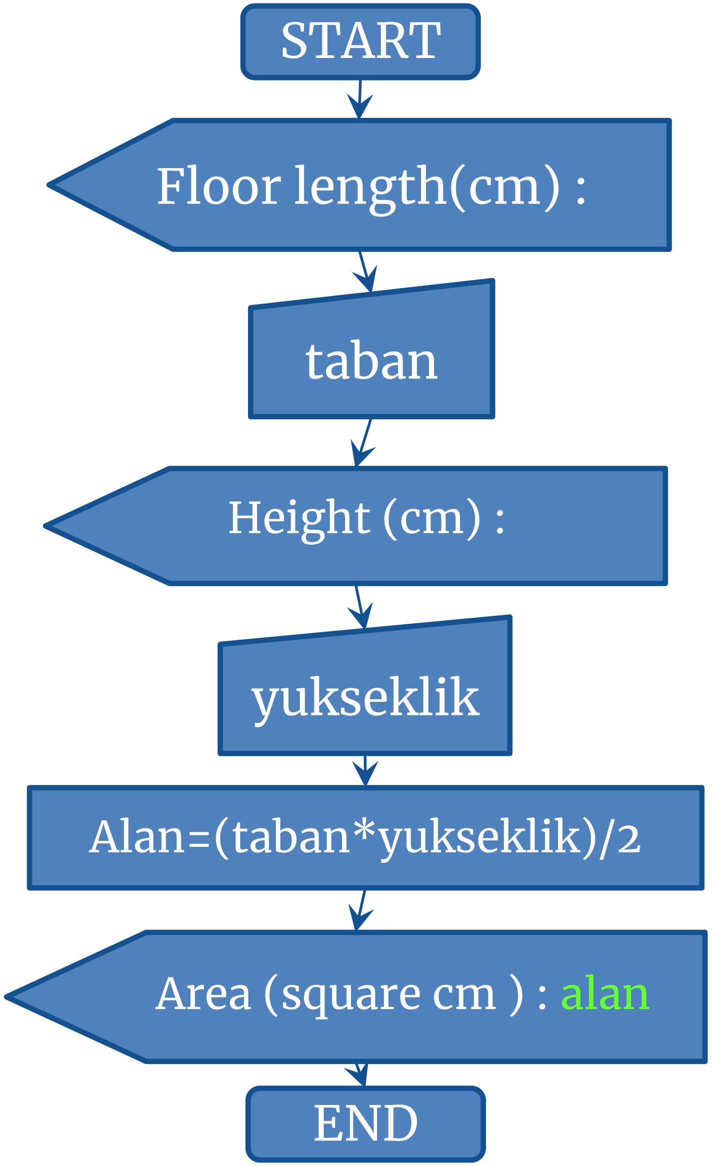


Example: Calculation Triangle Area



Example: Calculation Triangle Area

```c
#include <stdio.h>

void main()
{
    int taban,yukseklik,alan;

    printf("Taban uzunlugu(cm) :");
    scanf("%d",&taban);

    printf("Yukseklik(cm) :");
    scanf("%d",&yukseklik);

    alan=(taban*yukseklik)/2;

    printf("alan (cm kare) :%d",alan);
}
```



General Writing Rules of the C Language

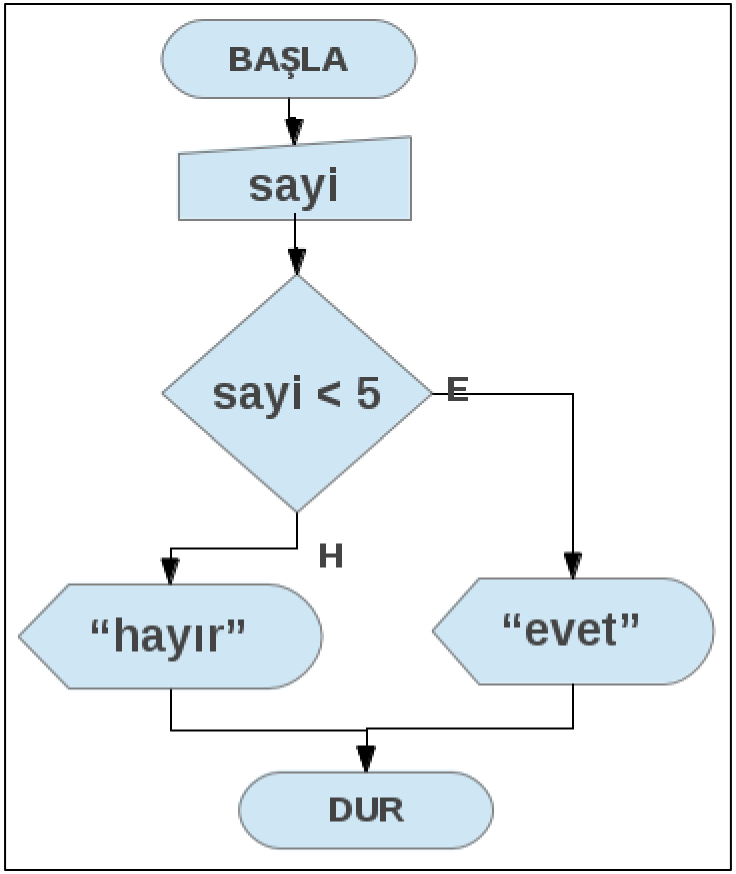
- Single and Multi-Line Comments

- // When used to do a single line comment.

- /* Multiple comment line
  * The method we use when we can not
  * explain with one line.
  */

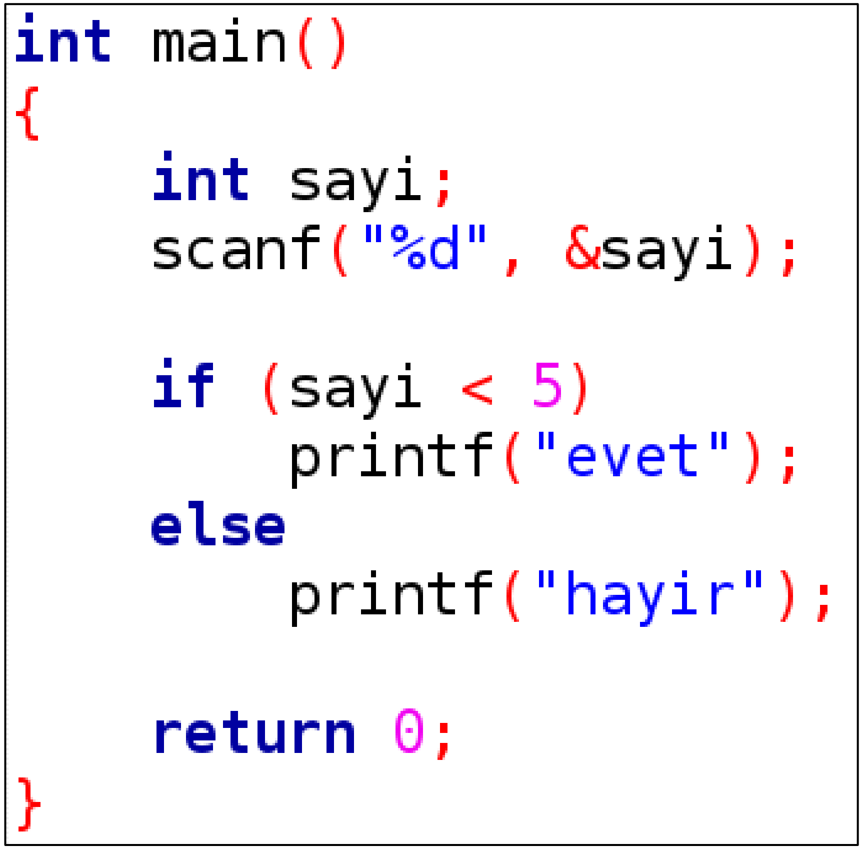
Example

- If the entered number is smaller than 5 write "yes" else write "no" with the program.



Example

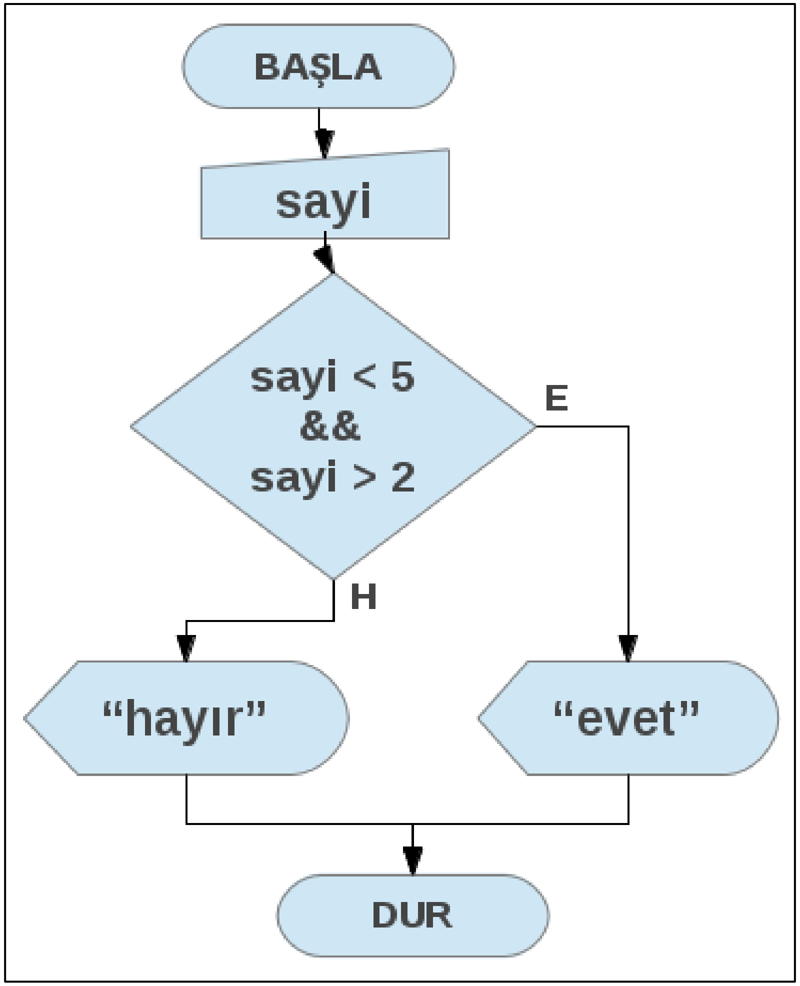
- If the entered number is smaller than 5 write "yes" else write "no" with the program.



```c
int main()
{
    int sayi;
    scanf("%d", &sayi);

    if (sayi < 5)
        printf("evet");
    else
        printf("hayir");

    return 0;
}
```

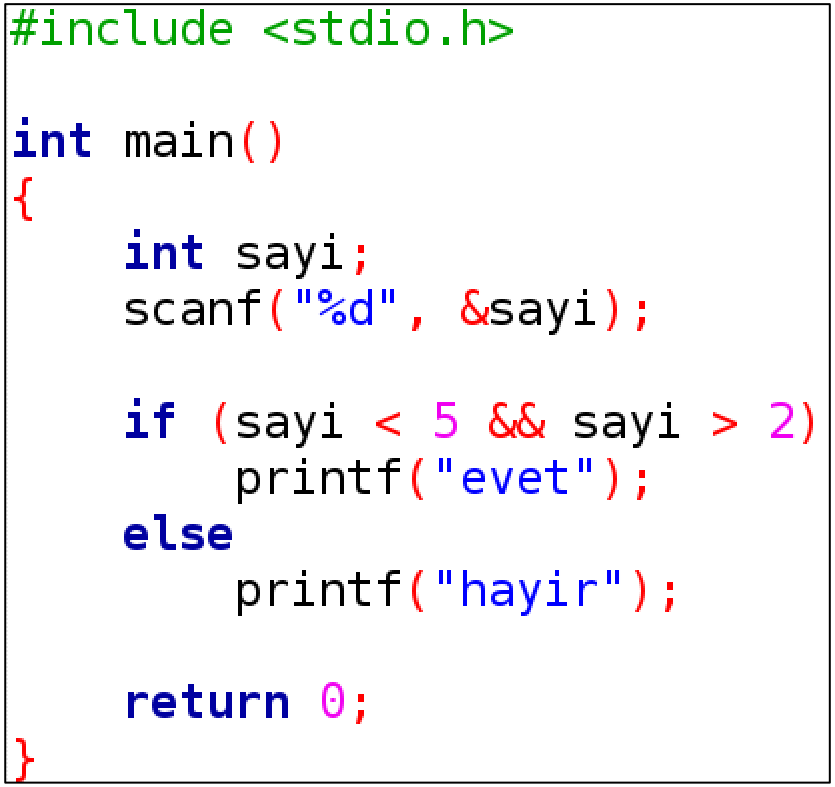
Example

- If the entered number is greater than 2 and smaller than 5 write "yes" else write "no" with the program.



Example

- If the entered number is greater than 2 and smaller than 5 write "yes" else write "no" with the program.



```c
#include <stdio.h>

int main()
{
    int sayi;
    scanf("%d", &sayi);

    if (sayi < 5 && sayi > 2)
        printf("evet");
    else
        printf("hayir");

    return 0;
}
```

References

- Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- J. G. Brookshear, “Computer Science: An Overview 10th Ed.”, Addison Wisley, 2009.
- Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- Paul J. Deitel, “C How to Program”, Harvey Deitel.
- Bayram AKGÜL, C Programlama Ders notları