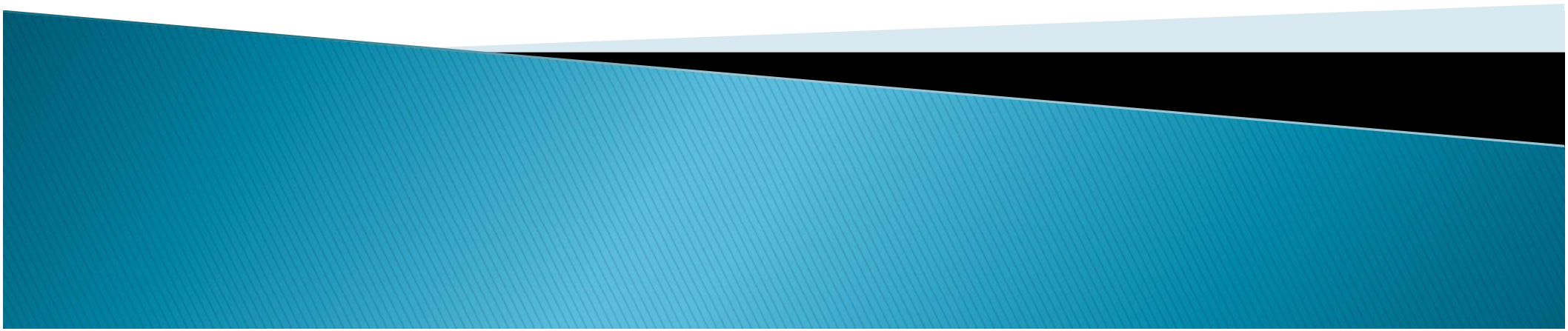
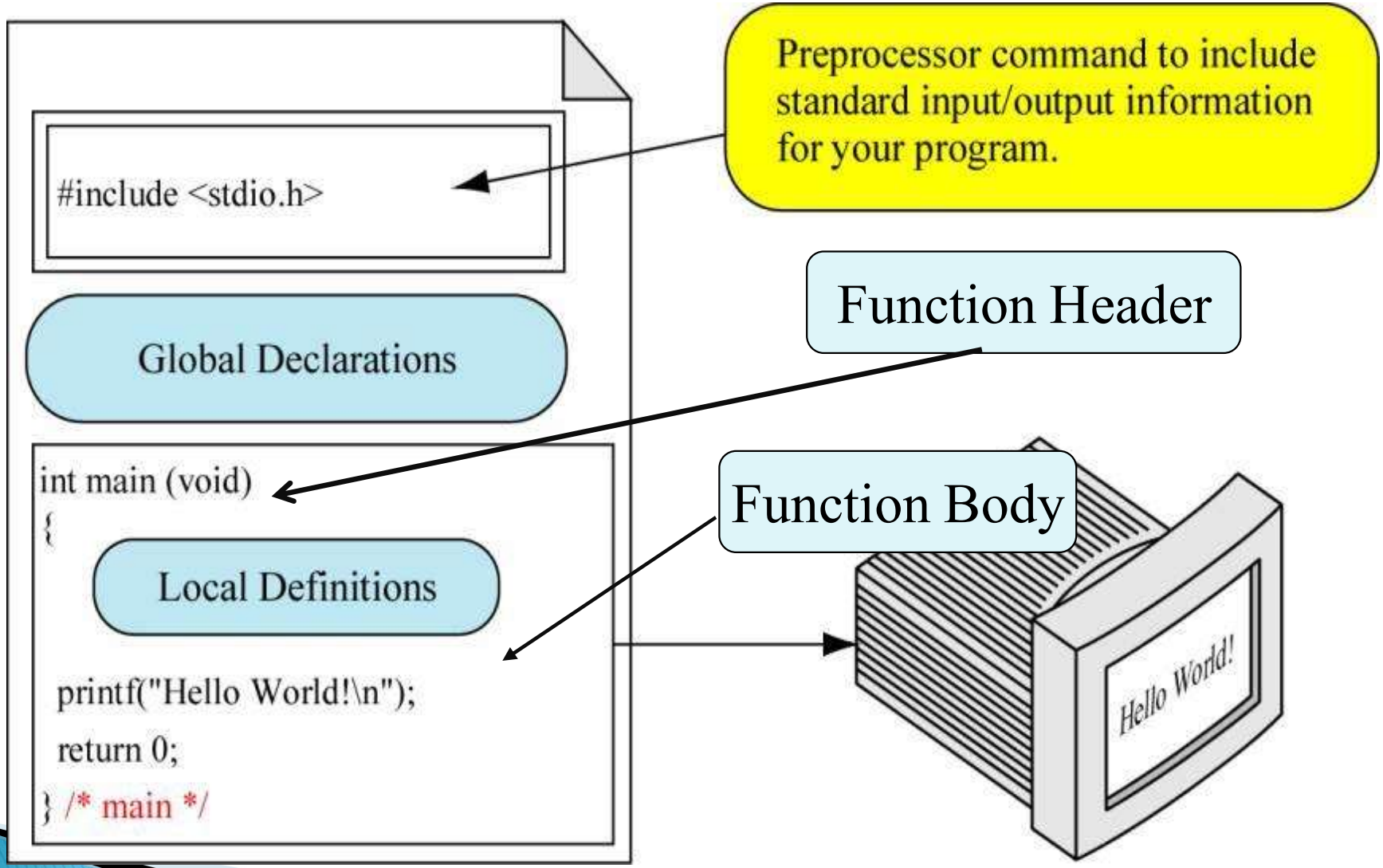
Review C Language



C Program structure

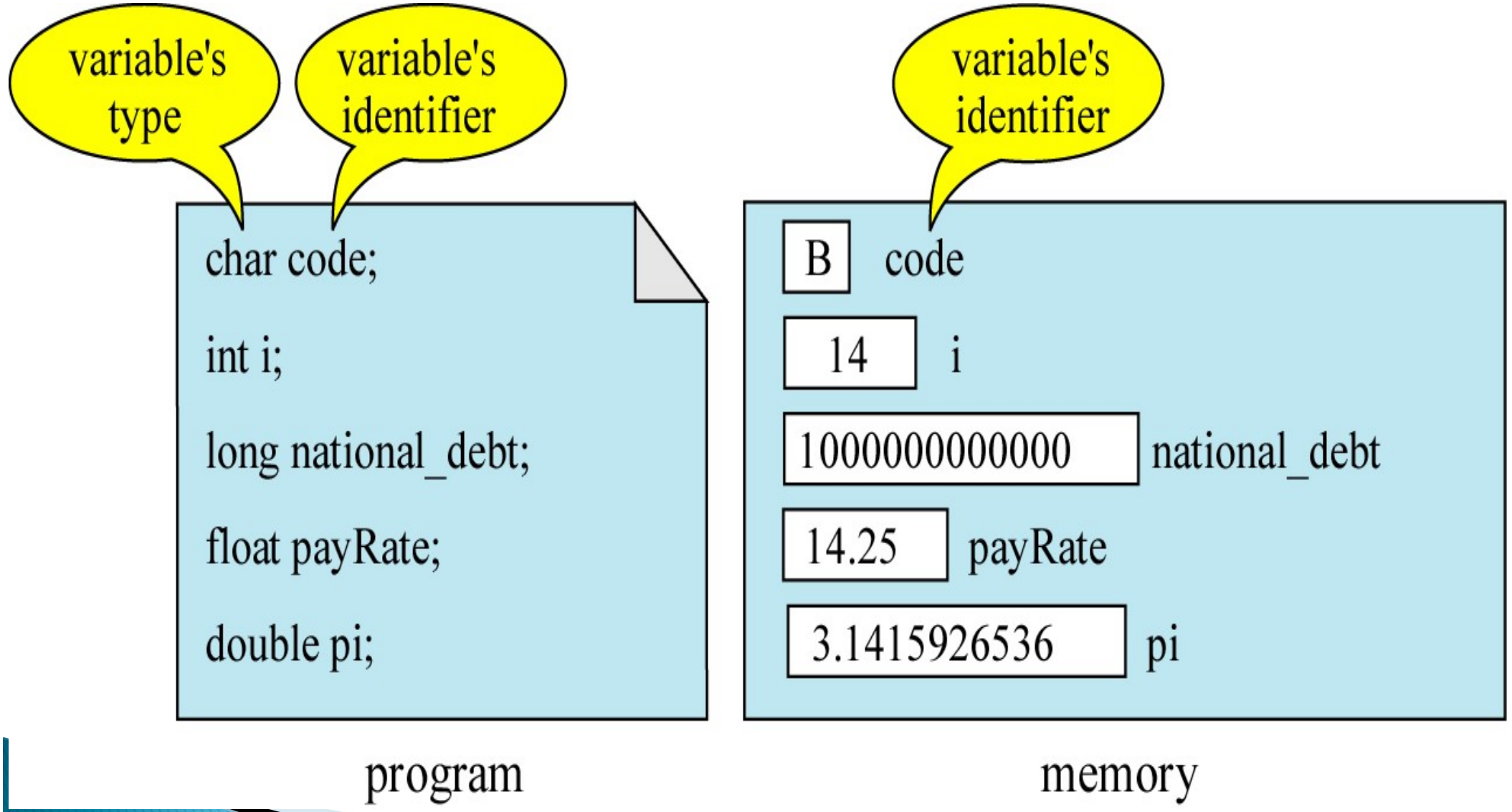


Identifiers Declarations

- 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.
- A standard identifier should not be redefined.
- C compilers are case sensitive. (Rate, rate and RATE are viewed as different identifiers)

- Valid identifiers: `letter1, inches, KM_PER_MILE`
- Invalid identifiers: `1letter, Happy*trout,return`

Giving a Value to a Variable

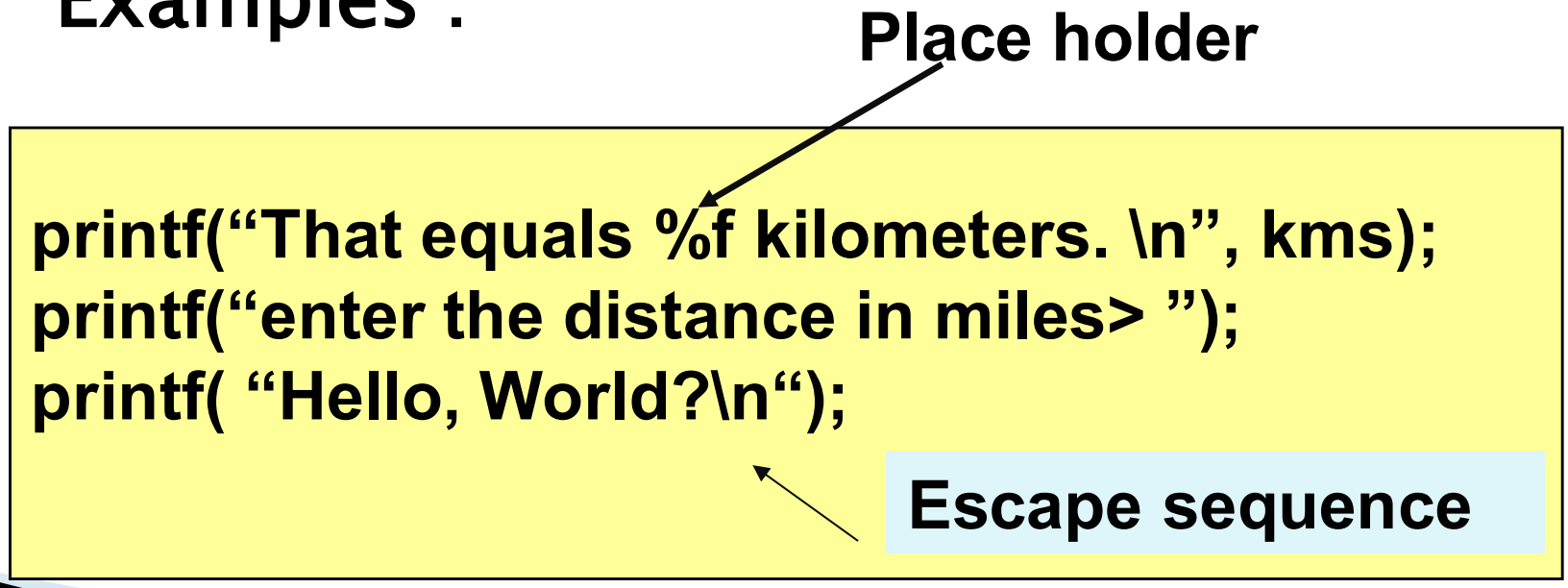


Output Function

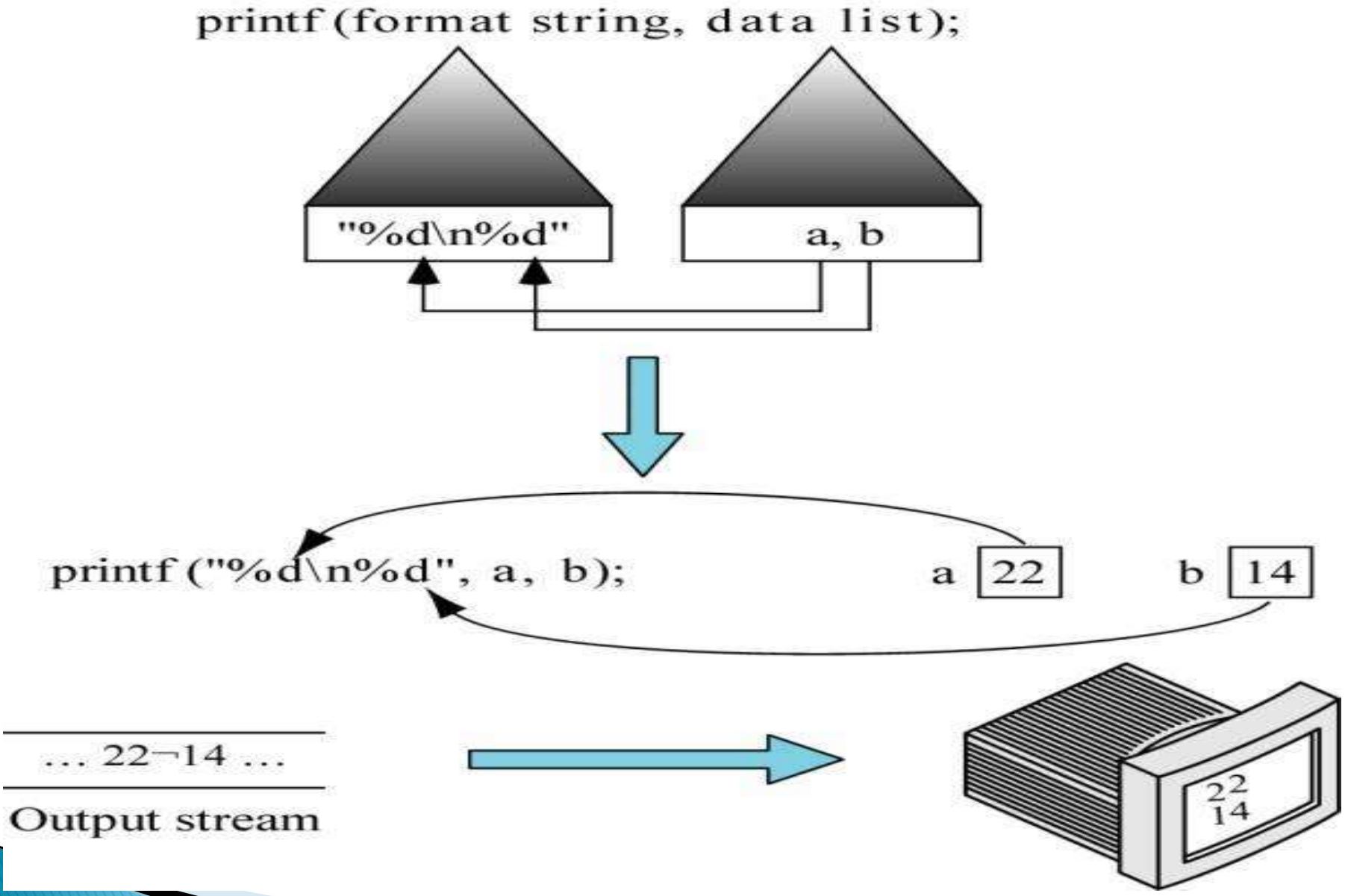
SYNTAX

```
printf( format string , print list ) ;
Printf(format string);
```

Examples :



Output Function



Input Function

SYNTAX

```
scanf( format string , input list ) ;
```

Examples :

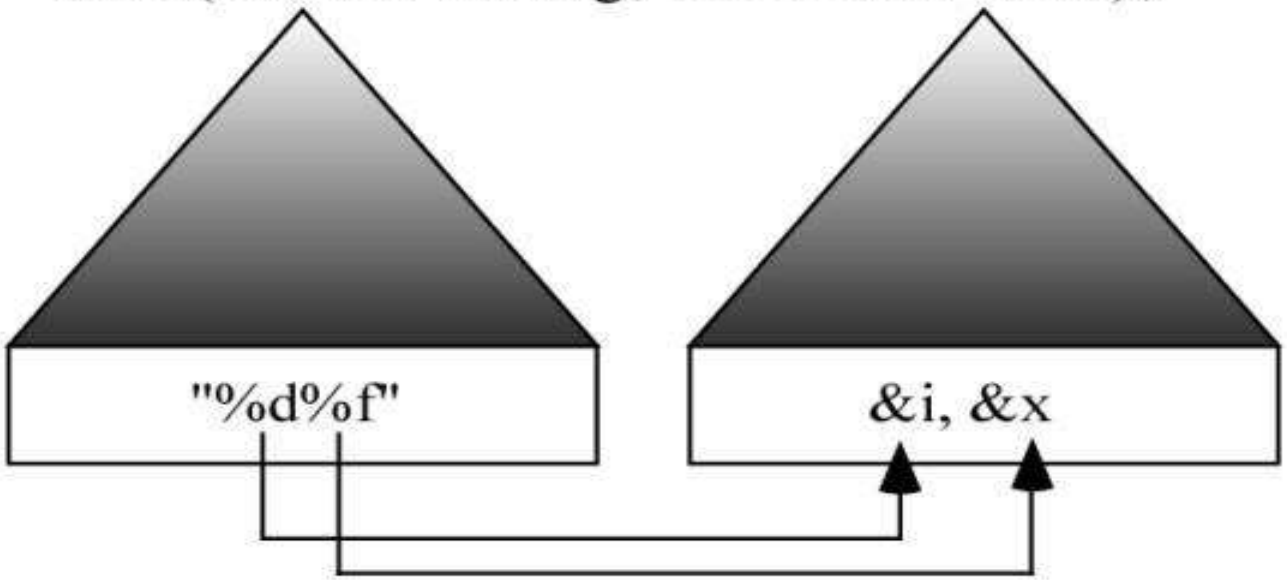
Place holder

```
scanf("%lf", &miles);
```

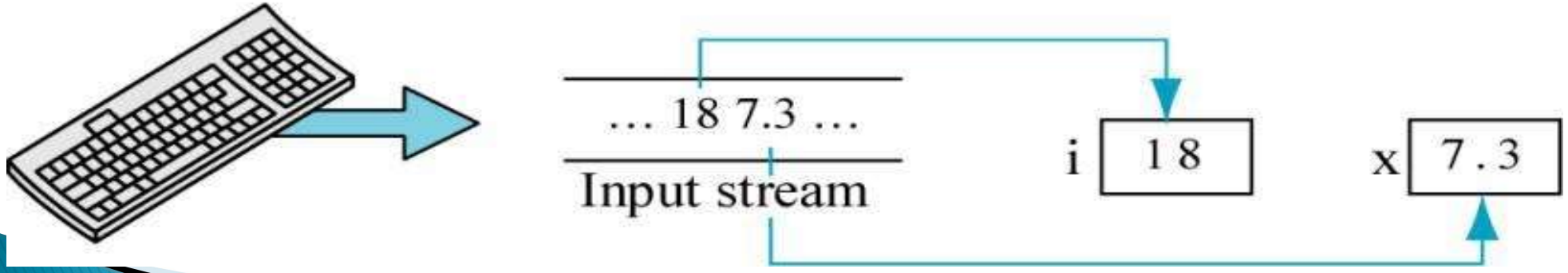
Ampersand

Input Function

scanf(format string, address list);



scanf("%d%f", &i, &x);



Programming Examples: Example-1

- Write a program to ask the user for the width and length of a piece of land and then tell him how many orange trees he can grow on it. Given that each orange tree requires 4 m^2.

Programming Examples

```
#include <stdio.h>
# define one_tree_space  4

int main()
{
   int   length,width, area, no_of_tree;

   printf("Enter length of the land> ");
   scanf("%d", &length);
   printf("Enter width of the land> ");
   scanf("%d", &width);
   area = length * width;
   no_of_tree = area / one_tree_space;
   printf("The available number of trees is %d
   trees\n", no_of_tree);

   return(0);
}
```