Abdelhafid Boussouf University
Center of Mila

Structure of Computers and Applications
1st year ST – ENGINEERING

Part 2: The basics of Algorithm and Program

Course 08: INPUT AND OUTPUT STATEMENTS (I/O)

By

Dr. Farouk KECITA

Academic year : 2024/2025

Introduction

- **Input** means to **provide** the program with some data to be used in the program
- **Output** means to display data on **screen** or write the data to a printer or a file.
- C programming language provides **many** built-in functions **to read** any given input and **to display** data on screen when there is a need to output the result.
- All these built-in functions are **present** in C header files that we discussed earlier.
- The Formatted i**nput/output** functions are standard functions capable to read and write all types of data values.
- The formatted **I/O** functions supported by C are **printf()** and **scanf().**

1_ **Scanf()** and **Printf()** functions

- The standard input-output header file, named **stdio.h** contains the definition of the functions **scanf()** and **printf()**, which are used *to **take*** Input from **user** and *to **display*** Output on **screen** respectively.

1_1. printf():

- This is an output statement. To output data on to a screen, we use the standard output library function, represented by the word **"printf"** followed by the open and closing parentheses **()**.
- It is used to display the **value** of a **variable** or a **message** on the **screen**.
- **Syntax:**
  printf("<message>");
  printf("<control string>", argument list separated with commas);

1_ Scanf() and Printf() functions

- **Examples:**

```
printf("This is C statement");
printf("The number is %d", a);
printf("The number %d is equal to %f", 2, 11.5);
printf("The number %d is not equal to %d", x, y);
```

1_2. scanf():

- This is an input statement. Data can be **stored** in the **variables** after accepting the **values** from the **user** through the **keyword**, by using a standard **library function** for **input operation**.
- This allows a **program** to get **user input** from the **keyboard**.
- This means that the **program** gets input values for **variables** from **users**.

1_ **Scanf()** and **Printf()** functions

1_2. scanf():

Syntax:

scanf("<format code>",**list of address** of variables separated by commas);

Examples:

scanf("%d", &*a*);

scanf("%f", &*b*);

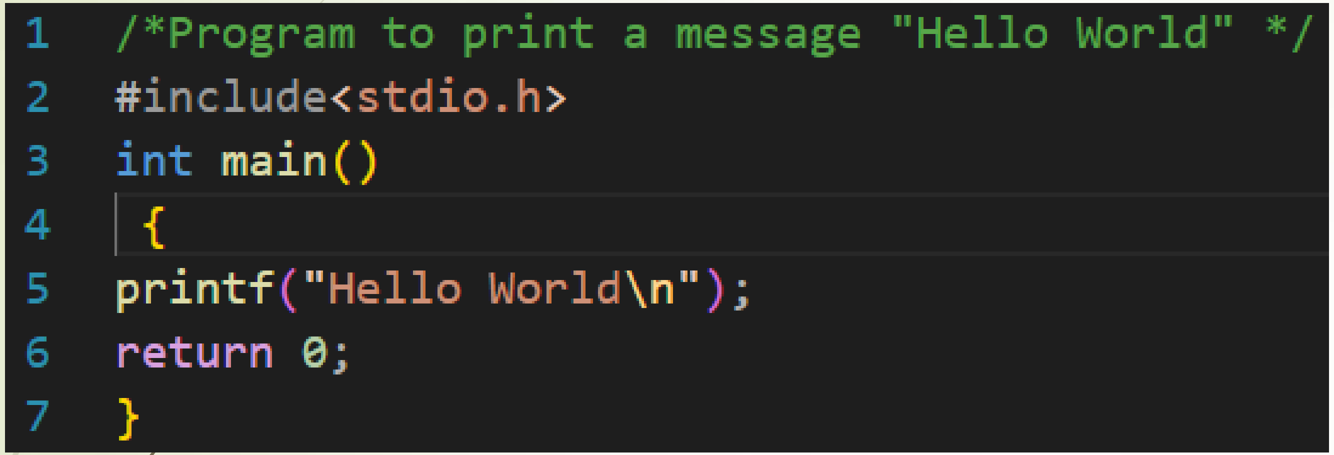
scanf("%d %f %c", &a, &b, &y);

Note 01: The '**&**' sign before ***a***. **&*a*** denotes the address of ***a*** and value is stored in that address.

Note 02: Conversion format string "**%f**" is used for floats to take input and to display floating value of a variable.

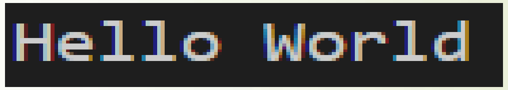
2_ Print a message or Multiple Statements

- ❑ To print a message "Hello World" on the screen



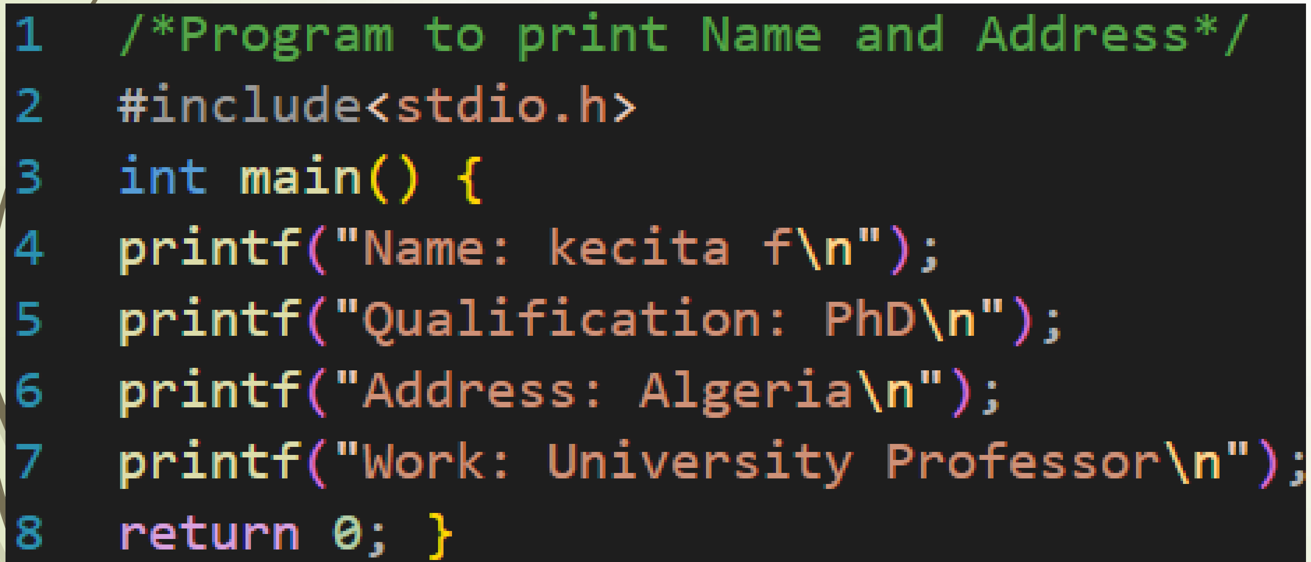
```c
/*Program to print a message "Hello World" */
#include<stdio.h>
int main()
 {
printf("Hello World\n");
return 0;
}
```

Output:



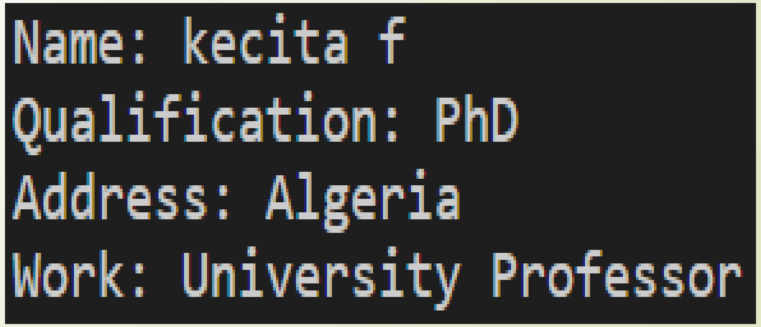
```
Hello World
```

- ❑ To Display Multiple Statements



```c
/*Program to print Name and Address*/
#include<stdio.h>
int main() {
printf("Name: kecita f\n");
printf("Qualification: PhD\n");
printf("Address: Algeria\n");
printf("Work: University Professor\n");
return 0; }
```

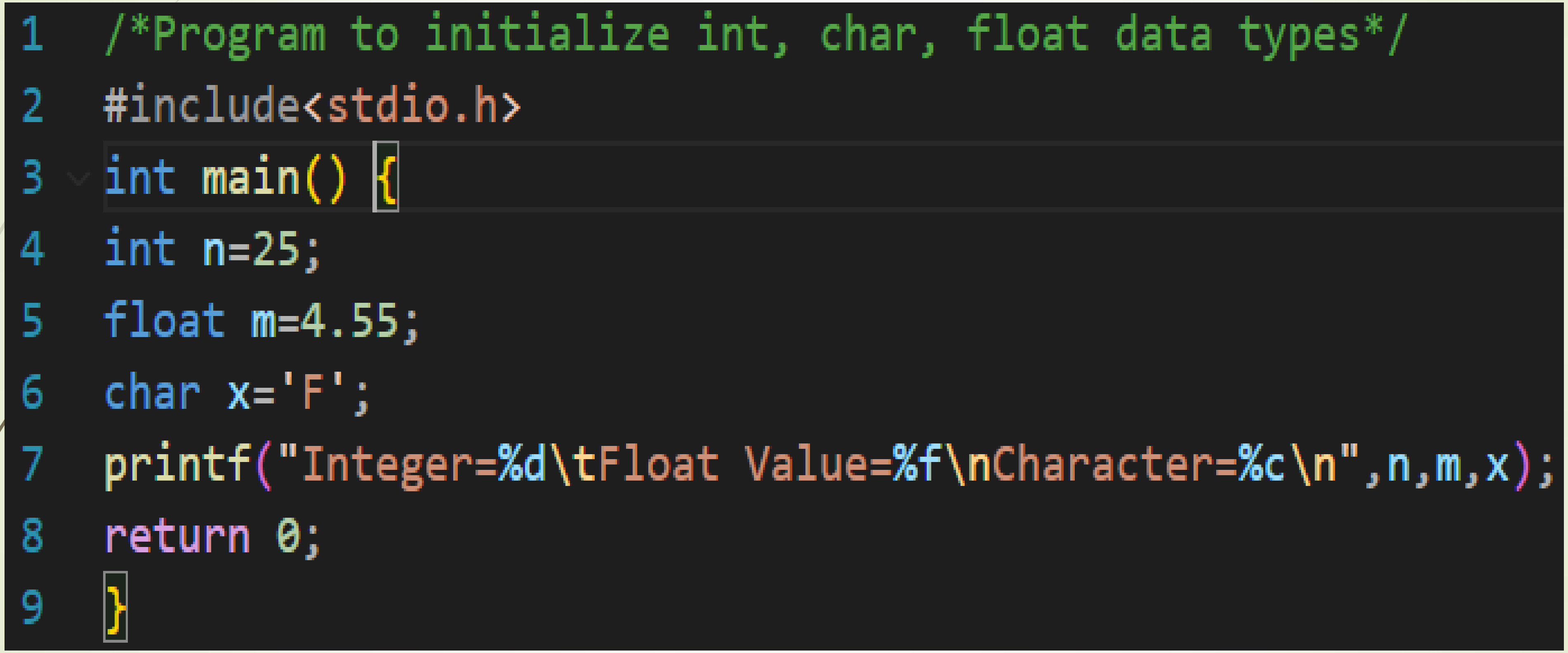
Output:



```
Name: kecita f
Qualification: PhD
Address: Algeria
Work: University Professor
```

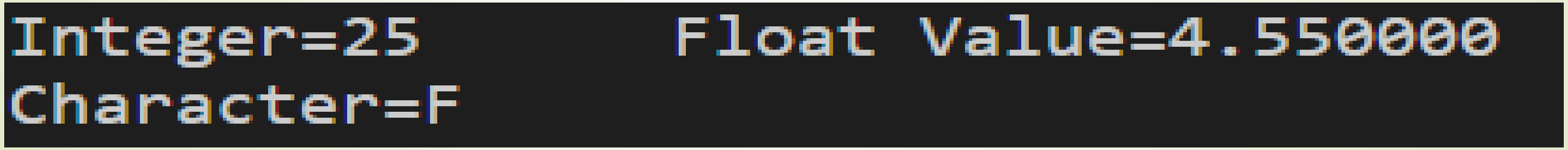
3_ To Initialize data types and display them

- ❑ Program to Initialize int, char, float data types display them



```c
/*Program to initialize int, char, float data types*/
#include<stdio.h>
int main() {
int n=25;
float m=4.55;
char x='F';
printf("Integer=%d\tFloat Value=%f\nCharacter=%c\n",n,m,x);
return 0;
}
```

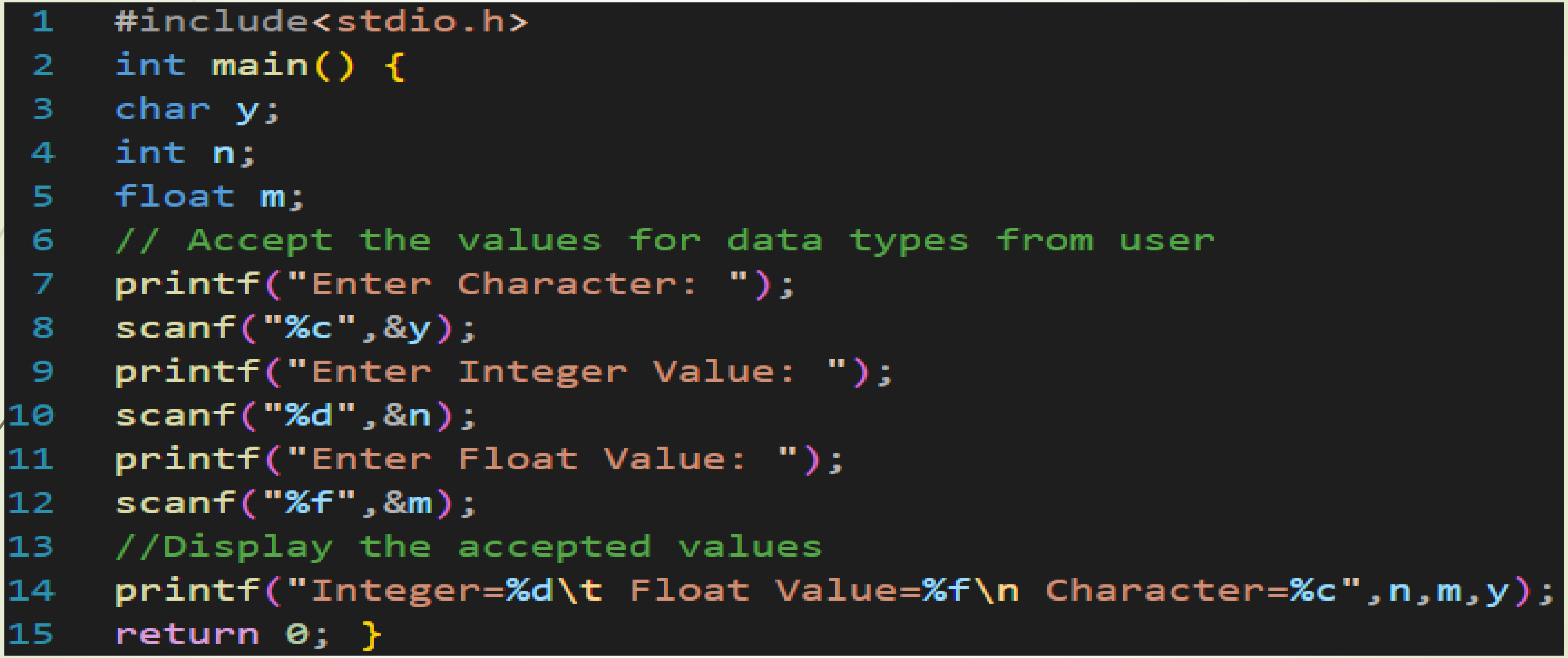
Output:



```
Integer=25      Float Value=4.550000
Character=F
```

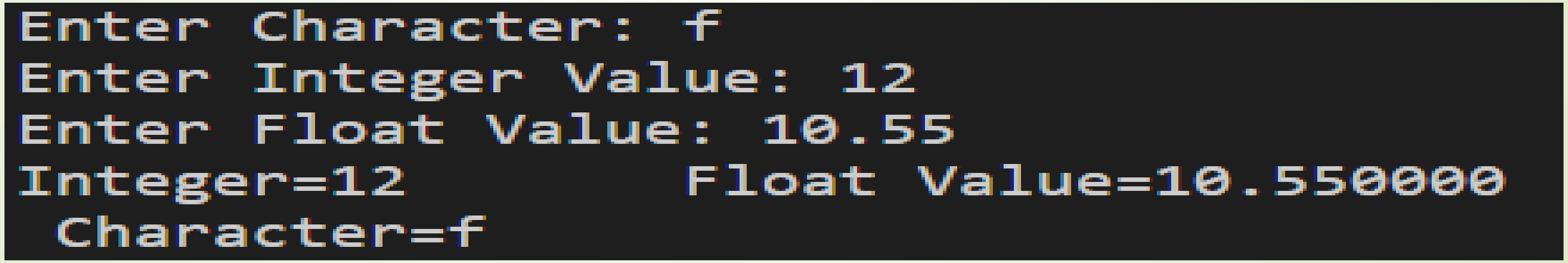
4_ To accept the values of data types and display them

- To accept the values of int, float, char data types and display them



```c
#include<stdio.h>
int main() {
char y;
int n;
float m;
// Accept the values for data types from user
printf("Enter Character: ");
scanf("%c",&y);
printf("Enter Integer Value: ");
scanf("%d",&n);
printf("Enter Float Value: ");
scanf("%f",&m);
//Display the accepted values
printf("Integer=%d\t Float Value=%f\n Character=%c",n,m,y);
return 0; }
```

Output:



```
Enter Character: f
Enter Integer Value: 12
Enter Float Value: 10.55
Integer=12          Float Value=10.550000
 Character=f
```