

C Programming Language

How to read from user

Dr. Mostafa A. Elhosseini

Revise

✂ How to print output?

- printf – print formatted
- Format specifier
- How to print constants
- Escape sequence

✂ Variable

- Variable?
- Types
- Variable declaration
- Variable definition

Agenda

✂ How to read from user

- `scanf( )`
 - & ampersand – address operator
- `getche( )` function
 - Get character with echo
- `getchar( )`
 - Get the character – it did not pass the character typed until the user hits [Return]
- `getch( )`
 - Get character without echo

✂ How to define constant

- Rvalue: Right value

■ scanf()

```
#include <stdio.h>

main()
{
    float x;
    scanf("%f", &x);
    ==
}


```

Diagram annotations:

- Decimal Var (points to `x`)
- Address operator Ampersand (points to `&`)
- format specifier (points to `%f`)
- Address of `x` (points to `&x`)

■ scanf()

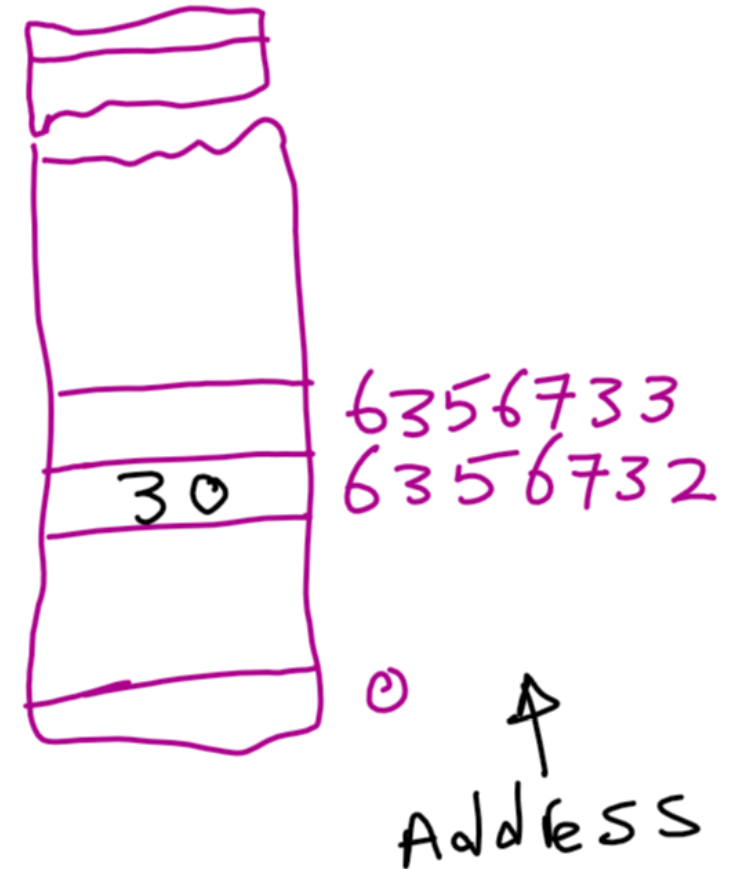
- ✧ Address operator represented by the ampersand (&)
 - The & allows the variable to be changed
 - Without this character, C prevents scanf() from altering variable and it would retain the random value it had previously (unless you'd remembered to initialize it)
- ✧ scanf() requires the use of an ampersand before the variable name
- ✧ The C compiler requires the arguments to scanf() to be the address of the variables rather than the variables themselves
- ✧ The memory of your computer is divided into bytes, and these bytes are numbered, from 0 to the upper limit of your memory. These numbers are called the "addresses" of the bytes.
- ✧ Every variable occupies a certain location in memory, and its address is that of the first byte it occupies

■ Address of variable

```
int main()  
{  
    int num = 30;  
    printf("\nvalue of number :%d", num);  
    printf("\naddress of number: %d", &num);  
  
    getch();  
  
    return 0;  
}
```

"E:\Courses\Taibah\Programming essential using C\Code\scanf01\bin\Debug\scanf01.exe"

value of number :30
address of number: 6356732



■ scanf()

```
int main()  
{  
    float x,y,z;  
    printf("Enter two numbers in the form x y to get the sum\n");  
    scanf("%f %f",&x,&y);  
    z = x+y;  
    printf("sum = %f",z);  
    return 0;  
}
```

"E:\Courses\Taibah\Programming essential using C\Code\scanf01\bin\Debug\scanf01.exe"

```
Enter two numbers in the form x y to get the sum  
20 30  
sum = 50.000000  
Process returned 0 (0x0)   execution time : 4.672 s  
Press any key to continue.
```

■ scanf()

```
int main()  
{  
    float x, y, z;  
    printf("Enter two numbers in the form x:y to get the sum\n");  
    scanf("%f:%f", &x, &y);  
    z = x+y;  
    printf("sum = %f", z);  
    return 0;  
}
```

"E:\Courses\Taibah\Programming essential using C\Code\scanf01\bin\Debug\scanf01.exe"

Enter two numbers in the form x:y to get the sum

12.1:1.9

sum = 14.000000

Process returned 0 (0x0) execution time : 6.797 s

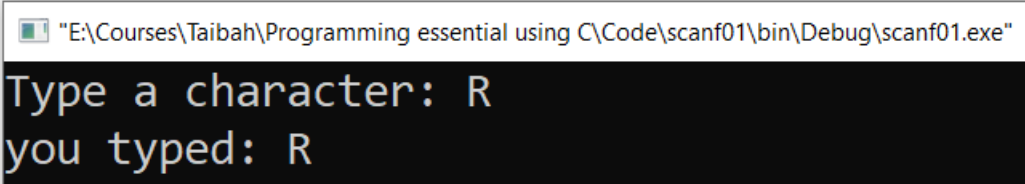
Press any key to continue.

getche() // getch() // getchar()

- ✧ The `scanf( )` function has **one weak point** — you need to type a **[Return]** before the function will digest what we have typed
- ✧ But we often want a function that will **read a single character** the instant it is typed **without waiting** for **[Return]**.
 - For instance, in a game we might want a space-ship to move each time we pressed one of the cursor control
- ✧ `getche( )` → get character with **e**cho
 - if you make a mistake, you can't backspace to correct it, since as soon as you type a character, it is delivered to your program
- ✧ `getch( )` → get character without echo
- ✧ `getchar( )` → **get character** — it does not pass the character typed by the user to the program until the user hits **[Return]**

getche() // getch() // getchar()

```
int main()  
{  
    char ch;  
    printf("Type a character: ");  
    ch = getchar();  
    printf("you typed: %c", ch);  
  
    return 0;  
}
```



Constant

```
4 | int main()  
5 | {  
6 |     const days_in_week = 7;  
7 |     const float xx = 10;  
8 |     xx = 40;  
9 |  
10 |     printf("days = %d\n", days_in_week);  
11 |     printf("days = %f\n", xx);  
12 |  
13 |     return 0;
```

Lvalue
Rvalue

L.. Message
... 6 warning: type defaults to 'int' in declar...
... 8 error: assignment of read-only variable 'xx'